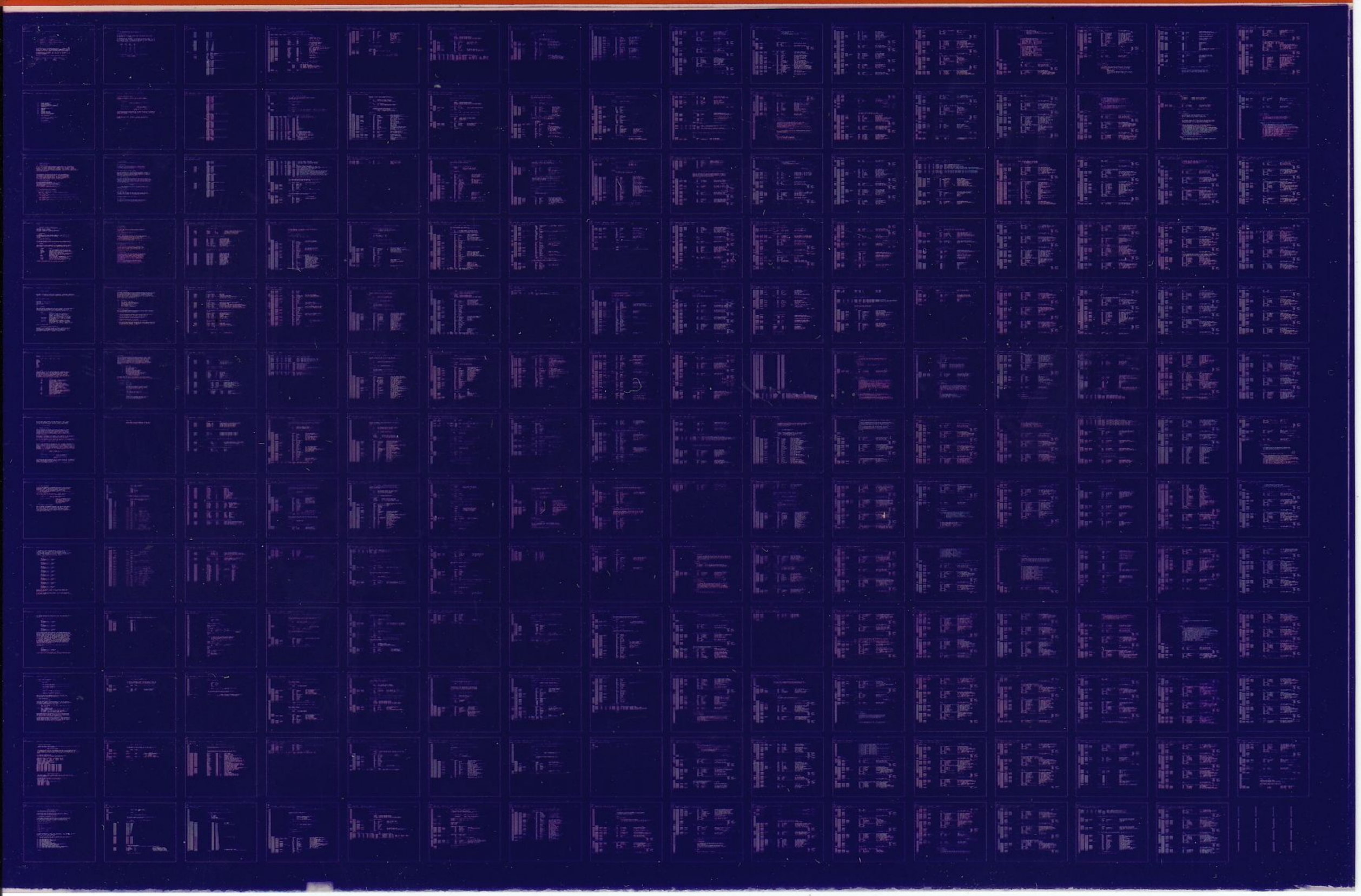
TSV05

TSV05 CTRL PT 2
CVTSBE0

AH-T096E-MC
1 OF 2 SEP 1987

digital
MADE IN USA



TSV05

TSV05 CTRL PT 2
CVTSBE0

AH-T096E-MC
2 OF 2 SEP 1987

digital
MADE IN USA

.REM_

IDENTIFICATION

PRODUCT ID: AC T095E-MC

PRODUCT TITLE: CVTSBEO TSV05 CTRL PART 2

DEPARTMENT: COMPUTER SPECIAL SYSTEMS/PGG

DATE: JUNE 08, 1987

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.
CVTSA? 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-E-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.

	TESTS	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 CALL D 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 Or 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" FO' NO.

```
Ø 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 "Ø 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.

```
        Ø UNITS (D) ?  8<CR>

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

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

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

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

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

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

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

        UNIT 8
        CSR ADDRESS (O)  160000<CR>
        SUB-DEVICE Ø (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 (O) ?  160000<CR>
        SUB-DEVICE # (O) ?  0,1<CR>
        Q-FACTOR (O) O ?  1.0<CR>

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

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

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

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

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

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

AS YOU CAN SEE FROM THIS EXAMPLE, NULL REPLIES (COMMAS ENCLOSING 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  (0) 172520 ? <CR>

 VECTOR (0) 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.

REVISION E - APRIL 1987

CHANGES MADE TO ALLOW DIAGNOSTICS TO WORK WITH
THE NEW TSV05 MICROCODE (REVISION 2). THE NEW
TSV05 MICROCODE ALWAYS IN EXTENDED FEATURE MODE.
-

```
                                .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.
                                ;

                                POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
002000                          HEADER  CVTSB,E,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  105                     .ASCII  /E/
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  101302                  .WORD   L$HARD
002020                  L$SPCP::                ;POINTER TO S.W. QUES.
002020  101434                  .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
```

J2

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  003404                  .WORD   L$DVTYP
002062                  L$REPP::                ;PTR. TO REPORT CODE
002062  022254                  .WORD   L$RPT
002064                  L$EXP4::
002064  000000                  .WORD   0
002066                  L$EXP5::
002066  000000                  .WORD   0
002070                  L$AUT::                 ;PTR. TO ADD UNIT CODE
002070  021742                  .WORD   L$AU
002072                  L$DUT::                 ;PTR. TO DROP UNIT CODE
002072  022040                  .WORD   L$DU
002074                  L$LUN::                 ;LUN FOR EXERCISERS TO FILL
002074  000000                  .WORD   0
002076                  L$DESP::                ;POINTER TO DIAG. DESCRIPTION
002076  003412                  .WORD   L$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  021146                  .WORD   L$INIT
002106                  L$CCP::                 ;PTR. TO CLEAN UP CODE
002106  022226                  .WORD   L$CLEAN
002110                  L$ACP::                 ;PTR. TO AUTO CODE
002110  022146                  .WORD   L$AUTO
002112                  L$PRT::                 ;PTR. TO PROTECT TABLE
002112  021136                  .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
```

K2

DISPATCH TABLE

```
                                        .SBTTL  DISPATCH TABLE

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

002122                                  DISPATCH 12
002122  000014                          .WORD   12
002124                          L$DISPATCH::
002124  023036                          .WORD   T1
002126  024016                          .WORD   T2
002130  026010                          .WORD   T3
002132  031404                          .WORD   T4
002134  034174                          .WORD   T5
002136  037766                          .WORD   T6
002140  050100                          .WORD   T7
002142  051360                          .WORD   T8
002144  062206                          .WORD   T9
002146  066256                          .WORD   T10
002150  074114                          .WORD   T11
002152  077624                          .WORD   T12
```

L2

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:
```

M2

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:

002176                                  ENDMOD
```

N2

SOFTWARE P-TABLE

```
1014                                    .TITLE  TSV3 - GLOBAL AREAS
1015                                    .SBTTL  GLOBAL EQUATES SECTION
1020
1026
1027 002176                             BGNMOD  TSV3
     002176                     TSV3::
1028
1029                                    .SBTTL  GLOBAL EQUATES SECTION
1030
1031                                    ;**
1032                                    ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
1033                                    ; ARE USED IN MORE THAN ONE TEST.
1034                                    ;--
1035
1039 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
```

B3

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
1040
1041 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
```

MEMORY MANAGEMENT DEFINITIONS

```
                        ;*USER "I" PAGE ADDRESS REGISTERS
                        UIPAR0= 177640
                        UIPAR1= 177642
                        UIPAR2= 177644
                        UIPAR3= 177646
                        UIPAR4= 177650
                        UIPAR5= 177652
                        UIPAR6= 177654
                        UIPAR7= 177656
                        .IF NB
                        ;*USER "D" PAGE ADDRESS REGISTERS
                        UDPAR0= 177660
                        UDPAR1= 177662
                        UDPAR2= 177664
                        UDPAR3= 177666
                        UDPAR4= 177670
                        UDPAR5= 177672
                        UDPAR6= 177674
                        UDPAR7= 177676
                        .ENDC
                        .ENDC
                        .IF NB
                        ;*SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS
                        SIPDR0= 172200
                        SIPDR1= 172202
                        SIPDR2= 172204
                        SIPDR3= 172206
                        SIPDR4= 172210
                        SIPDR5= 172212
                        SIPDR6= 172214
                        SIPDR7= 172216
                        .IF NB
                        ;*SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS
                        SDPDR0= 172220
                        SDPDR1= 172222
                        SDPDR2= 172224
                        SDPDR3= 172226
                        SDPDR4= 172230
                        SDPDR5= 172232
                        SDPDR6= 172234
                        SDPDR7= 172236
                        .ENDC
                        ;*SUPERVISOR "I" PAGE ADDRESS REGISTERS
                        SIPAR0= 172240
                        SIPAR1= 172242
                        SIPAR2= 172244
                        SIPAR3= 172246
                        SIPAR4= 172250
                        SIPAR5= 172252
                        SIPAR6= 172254
                        SIPAR7= 172256
                        .IF NB
                        ;*SUPERVISOR "D" PAGE ADDRESS REGISTERS
                        SDPAR0= 172260
                        SDPAR1= 172262
                        SDPAR2= 172264
                        SDPAR3= 172266
```

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

```
1098                                    ;*
1099                                    ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
1100                                    ;(XST1)
1101                                    ;-
1102        100000                      X1.DLT  = BIT15         ;DATA LATE
1103        040000                      X1.SPARE= BIT14         ;NOT USED
1104        020000                      X1.COR  = BIT13         ;CORRECTABLE DATA ERROR
1105        017375                      X1.MBZ  = BIT12+BIT11+BIT10+BIT9+BIT7+BIT6+BIT5+BIT4+BIT3+BIT2+BIT0 ;ALWAYS 0
1106        000400                      X1.RBP  = BIT8          ;READ BUS PARITY ERROR
1107        000002                      X1.UNC  = BIT1          ;UNCORRECTABLE DATA OR HARD ERROR
1108
1109                                    ;*
1110                                    ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
1111                                    ;(XST2)
1112                                    ;-
1113        100000                      X2.OPM  = BIT15         ;OPERATION IN PROGRESS (TAPE MOVING)
1114        040000                      X2.RCE  = BIT14         ;RAM CHECKSUM ERROR
1115        035400                      X2.SPARE= BIT13+BIT12+BIT11+BIT9+BIT8   ;NOT USED BY TSV05 (ALWAYS=0)
1116        002000                      X2.WCF  = BIT10         ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
1117        000200                      X2.EXTF = BIT7          ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
1118        000100                      X2.BUFE = BIT6          ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
1119        000077                      X2.REV  = 000077        ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
1120        000007                      X2.UNIT = BIT2+BIT1+BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.
1121
1122                                    ;*
1123                                    ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
1124                                    ;(XST3)
1125                                    ;-
1126        177400                      X3.MDE  = 177400        ;MICRO DIAGNOSTIC ERROR CODE
1127        000200                      X3.SPARE= BIT7          ;NOT USED BY TSV05
1128        000100                      X3.OPI  = BIT6          ;OPERATION INCOMPLETE
1129        000040                      X3.REV  = BIT5          ;REVERSE
1130        000020                      X3.TRF  = BIT4          ;TRANSPORT RESPONSE FAILURE
1131        000010                      X3.DCK  = BIT3          ;DENSITY CHECK
1132        000006                      X3.MBZ  =BIT2+BIT1      ;NOT USED ALWAYS 0
1133        000001                      X3.RIB  = BIT0          ;REVERSE INTO BOT
1134
1135                                    ;*
1136                                    ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
1137                                    ;(XST4)
1138                                    ;-
1139        100000                      X4.HSP  = BIT15         ;HIGH SPEED
1140        040000                      X4.RCE  = BIT14         ;RETRY COUNT EXCEEDED
1141        020000                      X4.TSM  = BIT13         ;TRANSPORT SPECIAL MODE
1142        017400                      X4.MBZ  = BIT12+BIT11+BIT10+BIT9+BIT8   ;NOT USED ALWAYS 0
1143        000377                      X4.WRC  = 0003^7        ;WRITE RETRY COUNT FIELD
1144
1145                                    ;*
1146                                    ;
1147                                    ;TSSR TERMINATION CODES (BIT 0-2)
1148                                    ;
1149                                    ;-
1150
1151        000006                              TSREJ=  3*2             ;COMMAND REJECTED
1152        000006                              UNREC=  6               ;UNRECOVERABLE ERROR
```

TSV05 REGISTER AND PACKET DEFINITIONS

```
                                ;*
                                ;
                                ;DEVICE REGISTER OFFSETS
                                ;
                                ;-

000000                          TSBA==  0
000000                          TSDB==  0               ;TSDB/TSBA REGISTER
000001                          TSBAH== 1
000001                          TSDBH== 1               ;TSDB/TSBA REGISTER HIGH BYTE
000002                          TSSR==  2               ;TSSR REGISTER
000003                          TSSRH== 3               ;TSSR REGISTER HIGH BYTE

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

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

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

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

H3

TSV05 REGISTER AND PACKET DEFINITIONS

```
1205                                    ;*
1206                                    ; CONTROLLER RAM DEFINITIONS
1207                                    ;-
1208    000167                          RMCHBEG = 167           ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
1209    000200                          RMCHEND = 200           ;CHARACTERISTICS IO DATA END RAM ADDRESS
1210    000201                          RMPKTBEG= 201           ;COMMAND PACKET BEGIN RAM ADDRESS
1211    000210                          RMPKTEND= 210           ;COMMAND PACKET END RAM ADDRESS
1212    000215                          RMMSGBEG= 215           ;MESSAGE BUFFER BEGIN RAM ADDRESS
1213    000234                          RMMSGEND= 234           ;MESSAGE BUFFER END RAM ADDRESS
1214                                    ;*
1215                                    ;
1216                                    ;REGISTER DEFINITIONS IN THE MESSAGE BUFFER
1217                                    ;
1218                                    ;-
1219
1220    000006                          XST0==  6               ;EXTENDED STATUS REGISTER 0 (WORD 4)
1221    000010                          XST1==  8.              ;EXTENDED STATUS REGISTER 1 (WORD 5)
1222    000012                          XST2==  10.             ;EXTENDED STATUS REGISTER 2 (WORD 6)
1223    000014                          XST3==  12.             ;EXTENDED STATUS REGISTER 3 (WORD 7)
1224    000016                          XST4==  14.             ;EXTENDED STATUS REGISTER 4 (WORD 8)
1225
1226                                    ;*
1227                                    ;
1228                                    ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
1229                                    ;
1230                                    ;-
1231
1232    000002                                  PKLOW   = 2             ;LOW ORDER CHARACTERISTIC DATA POINTER
1233    000004                                  PKHI    = 4             ;HIGH ORDER CHARACTERISTIC DATA POINTER
1234    000006                                  PKBCNT  = 6             ;NUMBER OF BYTES IN DATA PACKET
1235
1236    000010                                  EXBCNT=10               ;NUMBER OF BYTES IN EXTENDED DATA PACKET
1237
1238                                    ;*
1239                                    ;DATA PACKET OFFSETS FOR WRITE SUBSYTEM COMMAND
1240                                    ;-
1241    000000                                  BSEL0   = 0             ;BYTE 0
1242    000001                                  BSEL1   = 1             ;BYTE 1
1243    000002                                  SEL2    = 2             ;WORD 2
1244    000004                                  SELDATA = 4             ;WORD 3
```

I3


```
1246                                    ;+
1247                                    ;BSEL0 SELECT CODES FOR WRITE SUBSYSTEM COMMAND
1248                                    ;-
1249            000000                  PW.NOP          = 0             ;NO OP
1250            000001                  PW.RDRAM        = 1             ;READ RAM
1251            000002                  PW.WTRAM        = 2             ;WRITE RAM
1252            000003                  PW.RFIFO        = 3             ;READ FIFO
1253            000004                  PW.WFIFO        = 4             ;WRITE FIFO
1254            000005                  PW.RDSTAT       = 5             ;READ STATUS
1255            000006                  PW.WCTL         = 6             ;WRITE TAPE CONTROL
1256            000007                  PW.WFMT         = 7             ;WRITE TAPE FORMAT
1257            000010                  PW.WMISC        = 10            ;WRITE MISCELLANEOUS
1258            000011                  PW.WNPR         = 11            ;WRITE NPR CONTROL
1259            000020                  PW.D22          = 20            ;DO MICROTEST 22
1260            000021                  PW.D11          = 21            ;DO MICROTEST 11
1261            000022                  PW.D13          = 22            ;DO MICROTEST 13
1262            000023                  PW.NO1311       = 23            ;DISABLE MICROTEST 11 AND 13
1263            000024                  PW.RDEXT        = 24            ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS
1264
1265                                    ;+
1266                                    ;BSEL1 CODES FOR WRITE TAPE CONTROL
1267                                    ;-
1268            000200                  WC.IFAD         = BIT7          ;IFAD - FORMATTER ADDRESS
1269            000100                  WC.IOTAD        = BIT6          ;ITAD0  - TRANSPORT ADDRESS BIT 0
1270            000040                  WC.I1TAD        = BIT5          ;ITAD1    TRANSPORT ADDRESS BIT 1
1271            000020                  WC.ISRESV       = BIT4          ;IRESV5 - RESERVED #5
1272            000010                  WC.IREW         = BIT3          ;IREW   - REWIND
1273            000004                  WC.IRWU         = BIT2          ;IRWU     REWIND AND UNLOAD
1274            000002                  WC.IFEN         = BIT1          ;IFEN     FORMATTER ENABLE
1275            000001                  WC.IGO          = BIT0          ;GO
1276
1277                                    ;+
1278                                    ;BSEL1 CODES FOR WRITE FORMAT
1279                                    ;-
1280            000200                  WF.IHISP        = BIT7          ;IHISP  - HIGH SPEED
1281            000100                  WF.IWRT         = BIT6          ;IWRT     WRITE
1282            000040                  WF.IREV         = BIT5          ;IREV   - REVERSE
1283            000020                  WF.IWFM         = BIT4          ;IWFM   - WRITE FILE MARK
1284            000010                  WF.IEDIT        = BIT3          ;IEDIT  - EDIT
1285            000004                  WF.IERASE       = BIT2          ;IERASE - ERASE
1286            000002                  WF.I3RESV       = BIT1          ;IRESV3 - RESERVED #3
1287            000001                  WF.I4RESV       = BIT0          ;IRESV4 - RESERVED #4
1288
1289                                    ;+
1290                                    ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
1291                                    ;
1292            000200                  MS.EXT          = BIT7          ;INVERT SENSE OF EXTENDED FEATURES SWITCH
1293            000020                  MS.RSFIFO       = BIT4          ;RESET FIFO AND INPUT PARITY ERRORR
1294            000010                  MS.RSTAPE       = BIT3          ;RESET TAPE STATUS IN 2 FLIP FLOPS
1295            000006                  MS.ATTN         = BIT2!BIT1     ;ATTENTION TRIGGER FIELD
1296            000001                  MS.RSD          = BIT0          ;RESET TIMER A,B THEN DELAY TIMES IN SEL2
```

J3

TSV05 REGISTER AND PACKET DEFINITIONS

```
                                ;*
                                ; MS.ATTN SUBCODES
                                ;
                                        MSA.NOP = 0*2           ;NO OP (NOTHING TRIGGERED)
                                        MSA.VOL = 1*2           ;SIMULATE ON LINE/OFF LINE TRANSISTION
                                        MSA.NRAM= 2*2           ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
                                        MSA.FRAM= 3*2           ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
                                ;*
                                ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
                                ;
000200                          NP.IR           = BIT7          ;INTERRUPT REQUEST (0 1 TRANSITION)
000100                          NP.OUT          = BIT6          ;TAPE DATA DIRECTION OUT (0= IN)
000040                          NP.LOOP         = BIT5          ;ENABLE TRANSPORT LOOPBACK
000020                          NP.WRP          = BIT4          ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
                                ;*
                                ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
                                ;-

000200                          S2.DIM          = BIT7          ;WORD #9 BYTE 2 DATA IN MISS
000100                          S2.ILW          = BIT6          ;               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.

```
1340                                    .SBTTL  SPECIAL MACROS AND OPDEFS.
1341
1342
1343                                ;+
1344                                ;SAVE GENERAL REGS 1 TO 5
1345                                ;
1346
1347                                    .MACRO  SAVREG
1348                                    JSR     R5,REGSAV
1349                                    .ENDM
1350
1351                                ;+
1352                                ; MACRO TO FORCE AN ERROR
1353                                ;
1354                                .MACRO  FORCERROR       TAG,NOTSSR
1355                                .NLIST
1356                                .IIF NDF LISTALL, .NLIST
1357                                .LIST
1358                                .IF B NOTSSR
1359                                    MOV     TSSR(R5),R1     ;READ TSSR
1360                                .ENDC
1361                                    MOV     FORCER,FORCER   ;IS FORCER SET? (LEAVE C BIT ALONE)
1362                                    BNE     TAG             ;BR IF YES
1363                                .NLIST
1364                                .IIF NDF LISTALL, .LIST
1365                                .LIST
1366                                .ENDM
1367
1368                                ;+
1369                                ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
1370                                ;       WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
1371                                ;       SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
1372                                ;       FORCER TO 177777
1373                                ;       TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
1374                                ;-
1375                                .MACRO  FORCEXIT        TAG
1376                                .NLIST
1377                                .IIF NDF LISTALL, .NLIST
1378                                .LIST
1379                                    MOV     FORCER,FORCER   ;IS FORCER NEGATIVE?
1380                                    BMI     TAG             ;BR IF YES
1381                                .NLIST
1382                                .IIF NDF LISTALL, .LIST
1383                                .LIST
1384                                .ENDM
1385                                ;+
1386                                ; MACRO TO INCREMENT ERROR COUNTS
1387                                ;
1388                                .MACRO  NEXT.ERRNO
1389                                .NLIST
1390                                ;;;.IIF NDF LISTALL, .NLIST
1391                                    ERRNO=ERRNO+1
1392                                ;;;.IIF NDF LISTALL, .LIST
1393                                .LIST
                                    .ENDM
```

SPECIAL MACROS AND OPDEFS.

```
1395                                    ;*
1396                                    ;MACRO TO PERFORM XOR
1397                                    ;
1398
1399                                            .MACRO  XOR     A,B
1400                                            MOV     A,-(SP)
1401                                            BIC     B,(SP)
1402                                            BIC     A,B
1403                                            BIS     (SP)+,B
1404                                            .ENDM
1405
1406            000000                          EN=0                    ; INITIALIZE ERROR NUMBER
1407                                            .SBTTL  FORCER    FORCE ERROR FLAG
1408
1409                                    ;
1410                                    ; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
1411                                    ; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
1412                                    ;
1413
1414 002176  000000                     FORCER::        0       ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED
1415                                                    ;   BY THE MACRO "IFERROR").  AN ERROR NEED NOT
1416                                                    ; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.
```

```
                                        .SBTTL  GLOBAL DATA SECTION

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

                                ;
                                ;THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
                                ;SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.
                                ;
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 ITERATION 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                  REV::           .WORD   0       ;REVISION LEVEL
002232  000000                  BENBSW::        .WORD   0       ;BUFFER ENABLE SWITCH SW 0=OFF;1=ON
002234  000000                  EXPD::          .WORD   0       ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
002236  000000                  RECV::          .WORD   0       ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
002240  000000                  ERRHI::         .WORD   0       ;HIGH ADDRESS MEMORY ERROR
002242  000000                  ERRLO::         .WORD   0       ;LOW ADDRESS MEMORY ERROR
002244                          RAMDATA::       .BLKW   16.     ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
002304  000000                  RAMSIZ::        .WORD   0       ;RAM DATA SIZE FOR PRAMPKT ROUTINE
002306  000000                  RCVHIADD::      .WORD   0       ;RECEIVED BUFFER HIGH ADDRESS
002310  000000                  RCVLOADD::      .WORD   0       ;RECEIVED BUFFER LOW ADDRESS
002312  000000                  COUNT::         .WORD   0       ;TEST COUNT PATTERN
002314  000000                  DATA::          .WORD   0       ;TEST DATA
002316  000000                  TSTFLAG::       .WORD   0       ;TEST FLAG WORD
002320  000000                  TSTPTR::        .WORD   0       ;TSTBLK POINTER
002322  000000                  PRMNO::         .WORD   0       ;PRINT ROUTINE TEMP
002324                          EXPMSG::        .BLKB   100.    ;EXPECTED MESSAGE BUFFER DATA
002470                          RECMSG::        .BLKB   100.    ;RECEIVED MESSAGE BUFF R DATA
002634                          TMPBFR::        .BLKB   80.     ;TEMPORARY STORAGE FOR PRINT
```

N3


```
                                        .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
                                ;
                                ;

002754                          TSTBLK::
002754  000000                          .WORD   0               ;ALL ZEROS
002756  177777                          .WORD   177777          ;ALL ONES
002760  000001                          .WORD   BIT0            ;DATA FOR WALKING ONES
002762  000002                          .WORD   BIT1
002764  000004                          .WORD   BIT2
002766  000010                          .WORD   BIT3
002770  000020                          .WORD   BIT4
002772  000040                          .WORD   BIT5
002774  000100                          .WORD   BIT6
002776  000200                          .WORD   BIT7
003000  000400                          .WORD   BIT8
003002  001000                          .WORD   BIT9
003004  002000                          .WORD   BIT10
003006  004000                          .WORD   BIT11
003010  010000                          .WORD   BIT12
003012  020000                          .WORD   BIT13
003014  040000                          .WORD   BIT14
003016  100000                          .WORD   BIT15
003020  177776                          .WORD   ^CBIT0          ;DATA FOR WALKING ZEROS
003022  177775                          .WORD   ^CBIT1
003024  177773                          .WORD   ^CBIT2
003026  177767                          .WORD   ^CBIT3
003030  177757                          .WORD   ^CBIT4
003032  177737                          .WORD   ^CBIT5
003034  177677                          .WORD   ^CBIT6
003036  177577                          .WORD   ^CBIT7
003040  177377                          .WORD   ^CBIT8
003042  176777                          .WORD   ^CBIT9
003044  175777                          .WORD   ^CBIT10
003046  173777                          .WORD   ^CBIT11
003050  167777                          .WORD   ^CBIT12
003052  157777                          .WORD   ^CBIT13
003054  137777                          .WORD   ^CBIT14
003056  077777                          .WORD   ^CBIT15
003060  125252                          .WORD   125252          ;ALTERNATING ONES, ZEROS
003062  052525                          .WORD   052525          ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
003064                          TBLEND==.
```

```
                                        .SBTTL  GLOBAL ENVIRONMENT STORAGE
                                ;
                                ;STORAGE FOR DEVICE REGISTERS
                                ;
003064  000000  100000  000000  DUMMY:  0,100000,0,0     ;DUMMY DEVICE REGISTERS...
003074  000000  000000  000000          0,0,0,0,0,0,0,0
                                                        ;...FOR MULTI UNIT CHECKOUT.

003114  000000                  DUFLG::         .WORD   0               ;"DROPPED UNIT" FLAG.
                                                                        ;INHIBITS CODE IN "CLEAN UP".
003116  000000                  NODEV::         .WORD   0               ;FLAG TO SAY NO DEVICE.

003120  000000                  TEMP1::         .WORD   0               ;SOME TEMP LOCATIONS.
003122  000000                  TEMP2::         .WORD   0
003124  000000                  XXCOMM::        .WORD   0               ;XXDP+ COMM BLOCK POINTER.
003126  000000                  FREE::          .WORD   0               ;1ST FREE MEMORY ADDRESS...
003130  000000                  FRESIZ::        .WORD   0               ;...AND SIZE (IN WORDS).
003132  000000                  FREEHI: .WORD   0               ;LAST WORD IN FREE SPACE
003134  000000                  KTFLG::         .WORD   0               ;KT11, MEM AVAIL FLAG -
                                                                        ;- .WORD        0 = <24K OR NO KT -
                                                                        ;- NZ = >24K AND KT.
003136  000000                  KTENABLE::      .WORD   0               ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
003140  000000                  NXMFLG::        .WORD   0               ;SET IF WE CAN TEST CLEARED OTHERWISE
003142  000000                  NXMLO::         .WORD   0               ;NXM LO ADDRESS BITS
003144  000000                  NXMHI::         .WORD   0               ;NXM HI ADDRESS BITS FOR DAL'S 16-21
003146  000000                  T23A::          .WORD   0               ;11/23A FLAG
003150  000000                  T23B::          .WORD   0               ;11/23B FLAG
003152  000000                  T3BFLG::        .WORD   0               ;TEST 3B FLAG TO
003154  002000                  PST32W::        .WORD   2000            ;32W BLOCK ADDRESS FOR 32K START
003156  000000                  SIFLAG::        .WORD   0               ;
003160  000000                  BADDAT::        .WORD   0               ;ACTUAL DATA
003162  000000                  GDDAT::         .WORD   0               ;EXPECTED DATA
003164  000000                  LOOPFL::        .WORD   0
003166                          CTAB::                                  ;CONFIGURATION TABLES.
003166  000000                  CTABM::         .WORD   0               ;CONFIG WORK.
003170  000000                                  .WORD   0
003172  000000                                  .WORD   0
003174  000000                                  .WORD   0
003176  177777                                  .WORD   -1              ;END OF MEM TABLE.
003200                          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
                                ;
003200                          ERTABL:         .BLKW   64.
003400  000000                  ERTABE:         .WORD   0

003402  000000                  SKIPT:  .WORD   0               ;1=SKIP SUBTEST 0=NO SKIP OF SUBTEST
```

C4

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
                                ;
003404                                  DEVTYP  <TSV05>
003404                          L$DVTYP::
003404  124  123  126                    .ASCIZ  /TSV05/
                                        .EVEN

                                ;*
                                ;TEST DESCRIPTION
                                ;-
003412                                  DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****>
003412                          L$DESC::
003412  052  052  052                    .ASCIZ  /**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****/
                                        .EVEN

                                ;*
                                ;BIT TO ASCII CONVERSION FOR TSSR REGISTER
                                ;-
003504  003544  003547  003553  TSSRBIT::       .WORD   1$,2$,3$,4$,5$,6$,7$,8$
003524  003605  003611  003615                  .WORD   9$,10$,11$,12$,13$,14$,15$,16$
003544    123     103     000   1$:     .ASCIZ  'SC'
003547    102     111     105   2$:     .ASCIZ  'BIE'
003553    123     103     105   3$:     .ASCIZ  'SCE'
003557    122     115     122   4$:     .ASCIZ  'RMR'
003563    116     130     115   5$:     .ASCIZ  'NXM'
003567    116     102     101   6$:     .ASCIZ  'NBA'
003573    102     111     124   7$:     .ASCIZ  'BIT9'
003600    102     111     124   8$:     .ASCIZ  'BIT8'
003605    123     123     122   9$:     .ASCIZ  'SSR'
003611    117     106     114   10$:    .ASCIZ  'OFL'
003615    102     111     124   11$:    .ASCIZ  'BIT5'
003622    102     111     124   12$:    .ASCIZ  'BIT4'
003627    102     111     124   13$:    .ASCIZ  'BIT3'
003634    102     111     124   14$:    .ASCIZ  'BIT2'
003641    102     111     124   15$:    .ASCIZ  'BIT1'
003646    102     111     124   16$:    .ASCIZ  'BIT0'
                                        .EVEN
003654    124     123     123   SFIERR: .ASCIZ  'TSSR ERROR AFTER SOFT INIT'
003707    124     123     123   SFHERR: .ASCIZ  'TSSR ERROR AFTER BUS RESET'
003742    040     040     116   NXR:    .ASCIZ  /  NON EXISTANT DEVICE REGISTER/
004001    045     101     040   NXRX:   .ASCIZ  /%A  ADDRESS: %O6/
004022    045     101     040   TSSX:   .ASCII  /%A  TSBA,TSSR EXP'D: %O6%A,%O6%N/
004062    045     101     040           .ASCIZ  /%A  TSBA,TSSR REC'D: %O6%A,%O6/
004121    045     116     045   FUSI:   .ASCII  /%N%A/
004125    040     040     125   USI:    .ASCIZ  /  UNEXPECTED INTERRUPT/
004154    040     040     111   NSI:    .ASCIZ  /  INTERRUPT EXPECTED, NOT RECEIVED/
004217    045     116     045   FNOINTR:        .ASCII  /%N%A/
004223    040     040     116   NOINTR: .ASCIZ  /  NO INTERRUPT WAS GENERATED/
004260    040     040     111   IFAULT: .ASCIZ  /  INTERRUPT FAULT/
```

GLOBAL TEXT MESSAGES

```
1640 004302  045  101  040  INTX:    .ASCIZ  /%A  CPU PC: %O6%A  TSBA: %O6/
1641 004337  040  040  042  NOINIT:  .ASCIZ  /  "BUS INIT" DIDN'T INITIALIZE CONTROLLER/
1642 004411  040  040  042  NSINIT:  .ASCIZ  /  "SOFT INIT" DIDN'T INITIALIZE THE DPU/
1643 004461  040  040  042  BRINIT:  .ASCIZ  /  "BUS RESET" DIDN'T INITIALIZE THE DPU/
1644 004531  000            NUL:     .ASCIZ  //
1645 004532  045  116  000  NULCR:   .ASCIZ  /%N/
1646 004535  045  101  040  EXPGOT:  .ASCIZ  /%A EXP'D: %O6%A, REC'D: %O6/
1647 004571  045  116  045  EXPGT2:  .ASCIZ  /%N%A EXP'D: %O6%A, %O6%N%A REC'D: %O%A, %O6/
1648 004645  045  101  040  DUAD12:  .ASCIZ  /%A REG(W) WRITTEN TO: %O6%A REG(R) READ; EXP'D: %O6%A, REC'D: %O6/
1649 004747  122  101  115  PKTRAM::         .ASCIZ  'RAM Contents Do Not Match Packet Sent'
1650 005015  040  040  103  SCME:    .ASCIZ  /  CONFIG DOESN'T MATCH MFG. MASTER/
1651 005060  127  122  111  WRTMSG:  .ASCIZ  'WRITE CHARACTERISTICS Failed'
1652 005115  124  123  123  WRTERR:  .ASCIZ  'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
1653 005210  124  123  123  RDERR:   .ASCIZ  'TSSR Incorrect After READ Command, More Bits Set Than SSR'
1654 005302  106  101  124  SCHERR:  .ASCIZ  'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
1655 005374  105  122  122  RETERR:  .ASCIZ  'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
1656 005462  045  116  045  NOMEM:   .ASCIZ  '%N%A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. ******%N'
1657 005556  045  116  045  M8186:   .ASCIZ  '%N%A ***************** 11/23A SYSTEM *****************%N'
1658 005647  045  116  045  M8189:   .ASCIZ  '%N%A ***************** 11/23B SYSTEM *****************%N'
1659                          .EVEN
1660                          .SBTTL  GLOBAL ERROR REPORT SECTION
1661
1662                          ;++
1663                          ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
1664                          ; CALLS THAT ARE USED IN MORE THAN ONE TEST.
1665                          ; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
1666                          ;
1667 005740                          BGNMSG  NXRERR          ;NON-EXISTANT DEVICE REGISTER.
     005740                  NXRERR::
1668 005740                          PRINTX  #NXRX,NODEV     ;NODEV = NEXM ADDRESS.
     005740  013746  003116          MOV     NODEV,(SP)
     005744  012746  004001          MOV     #NXRX,(SP)
     005750  012746  000002          MOV     #2,-(SP)
     005754  010600                  MOV     SP,R0
     005756  104415                  TRAP    C$PNTX
     005760  062706  000006          ADD     #6,SP
1669 005764  004737  005772          JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
1670 005770                          ENDMSG
     005770                  L10002:
     005770  104423                  TRAP    C$MSG
1671                          ;
1672                          ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
1673                          ; TO ANY OF THE ABOVE ERROR SIGNATURES.
1674                          ;
1675 005772  005727          EXTEND: TST     (PC)+
1676 005774  000000          EXTA:   0                       ; 0 = NO EXTENSION.
1677 005776  001402                  BEQ     1$
1678 006000  004777  177770          JSR     PC,@EXTA        ; APPEND EXTENSION TEXT.
1679 006004                  1$:     PRINTX  #NULCR          ; PRINT A BLANK LINE
     006004  012746  004532          MOV     #NULCR,-(SP)
     006010  012746  000001          MOV     #1,-(SP)
     006014  010600                  MOV     SP,R0
     006016  104415                  TRAP    C$PNTX
     006020  062706  000004          ADD     #4,SP
1680 006024  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
                ;
                ;

006026          PRITSSR:
006026                  SAVREG                          ;SAVE GENERAL REGISTERS
006032  010104          MOV     R1,R4                   ;SAVE THE TSSR CONTENTS
006034                  PRINTB  #TSSRFOR,R4             ;PRINT THE CONTENTS OF TSSR
006034  010446          MOV     R4, (SP)
006036  012746  006417  MOV     #TSSRFOR,-(SP)
006042  012746  000002  MOV     #2, (SP)
006046  010600          MOV     SP,R0
006050  104414          TRAP    C$PNTB
006052  062706  000006  ADD     #6,SP
006056  010400          MOV     R4,R0                   ;GET TSSR BACK FOR CHKAMB
006060  004737  016104  JSR     PC,CHKAMB               ;ARE CONTENTS AMBIGUOUS ?
006064  103410          BCS     5$                      ;BRANCH IF NOT
006066                  PRINTX  #AMBTSSR                ;SHOW CONTENTS ARE AMBIGUOUS
006066  012746  006637  MOV     #AMBTSSR,-(SP)
006072  012746  000001  MOV     #1,-(SP)
006076  010600          MOV     SP,R0
006100  104415          TRAP    C$PNTX
006102  062706  000004  ADD     #4,SP
006106  010403  5$:     MOV     R4,R3                   ;CONTENTS OF TSSR
006110  042703  001476  BIC     #HIADDR!FATERR!TERCLS,R3        ;CLEAR ALL MULTIPLE BIT FIELDS
006114  001434          BEQ     20$                     ;NO BITS ARE SET
006116  012702  002634  MOV     #TMPBFR,R2              ;TEMPORARY ASCII BUFFER
006122  012701  003504  MOV     #TSSRBIT,R1             ;ASCII EQUIVALENT OF BITS
006126  005703  10$:    TST     R3                       REMAINING BITS TO CONVERT
006130  001413          BEQ     15$                     .BRANCH WHEN ALL ARE DONE
006132  000241          CLC                             ;CLEAR CARRY FOR SHIFT
006134  006103          ROL     R3                      ;SHIFT NEXT BIT TO CARRY
006136  103006          BCC     13$                     ;BRANCH IF BIT NOT SET
006140  011100          MOV     (R1),R0                 ;POINTER TO BIT DEFINITION
006142  112022  11$:    MOVB    (R0)+,(R2)+             ;MOVE ASCIZ TO BUFFER
006144  001376          BNE     11$                     ;MOVE ALL BITS
006146  112762  000054  177777  MOVB    #',,-1(R2)      ;INSERT A COMMA TO TERMINATE
006154  005721  13$:    TST     (R1)+                   ;POINT TO NEXT DESCRIPTION
006156  000763          BR      10$                     ;GET THE REMAINING BITS
006160  105042  15$:    CLRB    -(R2)                   ;TERMINATE THE LINE
006162                  PRINTX  #TSSDEF,#TMPBFR         ;PRINT THE BIT DEFINITIONS
006162  012746  002634  MOV     #TMPBFR, (SP)
006166  012746  006610  MOV     #TSSDEF,-(SP)
```

F4

PRITSSR - PRINT TSSR CONTENTS

```
                006172  012746  000002                  MOV     #2,-(SP)
                006176  010600                          MOV     SP,R0
                006200  104415                          TRAP    C$PNTX
                006202  062706  000006                  ADD     #6,SP
     1726
     1727       006206  010403              20$:        MOV     R4,R3           ;GET THE TSSR CONTENTS
     1728       006210  042703  177761                  BIC     #^CTERCLS,R3    ;CLEAR ALL BUT TERMINATION
     1729       006214  016303  006700                  MOV     TCOCOD(R3),R3   ;GET THE TERMINATION CODE MEANING
     1730       006220                                  PRINTX  #TCOASC,R3      ;PRINT THE TERMINATION CODE
                006220  010346                          MOV     R3,-(SP)
                006222  012746  006500                  MOV     #TCOASC,-(SP)
                006226  012746  000002                  MOV     #2,-(SP)
                006232  010600                          MOV     SP,R0
                006234  104415                          TRAP    C$PNTX
                006236  062706  000006                  ADD     #6,SP
     1731       006242  010403                          MOV     R4,R3           ;TSSR CONTENTS AGAIN
     1732       006244  042703  177717                  BIC     #^CFATERR,R3    ;CLEAR ALL BUT FATAL TERMINATION
     1733       006250  001416                          BEQ     25$             ;DON'T PRINT IF ZERO
     1734       006252  006203                          ASR     R3
     1735       006254  006203                          ASR     R3
     1736       006256  006203                          ASR     R3              ;ALINE TERMINATION CODE FOR INDEX
     1737       006260  016303  007240                  MOV     TSFCOD(R3),R3   ;GET THE FATAL TERMINATION CODE
     1738       006264                                  PRINTX  #TFCASC,R3      ;PRINT THE FATAL TERMINATION CODE
                006264  010346                          MOV     R3,-(SP)
                006266  012746  006541                  MOV     #TFCASC,-(SP)
                006272  012746  000002                  MOV     #2,-(SP)
                006276  010600                          MOV     SP,R0
                006300  104415                          TRAP    C$PNTX
                006302  062706  000006                  ADD     #6,SP
     1739       006306  042704  176377      25$:        BIC     #^CHIADDR,R4    ;CLEAR ALL BUT EXTENDED ADDRESS
     1740       006312  001411                          BEQ     30$             ;DON'T PRINT IF ZERO
     1741       006314                                  PRINTX  #TEXASC,R4      ;PRINT THE EXTENDED ADDRESS BITS
                006314  010446                          MOV     R4,-(SP)
                006316  012746  006437                  MOV     #TEXASC,-(SP)
                006322  012746  000002                  MOV     #2,-(SP)
                006326  010600                          MOV     SP,R0
                006330  104415                          TRAP    C$PNTX
                006332  062706  000006                  ADD     #6,SP
     1742       006336  013703  002200      30$:        MOV     EPRTSW,R3               ;PRINT MEAASGE BUFFER ADDRESS
     1743       006342                                  PRINTX  R3                      ;PRINT PROPER MESSAGE
                006342  010346                          MOV     R3,-(SP)
                006344  012746  000001                  MOV     #1,-(SP)
                006350  010600                          MOV     SP,R0
                006352  104415                          TRAP    C$PNTX
                006354  062706  000004                  ADD     #4,SP
     1744       006360  000207                          RTS     PC              ;RETURN TO CALLER
```

G4

PRITSSR - PRINT TSSR CONTENTS

```
   1751 006362                                  EPRT2:
   1752 006362     045     116     045  EPRT1:  .ASCIZ  '%N%A *****REPLACE M7196*****'
   1753
   1763 006417     045     116     045  TSSRFOR:        .ASCIZ  '%N%A TSSR = %O6'
   1764 006437     045     116     045  TEXASC:         .ASCIZ  '%N%A Extended Address Bits = %O6'
   1765 006500     045     116     045  TCOASC:         .ASCIZ  '%N%A Termination Class Code = %T'
   1766 006541     045     116     045  TFCASC:         .ASCIZ  '%N%A Fatal Termination Class Code = %T'
   1767 006610     045     116     045  TSSDEF:         .ASCIZ  '%N%A TSSR Bits Set: %T'
   1768 006637     045     116     045  AMBTSSR:        .ASCIZ  '%N%A TSSR Contents Are Amb'guous'
   1769                                          .EVEN
   1770 006700  006720  006743  006771  TCOCOD: .WORD   1$,2$,3$,4$,5$,6$,7$,8$
   1771 006720     116     157     162  1$:     .ASCIZ  'Normal Termination'
   1772 006743     124     145     162  2$:     .ASCIZ  'Termination Condition'
   1773 006771     124     141     160  3$:     .ASCIZ  'Tape Status Alert'
   1774 007013     106     165     156  4$:     .ASCIZ  'Funct'on Reject'
   1775 007033     122     145     143  5$:     .ASCIZ  'Recoverable Error   Tape Position One Record Down'
   1776 007115     122     145     143  6$:     .ASCIZ  'Recoverable Error   Tape Was Not "oved'
   1777 007164     125     156     162  7$:     .ASCIZ  'Unrecoverable Error'
   1778 007210     106     141     164  8$:     .ASCIZ  'Fatal Controller Error'
   1779                                          .EVEN
   1780
   1781 007240  007250  007304  007315  TSFCOD: .WORD   1$,2$,3$,4$
   1782 007250     111     156     164  1$:     .ASCIZ  'Internal Diagnostic Failure'
   1783 007304     122     145     163  2$:     .ASCIZ  'Reserved'
   1784 007315     102     165     163  3$:     .ASCIZ  'Bus Interface or Sanity Check Error'
   1785 007361     122     145     163  4$:     .ASCIZ  'Reserved'
   1786                                          .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.
                ;

007372                  PRIPKT::
007372                          SAVREG                  ;SAVE THE REGISTERS
007376  010005                  MOV     R0,R5           ;SAVE NO. OF WORDS IN PACKET
007400  005737  003136          TST     KTENABLE        ;ABOVE 28K UNDER TEST?
007404  001001                  BNE     10$             ;BR IF YES
007406  005003                  CLR     R3              ;SET HIGH ORDER ADDRESS TO 0
007410  010301          10$:    MOV     R3,R1           ;COPY HIGH ORDER ADDRESS
007412  010400                  MOV     R4,R0           ;GET LOWER ADDRESS
007414  006100                  ROL     R0              ;SHIFT BIT 15 INTO C BIT
007416  006101                  ROL     R1              ;AND INTO HIGH ORDER.
007420                          PRINTB  #PKTADD,R1,R4   ;PRINT PACKET ADDRESS
007420  010446                  MOV     R4,-(SP)
007422  010146                  MOV     R1,-(SP)
007424  012746  007556          MOV     #PKTADD,-(SP)
007430  012746  000003          MOV     #3,-(SP)
007434  010600                  MOV     SP,R0
007436  104414                  TRAP    C$PNTB
007440  062706  000010          ADD     #10,SP
007444  010300          15$:    MOV     R3,R0           ;GET HIGH ORDER ADDRESS
007446  001404                  BEQ     20$             ;BR IF NOT ABOVE 28K.
007450  010401                  MOV     R4,R1           ;GET LOW ORDER ADDRESS
007452  004737  017356          JSR     PC,SETMAP       ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
007456  010004                  MOV     R0,R4           ;GET RETURNED PAR6 ADDRESS BIAS
007460  005001          20$:    CLR     R1              ;SAVE WORD NUMBER
007462  012402          25$:    MOV     (R4)+,R2        ;GET PACKET CONTENTS
007464                          PRINTB  #PKTFRM,R1,R2   ;PRINT THE DATA
007464  010246                  MOV     R2,-(SP)
007466  010146                  MOV     R1,-(SP)
007470  012746  007520          MOV     #PKTFRM,-(SP)
007474  012746  000003          MOV     #3,-(SP)
007500  010600                  MOV     SP,R0
007502  104414                  TRAP    C$PNTB
007504  062706  000010          ADD     #10,SP
007510  005201                  INC     R1              ;NEXT WORD NUMBER
007512  020105                  CMP     R1,R5           ;DONE ALL PACKET WORDS?
007514  002762                  BLT     25$             ;LOOP TILL ALL DONE
007516  000207                  RTS     PC              ;RETURN

007520  045  116  045  PKTFRM: .ASCIZ  '%N%A Packet Word #%D1%A = %06'
007556  045  116  045  PKTADD: .ASCIZ  '%N%A Packet Address = %01%05'
                        .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
                                ;
007614                          PRIBXOR::
007614                                  SAVREG                          ;SAVE THE REGISTERS
007620  010203                          MOV     R2,R3                   ;EXPECTED DATA
007622                                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007632  012700  177400                  MOV     #^C<377>,R0             ;BYTE MASK
007636  040001                          BIC     R0,R1                   ;SAVE LOW BYTE RECV
007640  040002                          BIC     R0,R2                   ;SAVE LOW BYTE EXPD
007642  040003                          BIC     R0,R3                   ;SAVE LOW BYTE XOR
007644                                  PRINTB  #XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
007644  010346                          MOV     R3, (SP)
007646  010146                          MOV     R1, (SP)
007650  010246                          MOV     R2, (SP)
007652  012746  007676                  MOV     #XORBFOR, (SP)
007656  012746  000004                  MOV     #4,-(SP)
007662  010600                          MOV     SP,R0
007664  104414                          TRAP    C$PNTB
007666  062706  000012                  ADD     #12,SP
007672  010300                          MOV     R3,R0                   ;R0 HAS XOR ON RETURN
007674  000207                          RTS     PC                      ;RETURN TO CALLER

007676      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
                                ;
007744                          PRIXOR::
007744                                  SAVREG                          ;SAVE THE REGISTERS
007750  010203                          MOV     R2,R3                   ;EXPECTED DATA
007752                                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007762                                  PRINTB  #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
```

J4


```
PRIXOR    PRINT EXPD, RECV AND XOR

        007762  010346                          MOV     R3, (SP)
        007764  010146                          MOV     R1, (SP)
        007766  010246                          MOV     R2, (SP)
        007770  012746  010014                  MOV     #XORFOR, (SP)
        007774  012746  000004                  MOV     #4, (SP)
        010000  010600                          MOV     SP,R0
        010002  104414                          TRAP    C$PNTB
        010004  062706  000012                  ADD     #12,SP
  1880  010010  010300                          MOV     R3,R0           ;R0 HAS XOR ON RETURN
  1881  010012  000207                          RTS     PC              ;RETURN TO CALLER
  1882
  1883  010014      045     116     045  XORFOR: .ASCIZ   %N%A EXPD: %06%A RECV: %06%A XOR: %06
  1884                                          .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
                                ;
                                ;-

010062                          PRIEQU:
010062                                  SAVREG                          ;SAVE THE REGISTERS
010066  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
                                ;
                                ;-
010070                          PRIRAM:
010070                                  SAVREG                  ;SAVE R1 R5 UNTIL NEXT RETURN
010074                                  PRINTB  @RAMFOR,R4      ;PRINT RAM ADDRESS IN ERROR
010074  010446                          MOV     R4,-(SP)
010076  012746  010120                  MOV     @RAMFOR,-(SP)
010102  012746  000002                  MOV     #2, (SP)
010106  010600                          MOV     SP,R0
010110  104414                          TRAP    C$PNTB
010112  062706  000006                  ADD     #6,SP
010116  000207                          RTS     PC              ;RETURN

010120      045     116     045 RAMFOR: .ASCIZ  '%N%A CONTROLLER RAM ADDRESS = %O6'
                                        .EVEN
```

```
PRIADD    PRINT MEMORY ERROR ADDRESS

                                        .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
                                ;
                                ;-
010162                          PRIADD:
010162                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
010166  013700  002240                  MOV     ERRHI,R0                ;GET HIGH ADDRESSS
010172  013701  002242                  MOV     ERRLO,R1                ;GET LOW ADDRESS
010176  010102                          MOV     R1,R2                   ;COPY LOW ADDRESS
010200  006101                          ROL     R1                      ;SHIFT BIT 15 TO C BIT
010202  006100                          ROL     R0                      ;SHIFT INTO HIGH ORDER
010204                                  PRINTB  #PRIA0,R0,R2            ;PRINT MEMORY ADDRESS IN ERROR
010204  010246                          MOV     R2,-(SP)
010206  010046                          MOV     R0, (SP)
010210  012746  010232                  MOV     #PRIA0,-(SP)
010214  012746  000003                  MOV     #3,-(SP)
010220  010600                          MOV     SP,R0
010222  104414                          TRAP    C$PNTB
010224  062706  000010                  ADD     #10,SP
010230  000207                          RTS     PC                      ;RETURN

010232      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
                                ;
                                ;
010276                          PRITADD:
010276                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
010302  013702  002240                  MOV     ERRHI,R2                ;GET HIGH ADDRESSS
010306  013701  002242                  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
010312                                  PRINTB  #PRIT0,R1               ;PRINT MEMORY ADDRESS LOW IN ERROR
010312  010146                          MOV     R1,-(SP)
010314  012746  010360                  MOV     #PRIT0,-(SP)
010320  012746  000002                  MOV     #2,-(SP)
010324  010600                          MOV     SP,R0
010326  104414                          TRAP    C$PNTB
```

M4


```
PRITADD   PRINT MEMORY TEST ADDRESS

        010330  062706  000006                  ADD     #6,SP
   1968 010334                                  PRINTB  #PRIT1,R2       ;PRINT MEMORY ADDRESS HIGH IN ERROR
        010334  010246                          MOV     R2,-(SP)
        010336  012746  010423                  MOV     #PRIT1,-(SP)
        010342  012746  000002                  MOV     #2, (SP)
        010346  010600                          MOV     SP,R0
        010350  104414                          TRAP    C$PNTB
        010352  062706  000006                  ADD     #6,SP
   1969 010356  000207                          RTS     PC              ;RETURN
   1970
   1971 010360     045     116     045  PRIT0:  .ASCIZ  '<N><A MEMORY TEST ADDRESS LOW = <06>'
   1972 010423     045     116     045  PRIT1:  .ASCIZ  '<N><A MEMORY TEST ADDRESS HIGH = <06>'
   1973                                         .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:
                ;
                ;
                ;-

010470                  SPACE::
010470                          SAVREG                          ;SAVE THE GENERAL REGISTERS
010474  012737  000764  010660        MOV     #500.,SDELAY    ;SET UP DELAY
010502  012737  140010  010650        MOV     #140010,80$     ;SET UP COMMAND, SPACE FORWARD
010510  005703                        TST     R3              ;CHECK FOR DIRECTION
010512  100403                        BMI     5$              ;BR, IF REVERSE INDICATED
010514  010337  010652                MOV     R3,90$          ;LOAD UP NUMBER OF RECORDS TO SPACE
010520  000407                        BR      10$             ;GO DO COMMAND
010522  042703  100000        5$:     BIC     #BIT15,R3       ;CLEAR DIRECTION BIT
010526  010337  010652                MOV     R3,90$          ;LOAD UP NUMBER OF RECORDS TO SPACE
010532  052737  000400  010650        BIS     #BIT8,80$       ;SET REVERSE BIT IN COMMAND PACKET
010540  012704  010650        10$:    MOV     #80$,R4         ;SET UP R4 WITH PACKET ADDRESS
010544  010465  000000                MOV     R4,TSDB(R5)     ;SEND OUT COMMAND
010550  004737  016310        15$:    JSR     PC,WAITF        ;WAIT FOR SSR
010554  103420                        BCS     20$             ;BR, IF SSR IS SET AND OK
010556                                  DELAY   250             ;DELAY ABOUT .25 SECONDS
010556  012727  000250                MOV     #250.,(PC)+
010562  000000                        .WORD   0
010564  013727  002116                MOV     L$DLY,(PC)+
010570  000000                        .WORD   0
010572  005367  177772                DEC     -6(PC)
010576  001375                        BNE     .-4
```

B5

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

```
        010600  005367  177756                  DEC     -22(PC)
        010604  001367                          BNE     . 20
   2026 010606  005337  010660                  DEC     SDELAY                  ;BUMP DELAY COUNTER DOWN
   2027 010612  001356                          BNE     15$                     ;BR, IF MORE DELAY
   2028 010614  000411                          BR      60$                     ;BR IF TROUBLE CARRY = CLEAR
   2029 010616  016501  000002          20$:    MOV     TSSR(R5),R1             ;READ TSSR
   2030 010622  012702  000200                  MOV     #SSR,R2                 ;SET UP EXPECTED
   2031 010626  020201                  25$:    CMP     R2,R1                   ;ARE THEY OK
   2032 010630  001401                          BEQ     40$                     ;BR, IF EQUAL = OK
   2033 010632  000402                          BR      60$                     ;TROUBLE EXIT
   2034 010634  000261                  40$:    SEC                             ;SET CARRY NO TROUBLE
   2035 010636  000401                          BR      70$                     ;EXIT
   2036 010640  000241                  60$:    CLC                             ;CARRY CLEAR = ERROR
   2037 010642                          70$:
   2038 010642  010400                          MOV     R4,R0                   ;PASS PACKET ADDRESS
   2039 010644  000207                          RTS     PC                      ;RETURN
   2040                                 ;
   2041                                 ;
   2042                                 ;
   2043                                 ;PACKET FOR SPACE COMMAND
   2044                                 ;
   2046         010650                          .=<.+10>&177770
   2048                                 ;
   2049                                 ;COMMAND WORD
   2050 010650  000000                  80$:    .WORD
   2051                                 ;NUMBER OF RECORDS TO BE SPACED OVER WORD
   2052 010652  000000                  90$:    .WORD
   2053 010654  000000                          .WORD
   2054 010656  000000                          .WORD
   2055 010660  000000                  SDELAY: .WORD   0                       ;DELAY COUNTER
   2056                                         .EVEN
   2057                                         .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  002232                          CLR     BENBSW          ;CLEAR BUFFER ENABLE SWITCH
010672  005037  002226                          CLR     EXTFEA                  ;CLEAR EXTENDED FEATURES SW SWITCH
010676  010465  000000                  10$:    MOV     R4,TSD8(R5)             ;SEND OUT COMMAND
010702  004737  016376                          JSR     PC,CHKTSSR              ;WAIT FOR SSR
010706  103401                                  BCS     20$                     ;BR, IF SSR IS SET AND OK
010710  000455                                  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  000441                                  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  002232                          INC     BENBSW                  ;SET SOFTWARE SWITCH FOR ENABLED
011000                                  50$:
011000  016337  000012  002230                  MOV     XST2(R3),REV            ;MICROCODE REV LEVEL
011006  042737  017700  002230                  BIC     #17700,REV              ;CLEAR UNWANTED BITS
011014  022737  000001  002230                  CMP     #1,REV                  ;IS IT A NEW MICROCODE
011022  001406                                  BEQ     55$                     ;NO BR
011024  012737  000001  002226                  MOV     #1,EXTFEA               ;ALWAY EXTENDED FEATURE FOR NEW
                                                                                ;MICROCODE
011032  052763  000200  000012                  BIS     #X2.EXTF,XST2(R3)       ;EXTENDED FEATURE ALWAYS SET IN
                                                                                ;MICROCODE
```

D5

SEQ 0055

WRTCHR WRITE CHARACTERISTICS COMMAND

```
 2116 011040  000261                55$:    SEC                     ;SET CARRY NO TROUBLE
 2117 011042  000401                        BR      70$             ;EXIT
 2118 011044  000241                60$:    CLC                     ;CARRY CLEAR = ERROR
 2119 011046  016500  000002        70$:    MOV     TSSR(R5),R0     ;RETURN TSSR CONTENTS
 2120 011052  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
                                ;
                                ;
                                ;
011054                          REWIND::
011054                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011060  012704  011150                  MOV     @RWPACK,R4              ;GET PACKET ADDRESS
011064  010465  000000                  MOV     R4,TSDB(R5)             ;SEND PACKET ADDRESS TO EXECUTE
011070  012703  000550                  MOV     @360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
011074  004737  016310          10$:    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
011100  103417                          BCS     20$                     ;LEAVE WHEN SSR IS SET
011102                                  DELAY   250.                    ;WAIT FOR .25 SECONDS
011102  012727  000372                  MOV     @250.,(PC)+
011106  000000                          .WORD   0
011110  013727  002116                  MOV     L$DLY,(PC)+
011114  000000                          .WORD   0
011116  005367  177772                  DEC     -6(PC)
011122  001375                          BNE     .-4
011124  005367  177756                  DEC      22(PC)
011130  001367                          BNE     .-20
011132  005303                          DEC     R3                      ;BUMP COUNTER DOWN
011134  001357                          BNE     10$                     ;KEEP GOING
011136  000241                          CLC                             ;CLEAR CARRY TO SET ERROR
011140  010400          20$:    MOV     R4,R0                           ;PASS THE PACKET ADDRESS
011142  000207                  RTS     PC                              ;RETURN
                                
011150                                  .=<.+10>&177770
011150                          RWPACK:
011150  102010                          .WORD   102010                  ;POSTION COMMAND (REWIND)
011152  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
                                ;
                                ;-

011154                          CKRAM::
011154                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
011160  012701  002244                  MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
011164  012702  000201                  MOV     #RMPKTBEG,R2            ;BYTE ADDRESS OF FIRST RAM DATA
011170  005003                          CLR     R3                      ;CLEAR THE ERROR FLAG
011172  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
011176  112765  000000  000000          MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
011204  004737  016376          10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011210  010265  000000                  MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
011214  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011220  116511  000000                  MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
011224  122124                          CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
011226  001401                          BEQ     20$                     ;BRANCH IF OK
011230  005203                          INC     R3                      ;SET ERROR FLAG
011232  005202                  20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
011234  020227  000210                  CMP     R2,#RMPKTEND            ;REACHED END YET ;
011240  003761                          BLE     10$                     ;BRANCH TILL ALL READ
011242  005703                          TST     R3                      ;WAS AN ERROR FOUND ?
011244  001402                          BEQ     30$                     ;BRANCH IF NOT
011246  000241                          CLC                             ;CLEAR CARRY TO SHOW ERROR
011250  000401                          BR      50$                     ;AND EXIT
011252  000261                  30$:    SEC                             ;SHOW GOOD COMPARE
011254  012737  000010  002304  50$:    MOV     #8.,RAMSIZ              ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
011262  000207                          RTS     PC                      ;RETURN
```

CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA

```
                                        .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
                                ;
011264                          CKRAM2::
011264                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
011270  012701  002244                  MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
011274  012702  000167                  MOV     #RMCHBEG,R2             ;BYTE ADDRESS OF FIRST RAM DATA
011300  005003                          CLR     R3                      ;CLEAR THE ERROR FLAG
011302  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
011306  112765  000000  000000          MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
011314  004737  016376          10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011320  010265  000000                  MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
011324  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011330  116511  000000                  MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
011334  122124                          CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
011336  001401                          BEQ     20$                     ;BRANCH IF OK
011340  005203                          INC     R3                      ;SET ERROR FLAG
011342  005202                  20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
011344  012737  000010  002304          MOV     #8.,RAMSIZ              ;ASSUME EXTFEA NOT SET
011352  005737  002226                  TST     EXTFEA                  ;IS THE SOFTWARE EXTENDED FEATURES SET
011356  001407                          BEQ     25$                     ;BR, IF NOT SET
011360  012737  000012  002304          MOV     #10.,RAMSIZ             ;SET RAMSIZ FOR EXTEND FEATURES
011366  020227  000200                  CMP     R2,#RMCHEND             ;AT END OF EXTENDED BUFFER
011372  003750                          BLE     10$                     ;BR, IF NOT AT END YET
011374  000403                          BR      27$                     ;AT END BRANCH
011376  020227  000176          25$:    CMP     R2,#RMCHEND-2           ;REACHED END YET ?
011402  003744                          BLE     10$                     ;BRANCH TILL ALL READ
011404  005703                  27$:    TST     R3                      ;WAS AN ERROR FOUND ?
011406  001402                          BEQ     30$                     ;BRANCH IF NOT
011410  000241                          CLC                             ;CLEAR CARRY TO SHOW ERROR
011412  000401                          BR      50$                     ;AND EXIT
011414  000261                  30$:    SEC                             ;SHOW GOOD COMPARE
011416  000207                  50$:    RTS     PC                      ;RETURN
```

CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS

```
                                        .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
                                ;
                                ;-
011420                          CKMSG::
011420                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011424  010037  002306                  MOV     R0,RCVHIADD             ;SAVE RECV HIGH ADDRESS
011430  010137  002310                  MOV     R1,RCVLOAD              ;SAVE RECV LOW ADDRESS
011434  005737  003136                  TST     KTENABLE                ;TESTING ABOVE 28K?
011440  001403                          BEQ     10$                     ;BR IF NO
011442  004737  017356                  JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
011446  010001                          MOV     R0,R1                   ;GET RETURNED ADDRESS BIASED TO PAR6
011450  005004                  10$:    CLR     R4                      ;WORD IN BUFFER
011452  005003                          CLR     R3                      ;CLEAR ERROR SEEN FLAG
011454  010205                          MOV     R2,R5                   ;GET EXPD BUFFER ADDRESS
011456  011264  002324          15$:    MOV     (R2),EXPMSG(R4)         ;SAVE EXPD FOR ERROR REPORT
011462  011164  002470                  MOV     (R1),RECMSG(R4)         ;SAVE RECV FOR ERROR REPORT
011466  022221                          CMP     (R2)+,(R1)+             ;EXPD EQUAL RECV?
011470  001401                          BEQ     25$                     ;BR IF YES
011472  005203                          INC     R3                      ;SET ERROR SEEN FLAG
011474  062704  000002          25$:    ADD     #2,R4                   ;POINT TO NEXT WORD ADDRESS
011500  020427  000014                  CMP     R4,#14                  ;DONE FIRST 7 WORDS?
011504  003764                          BLE     15$                     ;BR IF NO
011506  032765  000200  000012          BIT     #X2.EXTF,XST2(R5);IS EXTENDED FEATURES SET IN EXPD?
011514  001403                          BEQ     50$                     ;BR IF NO
011516  020427  000016                  CMP     R4,#16                  ;DONE EXTENDED FEATURES WORD?
011522  003755                          BLE     15$                     ;BR IF NO
011524  005703                  50$:    TST     R3                      ;ANY ERRORS SEEN?
011526  001402                          BEQ     55$                      BR IF NO
011530  000241                          CLC                             ;SET FAILURE
011532  000401                          BR      60$                     ;
011534  000261                  55$:    SEC                             ;SET SUCCESS
011536  000207                  60$:    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
                                ;
011540                          CKMSG2::
011540                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011544  020327  000144                  CMP     R3,@RECMSG-EXPMSG;@@D IS COUNT ABOVE MAX ALLOWED?
011550  003412                          BLE     5$                      ;@@D BR IF NO
011552  012703  000144                  MOV     @RECMSG EXPMSG,R3;@@D
011556                                  PRINTF  @DEBUGMSG               ;@@D
011556  012746  011672                  MOV     @DEBUGMSG,-(SP)
011562  012746  000001                  MOV     @1, (SP)
011566  010600                          MOV     SP,R0
011570  104417                          TRAP    C$PNTF
011572  062706  000004                  ADD     @4,SP
011576  010037  002306          5$:     MOV     R0,RCVHIADD     ;SAVE RECV HIGH ADDRESS
011602  010137  002310                  MOV     R1,RCVLOAD      ;SAVE RECV LOW ADDRESS
011606  005737  003136                  TST     KTENABLE        ;TESTING ABOVE 28K?
011612  001403                          BEQ     10$             ;BR IF NO
011614  004737  017356                  JSR     PC,SETMAP       ;RETURN ADDRESS BIASED TO PAR6 IN R0
011620  010001                          MOV     R0,R1           ;GET RETURNED ADDRESS BIASED TO PAR6
011622  005004          10$:    CLR     R4              ;WORD IN BUFFER
011624  005005                  CLR     R5              ;CLEAR ERROR SEEN FLAG
011626  111264  002324  15$:    MOVB    (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
011632  111164  002470          MOVB    (R1),RECMSG(R4) ;SAVE RECV FOR ERROR REPORT
011636  122221                  CMPB    (R2)+,(R1)+     ;EXPD EQUAL RECV?
011640  001401                  BEQ     25$             ;BR IF YES
011642  005205                  INC     R5              ;SET ERROR SEEN FLAG
011644  062704  000001  25$:    ADD     @1,R4           ;POINT TO NEXT BYTE
011650  020403                  CMP     R4,R3           ;DONE ALL BYTES?
011652  002001                  BGE     50$             ;BR IF YES
011654  000764                  BR      15$             ;DO NEXT BYTE
011656  005705          50$:    TST     R5              ;ANY ERRORS SEEN?
011660  001402                  BEQ     55$             ;BR IF NO
011662  000241                  CLC                     ;SET FAILURE
011664  000401                  BR      60$             ;
011666  000261          55$:    SEC                     ;SET SUCCESS
011670  000207          60$:    RTS     PC              ;RETURN
```

CKMSG2 COMPARE EXPD RECV MESSAGE BUFFERS

```
2387 011672  120  122  117  DEBUGMSG: .ASCIZ 'PROGRAM INTERNAL ERROR  CKMSG2 MESSAGE BUFFER EXCEEDED ';a@0
2388 011762  045  116  045  FERCM:   .ASCII  /%N%A  ***/
2389 011773  040  040  124  ERCM:    .ASCIZ  /  TSSR ERROR CODE REC'D = /
2390 012026  056  056  056  SIMSG:   .ASCIZ  /.... AFTER DOING SOFT INIT/
2391 012061  124  105  123  TINERR:  .ASCIZ  /TEST: .../
2392                          .EVEN
2393                         ;+
2394                         ;
2395                         ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
2396                         ;
2397                         ;INPUT:
2398                         ;
2399                         ;       R1      CONTENTS OF TSSR AT ERROR
2400                         ;
2401                         ;SIDE EFFECTS:
2402                         ;
2403                         ;       EXECUTES DROP UNIT TO CEASE TESTING
2404                         ;
2405                         ;-
2406
2407 012074                          BGNMSG  SFIMSG
     012074                  SFIMSG::
2408 012074  004737  006026          JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
2409 012100  004737  017242          JSR     PC,CKDROP       ;DROP UNIT, IF ALLOWED
2410 012104                          ENDMSG
     012104                  L10003:
     012104  104423                 TRAP    C$MSG
2411
2412                         ;+
2413                         ;PRINT ROUTINE TO PRINT THE CONTENTS OF
2414                         ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
2415                         ;
2416                         ;INPUTS:
2417                         ;
2418                         ;       R1      TSSR CONTENTS
2419                         ;       R4      ADDRESS OF COMMAND PACKET
2420                         ;
2421                         ;
2422
2423 012106                          BGNMSG  PKTSSR
     012106                  PKTSSR::
2424 012106  004737  006026          JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
2425 012112  012700  000004          MOV     #4,R0           ;NO. OF WORDS IN PACKET
2426 012116  004737  007372          JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
2427 012122                          ENDMSG
     012122                  L10004:
     012122  104423                 TRAP    C$MSG
```

```
                                        ;*
                                        ;PRINT ROUTINE TO PRINT THE CONTENTS OF
                                        ;TSSR AND A GET STATUS COMMAND PACKET.
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      TSSR CONTENTS
                                        ;       R4      ADDRESS OF COMMAND PACKET
                                        ;
                                                BGNMSG  PKTGETS
                                        PKTGETS::
                                                JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
                                                MOV     #2,R0           ;NO. OF WORDS IN GET STATUS PACKET
                                                JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
                                                ENDMSG
                                        L10005:
                                                TRAP    C$MSG
                                        ;*
                                        ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
                                        ;
                                        ;INPUTS:
                                        ;       R1      TSSR CONTENTS
                                        ;       R4      ADDRESS OF COMMAND PACKET
                                        ;
                                                BGNMSG  SFFMSG
                                        SFFMSG::
                                                JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
                                                ENDMSG
                                        L10006:
                                                TRAP    C$MSG
                                                .SBTTL  PKTMES  PRINT TSSR AND MESSAGE BUFFER
                                        ;*
                                        ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
                                        ;BUFFER FOR ERROR REPORTS
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      CONTENTS OF TSSR
                                        ;       R2      LOW ORDER MESSAGE BUFFER
                                        ;       R3      HIGH ORDER MESSAGE BUFFER ADDRESS
                                        ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
                                        ;
                                                BGNMSG  PKTMES
                                        PKTMES::
                                                JSR     PC,PRITSSR              ;PRINT CONTENTS OF TSSR
                                                MOV     R2,R0                   ;LOW ORDER ADDRESS
                                                MOV     R3,R1                   ;HIGH ORDER ADDRESS
                                                JSR     PC,PRMESS               ;PRINT THE MESSAGE BUFFER
                                                ENDMSG
                                        L10007:
                                                TRAP    C$MSG
```

```
                                        .SBTTL  ADDSSR    PRINT TEST ADDRESS AND TSSR
                                ;+
                                ;PRINT ROUTINE TO PRINT THE CONTENTS OF
                                ;TSSR AND A MEMORY TEST ADDRESS
                                ;
                                ;INPUTS:
                                ;
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;       ERRHI   HIGH ORDER MEMORY TEST ADDRESS
                                ;       ERRLO   LOW ORDER MEMORY TEST ADDRESS
                                ;-
012166                                  BGNMSG  ADDSSR
012166                          ADDSSR::
012166  004737  010276                  JSR     PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
012172  016501  000002                  MOV     TSSR(R5),R1     ;GET CURRENT TSSR
012176  004737  006026                  JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
012202                                  ENDMSG
012202                          L10010:
012202  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
                                ;
012204                                  BGNMSG  MSGEXP
012204                          MSGE ?::
012204  012700  000007                  MOV     #7,R0           ;ASSUME NO EXT FEATURES
012210  005737  002226                  TST     EXTFEA          ;EXT FEATURES SET?
012214  001402                          BEQ     5$              ;BR IF NO
012216  012700  000010                  MOV     #8.,R0          ;EXT FEATURE BUFFER IS 8 WORDS
012222  004737  014612          5$:     JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012226                                  ENDMSG
012226                          L10011:
012226  104423                          TRAP    C$MSG
```

M5


```
                                        .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
                                FIFEXP::
                                        PRINTX  #FIF1MSG,R1     ;PRINT BYTES TRANSFERRED
                                        MOV     R1,-(SP)
                                        MOV     #FIF1MSG,-(SP)
                                        MOV     #2,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTX
                                        ADD     #6,SP
                                        PRINTX  #FIF2MSG        ;PRINT HEADER MSG
                                        MOV     #FIF2MSG,-(SP)
                                        MOV     #1,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTX
                                        ADD     #4,SP
                                        MOV     R1,R0           ;GET BYTE COUNT
                                        JSR     PC,PRBYTEXP     ;PRINT FIFO BYTES IN ERROR
                                        ENDMSG
                                L10012:
                                        TRAP    C$MSG
012302  045  116  045  FIF1MSG:         .ASCIZ  '%N%A NUMBER OF BYTES TRANSFERRED = %D2'
012351  045  116  045  FIF2MSG:         .ASCIZ  '%N%A FIFO DATA BYTES IN ERROR:'
                                        .EVEN
```

```
                                        .SBTTL  MSGSTAT   PRINT STATUS HEADER AND MESSAGE BUFFERS
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
                                ;
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
012410                                  BGNMSG  MSGSTAT
012410                          MSGSTAT::
012410  012701  012452                   MOV     @STATCOD,R1     ;ASCII ADDRESS TABLE
012414  012100                  10$:     MOV     (R1)+,R0        ;DONE ALL MSG LINES?
012416  001410                           BEQ     20$             ;BR IF YES
012420                                   PRINTX  R0              ;PRINT STATUS BIT NAMES
012420  010046                           MOV     R0,-(SP)
012422  012746  000001                   MOV     #1,-(SP)
012426  010600                           MOV     SP,R0
012430  104415                           TRAP    C$PNTX
012432  062706  000004                   ADD     #4,SP
012436  000766                           BR      10$             ;DO ANOTHER MSG LINE
012440  012700  000012          20$:     MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
012444  004737  014612                   JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012450                                   ENDMSG
012450                          L10013:
012450  104423                           TRAP    C$MSG

012452  012470  012532  012623  STATCOD:         .WORD   1$,2$,3$,4$,5$,6$,0
012470     045     116     045  1$:.ASCIZ  '<N><A Tape Bus Signals in Word #8:'
012532     045     116     045  2$:.ASCIZ  '<N><A         PARERR<15>  IEOT  <12>  IFMK  <9>  IRDY<6>  IRWD<2>'
012623     045     116     045  3$:.ASCIZ  '<N><A         IRESV2<14>  IIDENT<11>  IHER  <8>  IONL<5>  IFBY<1>'
012714     045     116     045  4$:.ASCIZ  '<N><A         IRESV1<13>  ICER  <10>  ISPEED<7>  ILDP<4>  IFPT<0>'
013005     045     116     045  5$:.ASCIZ  '<N><A Tape Bus Signals in Word #9:'
013047     045     116     045  6$:.ASCIZ  '<N><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
                                        ;
                                                BGNMSG  MSGLOOP
013124                                  MSGLOOP::
013124  012701  013166                          MOV     #LOOPCOD,R1     ;ASCII ADDRESS TABLE
013130  012100                          10$:    MOV     (R1)+,R0        ;DONE ALL MSG LINES?
013132  001410                                  BEQ     20$             ;BR IF YES
013134                                          PRINTX  R0              ;PRINT STATUS BIT NAMES
013134  010046                                  MOV     R0,-(SP)
013136  012746  000001                          MOV     #1,-(SP)
013142  010600                                  MOV     SP,R0
013144  104415                                  TRAP    C$PNTX
013146  062706  000004                          ADD     #4,SP
013152  000766                                  BR      10$             ;DO ANOTHER MSG LINE
013154  012700  000012                  20$:    MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
013160  004737  014612                          JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
013164                                          ENDMSG
013164                                  L10014:
013164  104423                                  TRAP    C$MSG

013166  013206  013261  013360        LOOPCOD:        .WORD   1$,2$,3$,4$,5$,6$,7$,0
013206     045     116     045        1$:.ASCIZ  '<N><A Tape Bus Loopback Signals in Word #8:'
013261     045     116     045        2$:.ASCIZ  '<N><A       PARERR<15>          IRESV2<14>          IRESV1<13>'
013360     045     116     045        3$:.ASCIZ  '<N><A IHISP=>IEOT<12>    IWRT=>IIDENT<11>    IREV  =>ICER  <10>'
013457     045     116     045        4$:.ASCIZ  '<N><A IWFM =>IFMK<09>    IEDIT=>IHER <08>    IFAD  =>ISPFED<07>'
013556     045     116     045        5$:.ASCIZ  '<N><A ITADO=>IRDY<06>    ITAD1=>IONL <05>    IERASE=>ILDP  <04>'
013655     045     116     045        6$:.ASCIZ  '<N><A IREW =>IDBY<03>    IRWU =>IRWD <02>    IFEN  =>IFBY  <01>'
013754     045     116     045        7$:.ASCIZ  '<N><A IGO  =>IFPT<00>'
                                                .EVEN
```

C6

MSGSUB PRINT WRITE SUBSYSTEM MESSAGE BUFFER

```
                                        .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
                                ;-
014002                                  BGNMSG  MSGSUB
014002                          MSGSUB::
014002  012700  000012                  MOV     #10.,R0         ;SIZE OF WRITE SUBSYSTEM BUFFER
014006  004737  014612                  JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
014012                                  ENDMSG
014012                          L10015:
014012  104423                          TRAP    C$MSG

                                        .SBTTL  MEMADD   PRINT MEMORY ADDRESS DATA ERROR
                                ;*
                                ;
                                ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       ERRHI   - MEMORY ERROR HIGH ORDER ADDRESS
                                ;       ERRLO   - MEMORY ERROR LOW ORDER ADDRESS
                                ;       EXP     - EXPECTED DATA
                                ;       RECV    - RECEIVED DATA
                                ;-
                                        BGNMSG  MEMADD
014014                          MEMADD::
014014  004737  010162                  JSR     PC,PRIADD       ;PRINT MEMORY ADDRESS IN ERROR
014020  013701  002234                  MOV     EXPD,R1         ;GET EXPD DATA
014024  013702  002236                  MOV     RECV,R2         ;GET RECEIVED DATA
014030  004737  007744                  JSR     PC,PRIXOR       ;PRINT EXPD/RECV
014034                                  ENDMSG
014034                          L10016:
014034  104423                          TRAP    C$MSG
```

D6


```
                                        .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
                                ;
014036                          PRAMPKT:
014036                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
014042  012701  002244                  MOV     @RAMDATA,R1             ;DATA FROM THE RAM
014046  005002                          CLR     R2                      ;INIT BYTE NUMBER
014050  122124                  5$:     CMPB    (R1)+,(R4)+             ;COMPARE EXPECTED, RECEIVED
014052  001005                          BNE     7$                      ;BR IF NO MATCH
014054                                  FORCERROR       7$,NOTSSR
014064  000436                          BR      10$                     ;@@0
014066  116105  177777          7$:     MOVB    -1(R1),R5               ;GET RECV RAM DATA
014072  116403  177777                  MOVB    -1(R4),R3               ;GET EXPD PACKET DATA
014076                                  XOR     R5,R3                   ;XOR EXPD/RECV
014106  042703  177400                  BIC     #177400,R3              ;LOW BYTE ONLY
014112  116137  177777  002236          MOVB    -1(R1),RECV             ;GET RECEIVED RAM DATA
014120  116437  177777  002234          MOVB     1(R4),EXPD             ;GET EXPECTED RAM DATA
014126                                  PRINTB  #RAMASC,R2,RECV,EXPD,R3
014126  010346                          MOV     R3,-(SP)
014130  013746  002234                  MOV     EXPD,-(SP)
014134  013746  002236                  MOV     RECV,-(SP)
014140  010246                          MOV     R2,-(SP)
014142  012746  014216                  MOV     #RAMASC,-(SP)
014146  012746  000005                  MOV     #5,-(SP)
014152  010600                          MOV     SP,R0
014154  104414                          TRAP    C$PNTB
014156  062706  000014                  ADD     #14,SP
014162  005202                  10$:    INC     R2                      ;UPDATE BYTE COUNT
014164  005737  002304                  TST     RAMSIZ                  ;DEFAULT TO 8.?
014170  001404                          BEQ     15$                     ;BR IF YES
014172  020237  002304                  CMP     R2,RAMSIZ               ;DONE ALL BYTES?
014176  003724                          BLE     5$                      ;BR IF NO
014200  000403                          BR      25$                     ;
014202  020227  000010          15$:    CMP     R2,#8.                  ;DONE DEFAULT NUMBER OF BYTES?
014206  002720                  20$:    BLT     5$                      ;BR IF NO
014210  005037  002304          25$:    CLR     RAMSIZ                  ;SET DEFAULT RAMSIZ
014214  000207                          RTS     PC                      ;RETURN

014216    045   116   045   RAMASC:   .ASCIZ  '%N%A  BYTE: %D2%A RAM: %03%A Packet: %03%A XOR:%03'
                                      .EVEN
```

E6

PRMESS PRINT CONTENTS OF MESSAGE BUFFER

```
                                        .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
                                ;
014302                          PRMESS: SAVREG                  ;SAVE THE REGISTERS
014306  010005                          MOV     R0,R5           ;SAVE LOW ORDER ADDRESS
014310  005737  003136                  TST     KTENABLE        ;ADDRESS ABOVE 28K?
014314  001001                          BNE     10$             ;BR IF YES
014316  005001                          CLR     R1              ;SET HIGH ORDER ADDRESS TO 0
014320  010103                  10$:    MOV     R1,R3           ;SAVE HIGH ORDER ADDRESS
014322  006100                          ROL     R0              ;SHIFT BIT15 TO C BIT
014324  006101                          ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
014326                                  PRINTX  #PROASC,R1,R5   ;PRINT MESSAGE BUFFER ADDRESS
014326  010546                          MOV     R5, (SP)
014330  010146                          MOV     R1,-(SP)
014332  012746  014460                  MOV     #PROASC, (SP)
014336  012746  000003                  MOV     #3,-(SP)
014342  010600                          MOV     SP,R0
014344  104415                          TRAP    C$PNTX
014346  062706  000010                  ADD     #10,SP
014352                                  PRINTX  #PR1ASC         ;PRINT HEADER FOR CONTENTS
014352  012746  014525                  MOV     #PR1ASC,-(SP)
014356  012746  000001                  MOV     #1,-(SP)
014362  010600                          MOV     SP,R0
014364  104415                          TRAP    C$PNTX
014366  062706  000004                  ADD     #4,SP
014372  005004                          CLR     R4              ;NUMBER OF THE NEXT WORD
014374  010501                          MOV     R5,R1           ;COPY LOW ORDER ADDRESS
014376  010300                          MOV     R3,R0           ;COPY HIGH ORDER ADDRESS
014400  001403                          BEQ     20$             ;BR IF NOT ABOVE 28K
014402  004737  017356                  JSR     PC,SETMAP       ;SETUP PAR ADDRESS IN R0
014406  010005                          MOV     R0,R5           ;GET PAR FORMAT ADDRESS ABOVE 28K
014410                          20$:    PRINTX  #PRASC,R4,(R5)+ ;PRINT THE CONTENTS OF MEMORY BUFFER
014410  012546                          MOV     (R5)+,-(SP)
014412  010446                          MOV     R4,-(SP)
014414  012746  014563                  MOV     #PRASC,-(SP)
014420  012746  000003                  MOV     #3, (SP)
014424  010600                          MOV     SP,R0
014426  104415                          TRAP    C$PNTX
014430  062706  000010                  ADD     #10,SP
014434  005204                          INC     R4              ;NUMBER OF THE NEXT
014436  020427  000007                  CMP     R4,#7           ;DONE ALL YET ?
014442  003005                          BGT     50$             ;BRANCH IF ALL DONE
014444  002761                          BLT     20$             ;PRINT FIRST 7 WORDS
014446  032763  000200  000012          BIT     #X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
014454  001355                          BNE     20$             ;PRINT EXTENDED STATUS WORD
014456  000207                  50$:    RTS     PC              ;RETURN
014460     045     116     045  PROASC: .ASCIZ  '<N><A Message Buffer Address = <01><05>'
014525     045     116     045  PR1ASC: .ASCIZ  '<N><A Message Buffer Contents:'
014563     045     116     045  PRASC:  .ASCIZ  '<N><A    Word<D1><A: <0>'
```

F6

SEQ 0070

PRMESS PRINT CONTENTS OF MESSAGE BUFFER

```
                                        .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
                                ;
014612                          PRMSGEXP::
014612                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
014616  010005                          MOV     R0,R5                   ;SAVE NUMBER OF WORDS
014620  013700  002310                  MOV     RCVLOADD,R0             ;GET RECV LOW ADDRESS
014624  010004                          MOV     R0,R4                   ;COPY LOW ADDRESS
014626  013701  002306                  MOV     RCVHIADD,R1             ;GET RECV HIGH ADDRESS
014632  006100                          ROL     R0                      ;SHIFT BIT15 TO C BIT
014634  006101                          ROL     R1                      ;SHIFT TO HIGH ORDER FOR PRINTOUT
014636                                  PRINTX  #PRMSG0,R1,R4           ;PRINT MESSAGE BUFFER ADDRESS
014636  010446                          MOV     R4,-(SP)
014640  010146                          MOV     R1,-(SP)
014642  012746  014772                  MOV     #PRMSG0,-(SP)
014646  012746  000003                  MOV     #3,-(SP)
014652  010600                          MOV     SP,R0
014654  104415                          TRAP    C$PNTX
014656  062706  000010                  ADD     #10,SP
014662                                  PRINTX  #PRMSG1                 ;PRINT HEADER FOR CONTENTS
014662  012746  015037                  MOV     #PRMSG1,-(SP)
014666  012746  000001                  MOV     #1,-(SP)
014672  010600                          MOV     SP,R0
014674  104415                          TRAP    C$PNTX
014676  062706  000004                  ADD     #4,SP
014702  005004                          CLR     R4                      ;NUMBER OF THE CURRENT WORD
014704  012701  002324                  MOV     #EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
014710  012702  002470                  MOV     #RECMSG,R2              ;GET RECV BUFFER ADDRESS
014714  011100                  20$:    MOV     (R1),R0                 ;GET EXPD
014716  011203                          MOV     (R2),R3                 ;GET RECV
014720                                  XOR     R0,R3                   ;XOR EXPD/RECV
014730                                  PRINTX  #PRMSG2,R4,(R1)+,(R2)+,R3
014730  010346                          MOV     R3,-(SP)
014732  012246                          MOV     (R2)+,-(SP)
014734  012146                          MOV     (R1)+,-(SP)
014736  010446                          MOV     R4,-(SP)
014740  012746  015075                  MOV     #PRMSG2,-(SP)
014744  012746  000005                  MOV     #5,-(SP)
014750  010600                          MOV     SP,R0
014752  104415                          TRAP    C$PNTX
014754  062706  000014                  ADD     #14,SP
014760  005204                          INC     R4                      ;NUMBER OF THE NEXT
014762  020405                          CMP     R4,R5                   ;DONE ALL YET?
014764  002001                          BGE     50$                     ;BR IF YES
014766  000752                          BR      20$                     ;DO ANOTHER
014770  000207                  50$:    RTS     PC                      ;RETURN
015037      045     116     045 PRMSG0: .ASCIZ   '%N%A Message Buffer Address = %01%05'
015075      045     116     045 PRMSG1: .ASCIZ   '%N%A Message Buffer Contents:'
                                PRMSG2: .ASCIZ   '%N%A    WORD #%02%A  EXPD: %06%A  RECV: %06%A  XOR: %06'
```

PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS

```
                                        .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
                                ;
015162                          PRBYTEXP::
015162                                  SAVREG                          ;SAVE R1 R5 UNTIL NEXT RETURN
015166  010005                          MOV     R0,R5                   ;SAVE NUMBER OF BYTES
015170  005037  002322                  CLR     PRMNO                   ;INIT ERROR COUNT
015174  005004                          CLR     R4                      ;NUMBER OF THE CURRENT BYTE
015176  012701  002324                  MOV     @EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
015202  012702  002470                  MOV     @RECMSG,R2              ;GET RECV BUFFER ADDRESS
015206  111100                  20$:    MOVB    (R1),R0                 ;GET EXPD BYTE
015210  042700  177400                  BIC     #^C<377>,R0             ;CLEAR UPPER BYTE
015214  110037  015530                  MOVB    R0,PRBEXP               ;SAVE FOR ERROR REPORT
015220  111203                          MOVB    (R2),R3                 ;GET RECV BYTE
015222  042703  177400                  BIC     #^C<377>,R3             ;CLEAR UPPER BYTE
015226  110337  015532                  MOVB    R3,PRBREC               ;FOR ERROR REPORT
015232                                  XOR     R0,R3                   ;XOR EXPD/RECV
015242  122122                          CMPB    (R1)+,(R2)+             ;EXPD = RECV?
015244  001431                          BEQ     30$                     ;BR IF YES
015246  005237  002322                  INC     PRMNO                   ;UPDATE ERROR COUNT
015252  023727  002322  000010          CMP     PRMNO,#8.               ;PRINTED 8?
015260  101023                          BHI     30$                     ;BR IF YES
015262                          27$:    PRINTX  #PRBMSG,R4,PRBEXP,PRBREC,R3
015262  010346                          MOV     R3, (SP)
015264  013746  015532                  MOV     PRBREC,-(SP)
015270  013746  015530                  MOV     PRBEXP,-(SP)
015274  010446                          MOV     R4, (SP)
015276  012746  015376                  MOV     #PRBMSG, (SP)
015302  012746  000005                  MOV     #5, (SP)
015306  010600                          MOV     SP,R0
015310  104415                          TRAP    C$PNTX
015312  062706  000014                  ADD     #14,SP
015316                                  FORCEXIT        50$             ;@@D
015326  000404                          BR      35$                     ;@@D
015330                          30$:
015330                                  FORCERROR  27$,NOTSSR           ;@@D
015340                          35$:                                    ;@@D
015340  005204                          INC     R4                      ;NUMBER OF THE NEXT
015342  020405                          CMP     R4,R5                   ;DONE ALL YET?
015344  002001                          BGE     50$                     ;BR IF YES
015346  000717                          BR      20$                     ;DO ANOTHER
015350                          50$:    PRINTX  #PRBTOT,PRMNO           ;PRINT TOTAL ERROR COUNT
015350  013746  002322                  MOV     PRMNO,-(SP)
015354  012746  015463                  MOV     #PRBTOT,-(SP)
015360  012746  000002                  MOV     #2, (SP)
015364  010600                          MOV     SP,R0
```

H6

PRBYTEXP PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

```
        015366  104415                          TRAP    C$PNTX
        015370  062706  000006                  ADD     #6,SP
  2799  015374  000207                          RTS     PC              ;RETURN
  2800
  2801  015376      045     116     045  PRBMSG: .ASCIZ  '#N#A   BYTE #*D2#A  EXPD: #03#A  RECV: #03#A  XOR: #03'
  2802  015463      045     116     045  PRBTOT: .ASCIZ  '#N#A NUMBER OF BYTES IN ERROR = #D2'
  2803                                          .EVEN
  2804  015530  000000                  PRBEXP: .WORD   0               ;EXPD
  2805  015532  000000                  PRBREC: .WORD   0               ;RECV
  2806                                          .SBTTL  EXPREC  - PRINT EXPD/RECV WORD DATA
  2807                                  ;+
  2808                                  ;
  2809                                  ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
  2810                                  ;
  2811                                  ;INPUTS:
  2812                                  ;
  2813                                  ;       R1      RECEIVED DATA
  2814                                  ;       R2      EXPECTED DATA
  2815                                  ;
  2816                                  ;-
  2817
  2818  015534                          EXPREC::        BGNMSG  EXPREC
        015534
  2819  015534  004737  007744                          JSR     PC,PRIXOR               ;PRINT THE DATA
  2820  015540                                          ENDMSG
        015540                          L10017:
        015540  104423                          TRAP    C$MSG
  2821                                          .SBTTL  EXPBREC - PRINT EXPD/RECV BYTE DATA
  2822                                  ;+
  2823                                  ;
  2824                                  ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
  2825                                  ;
  2826                                  ;
  2827                                  ;INPUTS:
  2828                                  ;
  2829                                  ;       R1      RECEIVED DATA BYTE
  2830                                  ;       R2      EXPECTED DATA BYTE
  2831                                  ;
  2832                                  ;
  2833
  2834  015542                          EXPBREC::       BGNMSG  EXPBREC
        015542
  2835  015542  004737  007614                          JSR     PC,PRIBXOR              ;PRINT THE DATA
  2836  015546                                          ENDMSG
        015546                          L10020:
        015546  104423                          TRAP    C$MSG
  2837
  2838
  2839                                          .SBTTL  RAMERR    PRINT RAM AND PACKET DATA
  2840                                  ;+
  2841                                  ;
  2842                                  ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
  2843                                  ;
  2844                                  ;INPUTS:
  2845                                  ;
  2846                                  ;       R4      POINTER TO COMMAND PACKET
  2847                                  ;
```

RAMERR PRINT RAM AND PACKET DATA

```
                                        ;IMPLICIT INPUTS:
                                        ;
                                        ;       RAMDATA         DATA AS READ FROM THE RAM
                                        ;       RAMSIZ          NUMBER OF BYTE  IN PACKET
                                        ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
                                        ;
                                        ;IMPLICIT OUTPUTS:
                                        ;
                                        ;       RAMSIZ  SET TO 0
                                        ;

                                                BGNMSG  RAMERR
                                        RAMERR::
015550  004737  014036                          JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
                                                ENDMSG
                                        L10021:
015554  104423                                  TRAP    C$MSG

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

                                                BGNMSG  RAMTADD
                                        RAMTADD::
015556  004737  010276                          JSR     PC,PRITADD      ;PRINT TEST ADDRESS
015562  004737  014036                          JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
                                                ENDMSG
                                        L10022:
015566  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
```

J6

RAMEXP PRINT RAM EXPD/RECV DATA

```
                                        ;       R4      CONTROLLER RAM ADDRESS
                                        ;
                                        ;
                                                BGNMSG  RAMEXP
                                        RAMEXP::
015570  042701  177400                          BIC     #^C<377>,R1             ;SAVE EXPD RAM DATA BYTE
015574  042702  177400                          BIC     #^C<377>,R2             ;SAVE EXPD RAM DATA BYTE
015600  004737  010070                          JSR     PC,PRIRAM               ;PRINT THE RAM ADDRESS
015604  004737  007744                          JSR     PC,PRIXOR               ;PRINT THE DATA
015610                                          ENDMSG
                                        L10023:
015610  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::
015612                                          PRINTX  #TIMSG0                 ;PRINT HEADER
015612  012746  015640                          MOV     #TIMSG0, (SP)
015616  012746  000001                          MOV     #1, (SP)
015622  010600                                  MOV     SP,R0
015624  104415                                  TRAP    C$PNTX
015626  062706  000004                          ADD     #4,SP
015632  004737  007744                          JSR     PC,PRIXOR               ;PRINT THE DATA
015636                                          ENDMSG
                                        L10024:
015636  104423                                  TRAP    C$MSG

015640  045  116  045  TIMSG0: .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::
015740  010246                                  MOV     R2, (SP)                ;SAVE DATA TRANSFERRED
015742  042702  177400                          BIC     #177400,R2              ;GET JUST ONE BYTE
```

K6

SEQ 0075

BADSSR PRINT TSSR ERRORS ON DATA TRANSFERS

```
 2944 015746                                    PRINTB  #XFERASC,R2
      015746  010246                            MOV     R2, (SP)
      015750  012746  016000                    MOV     #XFERASC,-(SP)
      015754  012746  000002                    MOV     #2, (SP)
      015760  010600                            MOV     SP,R0
      015762  104414                            TRAP    C$PNTB
      015764  062706  000006                    ADD     #6,SP
 2945 015770  012602                            MOV     (SP)+,R2                ;RESTORE R2
 2946 015772  004737  006026                    JSR     PC,PRITSSR              ;DECODE TSSR CONTENTS
 2947 015776                                    ENDMSG
      015776                            L10025:
      015776  104423                            TRAP    C$MSG
 2948 016000     045     116     045  XFERASC:          .ASCIZ  '#N#A Data Transferred = #03'
```

L6

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
                                        ;
                                        ;-

016034                                  SOFINIT::
016034                                          SAVREG                  ; SAVE THE REGISTERS
016040  012765  000000  000002                  MOV     #0,TSSR(R5)     ; DO THE INIT.
016046  004737  016310                          JSR     PC,WAITF        ; WAIT FOR SSR
016052  016500  000002                          MOV     TSSR(R5),R0     ;GET THE TSSR REGISTER
016056  010004                                  MOV     R0,R4           ;TSSR CONTENTS
016060  042704  176277                          BIC     #^C<HIADDR!OFL>,R4
016064  052704  002200                          BIS     #SSR!NBA,R4     ;R4 HAS EXPECTED CONTENTS
016070  020400                                  CMP     R4,R0           ;ONLY EXPECTED BITS SET ?
016072  001402                                  BEQ     5$              ;BRANCH IF OKAY
016074  000241                                  CLC                     ;CLEAR THE CARRY FOR ERROR
016076  000401                                  BR      10$             ;GO TO EXIT
016100  000261                          5$:     SEC                     ;SET THE CARRY BIT
016102  000207                          10$:    RTS     PC              ;RETURN TO CALLER
```

M6

CHKAMB CHECK TSSR FOR AMBIGUITY

```
                                        .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
                                        ;
                                        ;
016104                                  CHKAMB:
016104                                          SAVREG                  ;SAVE THE GENERAL REGISTERS
016110  010004                                  MOV     R0,R4           ;CONTENTS OF TSSR
016112  032700  100000                          BIT     #SC,R0          ;IS BIT 15 SET ?
016116  001004                                  BNE     5$              ;BRANCH IF YES
016120  032700  174077                          BIT     #^C<NBA!OFL!SSR!HIADDR>,R0      ;ANY OTHER BITS SET ?
016124  001023                                  BNE     40$             ;MUST BE AN ERROR
016126  000424                                  BR      45$             ;RETURN WITH SUCCESS
016130  032700  000200          5$:             BIT     #SSR,R0         ;IS READY BIT SET ?
016134  001011                                  BNE     10$             ;BRANCH IF READY BIT IS SET.
016136  032700  000040                          BIT     #BIT5,R0        ;IS FATAL ERROR BIT SET ?
016142  001414                                  BEQ     40$             ;ERROR IF NOT
016144  042704  177761                          BIC     #^CTERCLS,R4    ;CLEAR ALL BUT TERMINATION CODE
016150  020427  000016                          CMP     R4,#16          ;ALL THREE BITS MUST BE SET
016154  001007                                  BNE     40$             ;ERROR IF NOT SET
016156  000410                                  BR      45$             ;OK IF ALL ARE SET
016160  032700  000040          10$:            BIT     #BIT5,R0        ;IS FATAL ERROR BIT SET ?
016164  001405                                  BEQ     45$             ;ERROR IF BIT IS SET WITH SSR
016166  032700  000006                          BIT     #BIT2!BIT1,R0   ;IS THIS A FUNCTION REJECT
016172  001002                                  BNE     45$             ;BR, IF TSSR IS OK
016174  000241                  40$:            CLC                     ;AMBIGUOUS CONTENTS
016176  000401                                  BR      50$
016200  000261                  45$:            SEC                     ;SHOW SUCCESS - NO AMBIGUITY
016202  000207                  50$:            RTS     PC              ;RETURN TO CALLER
```

N6

ENAINT,DSBINT ENABLE/DISABLE INTERRUPTS

```
                                .SBTTL  ENAINT,DSBINT   ENABLE/DISABLE INTERRUPTS
                                ;
                                ; DEFAULT DISPLAY INTERRUPT HANDLERS.
                                ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
                                ; OTHERWISE, SAVE CPU 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
016204     000                  INTMASK:        .BYTE   0
                                ;INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
016205     000                  INTFLAG:        .BYTE   0

                                ;SAVED INTERRUPT VECTOR:
016206  000000                  INTVEC: .WORD   0
                                ;SAVE CPU PC
016210  000000                  INTCPC: .WORD   0

                                ;SUBROUTINE TO ENABLE INTERRUPTS:
016212  010046                  ENAINT: MOV     R0,-(SP)        ;SAVE R0
016214  013700  002210                  MOV     IVEC,R0         ;GET POINTER TO VECTORS
016220  012720  016256                  MOV     #INTR,(R0)+     ;SET UP INTERRUPT VECTOR
016224  012720  000340                  MOV     #PRI07,(R0)+
016230  012600                          MOV     (SP)+,R0        ;RESTORE R0
016232  011646                          MOV     (SP),-(SP)
016234  012766  000000  000002          MOV     #0,2(SP)        ;SET CPU TO LEVEL 0
016242  000002                          RTI

                                ;SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
016244  011646                  DSBINT: MOV     (SP),-(SP)
016246  012766  000340  000002          MOV     #PRI07,2(SP)
016254  000002                          RTI
                                        .SBTTL  INTR    - INTERRUPT HANDLERS

                                        BGNSRV  INTR            ;DEFINE INTERRUPT ENTRY
016256                          INTR::
016256  012737  000001  002224          MOV     #1,INTRECV      ;SET FLAG TO SHOW INTERRUPT RECEIVED
016264  105037  016205                  CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
016270  132737  000001  016204          BITB    #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
016276  001003                          BNE     1$              ;BR IF YES
016300  152737  000001  016205          BISB    #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.

                                ;SAVE REGISTERS, MSG BUFFER, ETC.
016306                          1$:
016306                                  ENDSRV
016306                          L10026:
016306  000002                          RTI
```

```
                                        .SBTTL  WAITF     WAIT FOR SUBSYSTEM READY
                                ;
                                ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
                                ;
                                ;INPUTS:
                                ;
                                ;       R5      ADDRESS OF FIRST DEVICE REGISTER
                                ;
                                ;OUTPUTS:
                                ;
                                ;       R0      CONTENTS OF LAST TSSR READ
                                ;       CARRY   SET - READY BIT SET
                                ;               CLR - TIMEOUT WAITING FOR READY
                                ;
016310  000401                  WAITF:: BR      1$              ;NOP WHEN SUPER FIXED
016312                                  BREAK                   ; DO A SUPVSR BREAK FIRST.
016312  104422                          TRAP    C$BRK
016314  012746  011000          1$:     MOV     #11000, (SP)    ;25-APRIL-83 REV B  - 1100 MSEC TIMER
016320  016500  000002          2$:     MOV     TSSR(R5),R0     ;READ THE TSSR REGISTER
016324  105700                          TSTB    R0              ;TEST FOR READY BIT SET
                                
016326  100420                          BMI     3$              ; EXIT ON STOP FLAG.
016330                                  DELAY   1               ; WAIT 100 USEC
016330  012727  000001                  MOV     #1,(PC)+
016334  000000                          .WORD   0
016336  013727  002116                  MOV     L$DLY,(PC)+
016342  000000                          .WORD   0
016344  005367  177772                  DEC      6(PC)
016350  001375                          BNE     .-4
016352  005367  177756                  DEC     -22(PC)
016356  001367                          BNE     . 20
016360  005316                          DEC     (SP)            ;REDUCE DELAY COUNT
016362  001356                          BNE     2$              ;RETRY UNTIL TIMER EXPIRES
016364  000241                          CLC                     ; C = 0, CONTROLLER STILL RUNNING...
016366  000401                          BR      4$              ;...OR HUNG UP AFTER 300 MSEC.
016370  000261                  3$:     SEC                     ; C = 1, CONTROLLER IS STOPPED.
016372  005326                  4$:     DEC     (SP)+           ;RESTORE STACK WITHOUT CHANGING CARRY BIT
016374  000207                          RTS     PC
```

CHKTSSR CHECK TSSR FOR READY

```
                                        .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
                                        ;-
016376                                  CHKTSSR:
016376  004737  016310                          JSR     PC,WAITF                ;WAIT FOR READY
016402  103014                                  BCC     20$                     ;BRANCH IF TIME OUT
016404  004737  016104                          JSR     PC,CHKAMB               ;TSSR AMBIGOUS?
016410  103006                                  BCC     10$                     ;BR IF YES
016412  032700  100000                          BIT     #SC,R0                  ;SPECIAL CONDITION SET?
016416  001405                                  BEQ     15$                     ;BR IF NO
016420  032700  074000                          BIT     #<SCE!BIE!RMR!NXM>,R0   ;ANY ERROR BITS SET?
016424  001402                                  BEQ     15$                     ;BR IF NO
016426  000241                          10$:    CLC                             ;SET FAILURE
016430  000401                                  BR      20$                     ;
016432  000261                          15$:    SEC                             ;SET SUCCESS
016434  000207                          20$:    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.
016436  012737  016470  000004  XNXM:   MOV     #2$,@#4         ; SET BUSERR VECTOR.
016444  012737  000200  000006          MOV     #PRI04,@#6
016452  005003                          CLR     R3                      ;FLAG.
016454  005711                  1$:     TST     (R1)                    ;TEST THE ADDRESS(ES).
                                                                        ;IF ANY TRAP, CONTINUE AT 2$.
016456  020102                          CMP     R1,R2                   ;OTHERWISE, CONTINUE HERE.
016460  001407                          BEQ     3$                      ;BR IF FINISHED (NO NEXM'S).
016462  062701  000002                  ADD     #2,R1                   ;SET NEXT ADDRESS...
016466  000772                          BR      1$                      ;...AND CONTINUE.
016470  005103                  2$:     COM     R3                      ;GOT ONE, SET FLAG...
016472  012716  016500                  MOV     #3$,(SP)
016476  000002                          RTI                             ;...AND DISMISS INTERRUPT...
016500                          3$:     CLRVEC  #4                      ;...AND GIVE BACK THE VECTOR.
016500  012700  000004                  MOV     #4,R0
016504  104436                          TRAP    C$CVEC
016506  005703                          TST     R3                      ;DID WE CATCH ONE ??
016510  001401                          BEQ     .+4                     ;NO, "C" = 0, SKIP NEXT.
016512  000261                          SEC                             ;YES, "C" = 1, (R1) = NEXM ADDR.
016514  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
                                ;
016516                          TSTLOOP::
016516  005737  002170                  TST     NOITS           ; ITERATIONS INHIBITED?
016522  001006                          BNE     1$              ; YES.
016524  005737  002204                  TST     QVP             ; NO.
016530  100403                          BMI     1$              ;LOOPS DISALLOWED IN QUICK PASS.
016532  005337  002216                  DEC     LOOPCNT         ; BUMP LOOP COUNTER.
016536  001002                          BNE     2$
016540  000241                  1$:     CLC                     ;LOOP DISALLOWED, OR DONE.
016542  000401                          BR      3$
016544  000261                  2$:     SEC                     ;LOOP ENABLED.
016546  000207                  3$:     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
                                ;
                                ;-

016550                          TSTSETUP::
016550  010046                          MOV     R0,-(SP)        ;SAVE THE TEST ID MESSAGE
016552  005037  003156                  CLR     SIFLAG          ; CLEAR "SOFT INIT" FLAG
016556  005037  017016                  CLR     ERRK            ; CLEAR LOCAL ERROR COUNTER.
016562  005037  005774                  CLR     EXTA            ; CLEAR ERROR EXTENSION FLAG.
016566  105037  016204                  CLRB    INTMASK         ; CLEAR INTERRUPT MASK (CHECK ERROR)
016572  013700  002202                  MOV     UNITN,R0        ; GET THE UNIT NUMBER,
016576  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
016600  005737  003116                  TST     NODEV           ; DID STARTUP FIND THE DEVICE?
016604  001430                          BEQ     4$              ; BR IF YES
016606  100010                          BPL     3$              ; BR IF NOT IDLE
```

TSTSETUP PRINT TEST NAME AND INIT ERROR COUNTS

```
3234 016610  052760  160000  003200          BIS     #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
3235 016616                                  ERRDF   1,NXR,NXRERR     ; NO DEVICE HERE    PRINT IT
     016616  104455                          TRAP    C$ERDF
     016620  000001                          .WORD   1
     016622  003742                          .WORD   NXR
     016624  005740                          .WORD   NXRERR
3236 016626  000407                          BR      2$
3237 016630  052760  160001  003200  3$:     BIS     #160001,ERTABL(R0)      ; FLAG ERROR IN THE ERROR TABLE
3238 016636                                  ERRDF   2,NOINIT        ; DEVICE NOT IDLE
     016636  104455                          TRAP    C$ERDF
     016640  000002                          .WORD   2
     016642  004337                          .WORD   NOINIT
     016644  000000                          .WORD   0
3239 016646  012737  177777  003114  2$:     MOV     #-1,DUFLG       ; DROP THE UNIT
3240 016654                                  DODU    UNITN
     016654  013700  002202                  MOV     UNITN,R0
     016660  104451                          TRAP    C$DODU
3241 016662                                  DOCLN                   ; ABORT THE PASS
     016662  104444                          TRAP    C$DCLN
3242 016664  000423                          BR      5$
3243
3244 016666                          4$:     RFLAGS  R0              ; GET THE OPERATOR FLAGS.
     016666  104421                          TRAP    C$RFLA
3245 016670  032700  001000                  BIT     #PNT,R0         ; PRINT THE TEST NUMBERS?
3246 016674  001412                          BEQ     1$              ; BR IF NO
3247 016676  011600                          MOV     (SP),R0         ;GET THE ID MESSAGE
3248 016700                                  PRINTF  #TNAM,R0                ;DISPLAY THE TEST ID
     016700  010046                          MOV     R0,-(SP)
     016702  012746  016744                  MOV     #TNAM,-(SP)
     016706  012746  000002                  MOV     #2,-(SP)
     016712  010600                          MOV     SP,R0
     016714  104417                          TRAP    C$PNTF
     016716  062706  000006                  ADD     #6,SP
3249 016722  005237  002214          1$:     INC     TSTCNT          ; BUMP TEST COUNTER.
3250 016726                                  SETPRI  IPRI            ;PRIORITY THAT OF DEVICE
     016726  013700  002212                  MOV     IPRI,R0
     016732  104441                          TRAP    C$SPRI
3251 016734  005726                  5$:     TST     (SP)+           ;FIX UP THE STACK
3252 016736  013705  002206                  MOV     CSRADDR,R5      ; ADDRESS OF TSV REGISTERS ON UNIBUS
3253 016742  000207                          RTS     PC
3254 016744     045     123     045  TNAM:   .ASCIZ  '%S%T%A Test'
3255                                         .EVEN
3256                                         .SBTTL  TSTEND  - PRINT ERRORS RECEIVED
3257                                 ;
3258                                 ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
3259                                 ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
3260                                 ;
3261 016760                          TSTEND: RFLAGS  R0
     016760  104421                          TRAP    C$RFLA
3262 016762  030027  020000                  BIT     R0,#IER
3263 016766  001412                          BEQ     1$              ; BR IF "IER" NOT SET.
3264 016770                                  PRINTF  #ESUM,ERRK      ; PRINT ERROR COUNT.
     016770  013746  017016                  MOV     ERRK,-(SP)
     016774  012746  017020                  MOV     #ESUM,-(SP)
     017000  012746  000002                  MOV     #2,-(SP)
     017004  010600                          MOV     SP,R0
     017006  104417                          TRAP    C$PNTF
```

F7

TSTEND - PRINT ERRORS RECEIVED

```
        017010  062706  000006                  ADD     #6.SP
   3265 017014  000207                  1$:     RTS     PC
   3266
   3267 017016  000000                  ERRK:   0                       ; LOCAL ERROR COUNT.
   3268 017020     045     101     040  ESUM:   .ASCIZ  /%A %D%A ERRORS/
   3269 017037     105     122     122  EMAXDU: .ASCIZ  /ERROR LIMIT REACHED    DROPPING UNIT/
   3270                                         .EVEN
```

INCERK - INCREMENT LOCAL ERROR COUNT

```
                                        .SBTTL  INCERK    INCREMENT LOCAL ERROR COUNT
                                ;+
                                ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
                                ;-
017104  005237  017016          INCERK: INC     ERRK            ; INCREMENT LOCAL ERROR COUNT
017110  010046                          MOV     R0, (SP)        ; SAVE R0
017112  013700  002202                  MOV     UNITN,R0        ; GET UNIT NUMBER,
017116  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
017120  062700  003200                  ADD     #ERTABL,R0      ; R0 GETS ADDRESS OF ERROR TABLE ENTRY.
017124  005210                          INC     (R0)            ; INCREMENT THE DEVICE ERROR COUNT
017126  032710  007777                  BIT     #7777,(R0)      ; DID WE OVERFLOW THE FIELD?
017132  001001                          BNE     1$              ; BR IF NO.
017134  005310                          DEC     (R0)            ; YES -- BACK IT UP TO 7777.
017136  012600                  1$:     MOV     (SP)+,R0        ; RESTORE R0
017140  000207                          RTS     PC              ; RETURN TO CALLER

017142  010046                  CKEMAX: MOV     R0,-(SP)        ; SAVE R0
017144  013700  002202                  MOV     UNITN,R0        ; GET UNIT NUMBER
017150  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET
017152  016000  003200                  MOV     ERTABL(R0),R0   ; GET ERROR TABLE ENTRY
017156  042700  170000                  BIC     #170000,R0      ; EXTRACT ERROR COUNT FIELD
017162  020037  002174                  CMP     R0,GERRMAX      ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
017166  103004                          BHIS    1$              ; BR IF YES
017170  023737  017016  002172          CMP     ERRK,LERRMAX    ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
017176  103417                          BLO     2$              ; BR IF NO
017200                          1$:     RFLAGS  R0              ; GET OPERATOR FLAGS
017200  104421                          TRAP    C$RFLA
017202  032700  000040                  BIT     #IDU,R0         ; IS DROPPING INHIBITED?
017206  001013                          BNE     2$              ; BR IF YES.
017210  012737  177777  003114          MOV     #-1,DUFLG       ; NO -  DROP THE UNIT
017216                                  ERRDF   4,EMAXDU
017216  104455                          TRAP    C$ERDF
017220  000004                          .WORD   4
017222  017037                          .WORD   EMAXDU
017224  000000                          .WORD   0
017226                                  DODU    UNITN
017226  013700  002202                  MOV     UNITN,R0
017232  104451                          TRAP    C$DODU
017234                                  DOCLN
017234  104444                          TRAP    C$DCLN
017236  012600                  2$:     MOV     (SP)+,R0        ; RESTORE R0
017240  000207                          RTS     PC              ; RETURN TO CALLER
```

CKDROP CHECK IF UNIT SHOULD BE DROPPED

```
                                        .SBTTL  CKDROP    CHECK IF UNIT SHOULD BE DROPPED
                                ;+
                                ; CHECK IF UNIT SHOULD BE DROPPED
                                ;-
017242  010046                  CKDROP: MOV     R0, (SP)
017244                                  FORCERROR       1$,NOTSSR
017254                                  RFLAGS  R0
017254  104421                          TRAP    C$RFLA
017256  032700  000040                  BIT     #IDU,R0
017262  001010                          BNE     1$
017264  011600                          MOV     (SP),R0
017266  012737  177777  003114          MOV     #-1,DUFLG
017274                                  DODU    UNITN
017274  013700  002202                  MOV     UNITN,R0
017300  104451                          TRAP    C$DODU
017302                                  DOCLN                   ;ABORT THE PASS
017302  104444                          TRAP    C$DCLN
017304  012600                  1$:     MOV     (SP)+,R0
017306  000207                          RTS     PC

                                        .SBTTL  CONFIG  - DETERMINE CONFIGURATION OF SYSTEM
                                ;
                                ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
                                ;
017310                          CONFIG:
017310  004737  016034                  JSR     PC,SOFINIT
017314  000207                          RTS     PC
                                        .SBTTL  KTON,KTOFF        ENABLE/DISABLE MEMORY MANAGEMENT
                                ;
                                ; SUBROUTINE - ENABLE MEM MGT.
                                ;
017316  005737  003134          KTON:   TST     KTFLG           ; GOT KT?
017322  001403                          BEQ     1$              ; NO.
017324  012737  000001  177572          MOV     #1,SR0          ; YES. ENABLE KT11.
017332  000207                  1$:     RTS     PC

                                ;
                                ; SUBROUTINE - DISABLE MEM MGT.
                                ;
017334  005737  003134          KTOFF:  TST     KTFLG           ; GOT KT11?
017340  001405                          BEQ     1$              ; NO.
017342  000240                          NOP
017344  000240                          NOP
017346  012737  000000  177572          MOV     #0,SR0          ; DISABLE KT.
017354  000207                  1$:     RTS     PC
```

SETMAP SETUP PAR6 MAPPING

```
                                        .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
                                ;
017356                          SETMAP:
017356                                  SAVREG                          ;SAVE R1 R4 UNTIL NEXT RETURN
017362  005737  003134                  TST     KTFLG                   ;SYSTEM HAVE ABOVE 28K?
017366  001433                          BEQ     10$                     ;BR IF NO
017370  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
017422  042701  000177                  BIC     #177,R1                 ;ALIGN FOR LOWER 4K BOUNDARY
017426  020137  003134                  CMP     R1,KTFLG                ;HIGHER THAN EXISTING MEMORY?
017432  103011                          BHIS    10$                     ;BR IF YES
017434  010137  172352                  MOV     R1,@#KIPAR5             ;SETUP MAPPING REGISTER PAR5
017440  042702  160000                  BIC     #160000,R2              ;SETUP DISPLACEMENT IN PAGE
017444  062702  120000                  ADD     #120000,R2              ;ADD IN PAR5 BIAS
017450  010200                          MOV     R2,R0                   ;RETURN IN R0
017452  000261                          SEC                             ;SET SUCCESS
017454  000401                          BR      15$                     ;
017456  000241                  10$:    CLC                             ;SET FAILURE
017460  000207                  15$:    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
                                ;
```

J7

REGSAV SAVE R1 R5 ON STACK

```
3407                                    ;
3408
3409 017462                             REGSAV:
3410 017462  010446                             MOV     R4,-(SP)
3411 017464  010346                             MOV     R3,-(SP)
3412 017466  010246                             MOV     R2,-(SP)
3413 017470  010146                             MOV     R1,-(SP)
3414 017472  010546                             MOV     R5,-(SP)
3415 017474  016605  000012                     MOV     10.(SP),R5
3416 017500  004736                             JSR     PC,@(SP)+
3417 017502  012601                             MOV     (SP)+,R1
3418 017504  012602                             MOV     (SP)+,R2
3419 017506  012603                             MOV     (SP)+,R3
3420 017510  012604                             MOV     (SP)+,R4
3421 017512  012605                             MOV     (SP)+,R5
3422 017514  000207                             RTS     PC
```

```
                                        .SBTTL  GETPAT    GET 8 BIT PATTERN FROM OPERATOR
                                ;+
                                ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
                                ;
                                ;INPUTS:        NONE.
                                ;
                                ;OUTPUTS:
                                ;       R0      OCTAL NUMBER FROM THE OPERATOR
                                ;
                                ;CALLING SEQUENCE:
                                ;       JSR     PC,GETPAT
                                ;
                                GETPAT::
                                        SAVREG                          ;SAVE THE GENERAL REGISTERS
                                1$:     GMANID  DATASC,PATDAT,0,377,0,377,NO
017522  104443                          TRAP    C$GMAN
017524  000406                          BR      10000$
017526  017552                          .WORD   PATDAT
017530  000022                          .WORD   T$CODE
017532  017554                          .WORD   DATASC
017534  000377                          .WORD   377
017536  000000                          .WORD   T$LOLIM
017540  000377                          .WORD   T$HILIM
017542                          10000$:
                                        BNCOMPLETE      1$      ;RETRY IF ERROR
017542  103367                          BCC     1$
017544  013700  017552                  MOV     PATDAT,R0       ;DATA PATTERN FROM OPERATOR
017550  000207                          RTS     PC              ;RETURN TO CALLER

                                ;+
                                ;LOCAL DATA AREA
                                ;

017552  000000                  PATDAT: .WORD   0               ;TEMPORARY STORAGE FOR DATA
017554    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
                                        ;
017600                                  GETSEL::
017600                                          SAVREG                          ;SAVE GENERAL REGISTERS
017604  010002                                  MOV     R0,R2                   ;SAVE THE MENU ADDRESS
017606  010203                          1$:     MOV     R2,R3                   ;START OF MENU STRING
017610  005713                          2$:     TST     (R3)                    ;END OF ASCII ?
017612  001412                                  BEQ     3$                      ;BRANCH IF ALL LINES DISPLAYED
017614                                          PRINTF  #SELASC,(R3)+           ;DISPLAY THE MENU
01761+  012346                                  MOV     (R3)+, (SP)
017616  012746  017764                         MOV     #SELASC,-(SP)
017622  012746  000002                         MOV     #2,-(SP)
017626  010600                                  MOV     SP,R0
017630  104417                                  TRAP    C$PNTF
017632  062706  000006                         ADD     #6,SP
017636  000764                                  BR      2$
017640                                  3$:     GMANID  MENASC,MENRES,D, 1,0, 1,NO
017640  104443                                  TRAP    C$GMAN
017642  000406                                  BR      10001$
017644  020020                                  .WORD   MENRES
017646  000042                                  .WORD   T$CODE
017650  017771                                  .WORD   MENASC
017652  177777                                  .WORD   -1
017654  000000                                  .WORD   T$LOLIM
017656  177777                                  .WORD   T$HILIM
017660                                  10001$:
017660                                          BNCOMPLETE      1$      ;RETRY IF ERROR
017660  103352                                  BCC     1$
017662  013700  020020                         MOV     MENRES,R0               ;GET THE OPERATOR'S REPLY
017666  020001                                  CMP     R0,R1                   ;COMPARE TO MAXIMUM ALLOWED
017670  101411                                  BLOS    5$                      ;BRANCH IF OK
017672                                          PRINTF  #MENERR                 ;DISPLAY ERROR MESSAGE
017672  012746  017716                         MOV     #MENERR,-(SP)
017676  012746  000001                         MOV     #1,-(SP)
017702  010600                                  MOV     SP,R0
017704  104417                                  TRAP    C$PNTF
017706  062706  000004                         ADD     #4,SP
017712  000735                                  BR      1$                      ;RETRY
017714  000207                          5$:     RTS     PC                      ;RETURN TO CALLER
017716     045     116     045  MENERR: .ASCIZ  '<N><A *** Menu Selection Too Large ***'
017764     045     116     045  SELASC: .ASCIZ  '<N><T>'
017771     105     156     164  MENASC: .ASCIZ  'Enter Menu Selection: '
                                        .EVEN
020020  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.
                                ;
                                ;

020022                          CHKMAN::
                                        SAVREG                  ;SAVE THE REGISTERS
020026                                  MANUAL                  ;SEE IF MANUAL INTERVENTION OK
020026  104450                          TRAP    C$MANI
020030                                  BCOMPLETE 1$            ;BRANCH IF ALLOWED
020030  103411                          BCS     1$
020032                                  PRINTF  #NOMAN          ;PRINT THE WARNING MESSAGE
020032  012746  020056                  MOV     #NOMAN,-(SP)
020036  012746  000001                  MOV     #1,-(SP)
020042  010600                          MOV     SP,R0
020044  104417                          TRAP    C$PNTF
020046  062706  000004                  ADD     #4,SP
020052  000241                          CLC                     ;CLEAR CARRY FOR ERROR
020054  000207                  1$:     RTS     PC              ;RETURN

020056      045     116     045 NOMAN:  .ASCIZ  '%N%A *** Manual Intervention not Allowed - Test Aborted ***'
                                        .even
```

```
3517                                            .SBTTL  ENVIRN    SETUP FREE DIAGNOSTIC SPACE
3518                                    ;
3519                                    ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
3520                                    ;
3521 020152                            ENVIRN: MEMORY  R0
     020152  104431                            TRAP    C$MEM
3522 020154  010037  003126                    MOV     R0,FREE         ; GET 1ST FREE ADDRESS...
3523 020160  062737  000002  003126            ADD     #2,FREE
3524 020166  011037  003130                    MOV     (R0),FRESIZ     ;...AND WORD COUNT.
3525 020172  162737  000004  003130            SUB     #4,FRESIZ
3526 020200  013702  002012                    MOV     L$UNIT,R2       ; GET NUMBER OF UNITS
3527 020204  162737  000007  003130  10$:      SUB     #7,FRESIZ       ; TAKE AWAY 7 WORDS PER UNIT
3528 020212  005302                           DEC     R2
3529 020214  001373                           BNE     10$
3530 020216  013700  003126                    MOV     FREE,R0         ;GET FIRST FREE ADDRESS
3531 020222  063700  003130                    ADD     FRESIZ,R0       ;POINT TO LAST FREE ADDRESS
3532 020226  162700  000002                    SUB     #2,R0           ;BACKUP 1 WORD
3533 020232  010037  003132                    MOV     R0,FREEHI       ;STORE LAST FREE ADDRESS
3534 020236  000240                           NOP                     ;*****************************************
3535 020240  012701  177520                    MOV     #BDVPCR,R1      ;GET BDV11 PCR ADDRESS
3536 020244  010102                           MOV     R1,R2           ;COPY TO R2
3537 020246  062702  000002                    ADD     #2,R2           ;SET THE RANGE
3538 020252  004737  016436                    JSR     PC,XNXM         ;SEE IF WE HAVE ONE
3539 020256  103001                           BCC     15$             ;OK TO SET FLAGS
3540 020260  000423                           BR      40$             ;RETURN WITH FLAGS CLEAR
3541 020262  013701  177520            15$:    MOV     BDVPCR,R1       ;SAVE PCR CONTENTS
3542 020266  062701  000001                    ADD     #1,R1           ;ADD ONE TO IT
3543 020272  012702  177520                    MOV     #BDVPCR,R2      ;GET BDV11 PCR ADDRESS
3544 020276  005212                           INC     (R2)            ;TRY TO WRITE TO IT
3545 020300  013703  177520                    MOV     BDVPCR,R3       ;GET RESULTS
3546 020304  020103                           CMP     R1,R3           ;DID IT CHANGE?
3547 020306  001006                           BNE     20$             ;NO, MUST BE 11/23B
3548 020310  005237  003146                    INC     T23A            ;SET THE FLAG
3549 020314  042737  170000  002120            BIC     #170000,L$HIME  ;SUPERVISOR COULD BE WRONG
3550                                    ;       NOP                     ;BR 40$ FOR RELEASE
3551                                    ;       PRINTF  #M8186          ;TELL THE SYSTEM TYPE
3552 020322  000402                           BR      40$             ;RETURN
3553 020324  005237  003150            20$:    INC     T23B            ;SET THE FLAG
3554                                    ;       NOP                     ;BR 40$ FOR RELEASE
3555                                    ;       PRINTF  #M8189          ;TELL THE SYSTEM TYPE
3556 020330                            40$:
3557 020330  000207                           RTS     PC              ;RETURN
```

```
                                        .SBTTL  KTINIT    SETUP KT11 MEMORY MANAGEMENT REGISTERS
                                ;+
                                ;
                                ;ROUTINE TO INIT KT-11
                                ;
                                ;-
020332                          KTINIT:
020332  005037  003134                  CLR     KTFLG           ; INIT >28K MEMORY FLAG
020336  005037  003136                  CLR     KTENABLE        ; INIT TEST >28K FLAG
020342  023727  002120  001577          CMP     L$HIME,#1577    ; GOT ENOUGH MEMORY (>28K)?
020350  101454                          BLOS    9$              ; NO.
020352  013700  000004                  MOV     @#ERRVEC,R0     ; SAVE OLD ERR VEC PTR.
020356  012737  020470  000004          MOV     #2$,@#ERRVEC    ; SET ERR VEC PTR.
020364  005737  177572                  TST     @#SR0           ; GOT KT11?
020370  000240                          NOP                     ;  (TRAP IF NO).
020372  013737  002120  003134          MOV     L$HIME,KTFLG    ; YES. SET KT FLAG.
020400  022737  010000  003134          CMP     #10000,KTFLG    ; MORE THAN 18 BITS?
020406  100404                          BMI     4$              ; NO
020410  042737  003777  003134          BIC     #3777,KTFLG     ;
020416  000403                          BR      5$              ;
020420  042737  000177  003134  4$:     BIC     #177,KTFLG      ;
020426  010037  000004          5$:     MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
020432  005000                          CLR     R0              ; R0 = AR DATA.
020434  012701  172340                  MOV     #KIPAR0,R1      ; R1 = KI REGS PTR.
020440  012761  077406  177740  1$:     MOV     #77406,-40(R1)  ; SET DESCRIPTOR REG.
020446  010021                          MOV     R0,(R1)+        ; SET KIPAR REG.
020450  062700  000200                  ADD     #200,R0         ; BUMP AR DATA BY "4K".
020454  020027  002000                  CMP     R0,#2000        ; AT "I/O"?
020460  001367                          BNE     1$              ; NO.
020462  012741  177600                  MOV     #177600, (R1)   ; YES. SET KTPAR7 FOR I/O.
020466  000405                          BR      9$
                                ;
020470  012716  020476          2$:     MOV     #6$,(SP)        ; SET UP RETURN
020474  000002                          RTI                     ; RTI TO NEXT LOCATION
                                ;
020476  010037  000004          6$:     MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
                                ;
020502  000207                  9$:     RTS     PC
```

```
                                        ;*
                                        ;       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.
                                        ;
                                        ;

020504                                  INVERT::
020504  013705  002206                          mov     csraddr,r5
020510  004737  016034                          jsr     pc,sofinit
020514  103406                                  bcs     25$
020516  010001                                  mov     r0,r1
020520                                          errdf   errno,sfierr,sfimsg
020520  104455                                  TRAP    C$ERDF
020522  002261                                  .WORD   1201
020524  003654                                  .WORD   sfierr
020526  012074                                  .WORD   sfimsg
020530  104455                                                          trap c$erdf
020532                                  25$:    ckloop
020532  104406                                  TRAP    C$CLP1
020534  013737  002202  076340                  mov     unitn,t39dsw
020542  012704  076320                          mov     #t39pk2,r4
020546  004737  010662                          jsr     pc,wrtchr
020552  103406                                  bcs     50$
020554  010001                                  mov     r0,r1
020556                                          errhrd  errno,wrtmsg,sfimsg
020556  104456                                  TRAP    C$ERHRD
020560  002261                                  .WORD   1201
020562  005060                                  .WORD   wrtmsg
020564  012074                                  .WORD   sfimsg
020566  104456                                                          trap c$erhrd
020570                                  50$:    ckloop
020570  104406                                  TRAP    C$CLP1
020572  013701  075650                          mov     t39bfr+12,r1
020576  032701  000200                          bit     #bit7,r1
020602  001020                                  bne     1$
020604  012737  100206  020650  10$:    MOV     #100206,CMDPKT          ; WRT SUB-SYS MEM CMD
020612  012737  020660  020652          MOV     #WSMBK,CMDPKT+2         ; MSG BUF ADDR
020620  012737  000006  020656          MOV     #6,CMDPKT+6             ; BYTE COUNT
020626  012737  100010  020660          MOV     #100010,WSMBK           ; INVERT THE SWITCH
020634  012704  020650                  MOV     #CMDPKT,R4              ; SET CMDPKT INTO R4
020640  004737  010662                  JSR     PC,WRTCHR               ; DO IT
020644  000207                  1$:     RTS     PC                      ; RETURN

                                        ;       COMMAND PACKET.

                020650                  .       =       <.+3>&177774    ;MUST BE ON MOD 4 BOUNDRY.

020650  000000                  CMDPKT:: 0                              ;1ST WORD IS TSO5 COMMAND.
020652  000000                            0                             ;2ND WORD IS THE BUFFER LOW ADDRESS.
020654  000000                            0                             ;3RD WORD IS THE BUFFER HIGH ADDRESS.
```

```
KTINIT    SETUP KT11 MEMORY MANAGEMENT REGISTERS

3646 020656  000000                          0                       ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
3647
3648                                ;       WRITE SUB SYSTEM MEMORY CHARACTERISTIC BLOCK.
3649
3650 020660  000000                WSMBK:: 0                       ;1ST WORD:: SEL 0
3651 020662  000000                        0                       ;2ND WORD:: SEL 2
3652 020664  000000                        0                       ;3RD WORD:: SEL 4
3653                                        .EVEN
3654                                ;*
3655                                ;       SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
3656                                ;
3657                                ;
3658                                ;INPUTS:
3659                                ;OUTPUTS:
3660                                ;       The NXMFLG is set if we can test.
3661                                ;       The NXMLO and NXMHI addresses are setup.
3662                                ;-
3663
3664 020666                         MEMCK::
3665
3666 020666                                 SAVREG                          ;SAVE THE REGISTERS
3667 020672  005037  003140                 CLR     NXMFLG                  ;CLEAR THE FLAG
3668 020676  005037  003142                 CLR     NXMLO                   ;CLEAR THE TEST ADDRESS LO
3669 020702  005037  003144                 CLR     NXMHI                   ;CLEAR THE TEST ADDRESS HI
3670 020706  005737  003150                 TST     T23B                    ;IS IT A 11/23B?
3671 020712  001407                       BEQ     1$                      ;NO
3672 020714  023727  002120  007777       CMP     L$HIME,#7777            ; GREATER THAN 128K
3673 020722  103406                       BLO     2$                      ; NO
3674 020724  004737  021042               JSR     PC,NXMTST               ;SETUP THE ADDRESS
3675 020730  000427                       BR      13$                     ;SET THE FLAG AND EXIT
3676 020732  005737  003146         1$:   TST     T23A                    ;IS IT A 11/23A?
3677 020736  001413                       BEQ     4$                      ;NO
3678 020740  023727  002120  005777 2$:   CMP     L$HIME,#5777            ;GREATER THAN 96K
3679 020746  101023                       BHI     14$                     ;YES,23A/23B WITH 128K MEMORY
3680 020750  023727  002120  003777       CMP     L$HIME,#3777            ;GREATER THAN 64K BUT LESS THAN 92K?
3681 020756  103403                       BLO     4$                      ;NO, CHECK 24K
3682 020760  004737  021042               JSR     PC,NXMTST               ;SETUP THE ADDRESS
3683 020764  000411                       BR      13$                     ;SET THE FLAG AND EXIT
3684 020766  023727  002120  001577 4$:   CMP     L$HIME,#1577            ;GREATER THAN 24K BUT LESS THAN 64K?
3685 020774  103410                       BLO     14$                     ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
3686 020776  004737  021042               JSR     PC,NXMTST               ;SETUP THE ADDRESS
3687 021002  062737  000077  003144       ADD     #77,NXMHI               ;FOOL THE 11/02 & 11/03
3688 021010  005237  003140         13$:  INC     NXMFLG                  ;SET THE FLAG
3689 021014  000411                       BR      15$                     ;EXIT
3690 021016  000410                 14$:  BR      15$                     ;NOP FOR PRINTOUT
3691 021020                               PRINTF  #NOMEM                  ;TELL THEM & EXIT ***NO PRINT*****
     021020  012746  005462               MOV     #NOMEM,-(SP)
     021024  012746  000001               MOV     #1, (SP)
     021030  010600                       MOV     SP,R0
     021032  104417                       TRAP    C$PNTF
     021034  062706  000004               ADD     #4,SP
3692 021040  000207                 15$:  RTS     PC                      ;RETURN
3693
3694                                ;*
3695                                ;       SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
3696                                ;
3697                                ;OUTPUTS:NXMLO,NXMHI            ;SETUP WITH NXM ADDRESS
```

E8

KTINIT SETUP KT11 MEMORY MANAGEMENT REGISTERS

```
3698                                    ;
3699                                    ;-
3700
3701 021042  013701  002120            NXMTST: MOV     L$HIME,R1               ;GET TOP OF MEMORY
3702 021046  062701  000200                    ADD     #200,R1                 ;MAKE IT I/O BLOCK OR OTHER NXM
3703 021052  042701  000177                    BIC     #177,R1
3704 021056  010102                            MOV     R1,R2                   ;RESAVE RESULTS
3705         000006                            .REPT   6
3706                                            ASL     R1                      ;PUT IN PLACE FOR XFER
3707                                            .ENDR
3708 021074  010137  003142                    MOV     R1,NXMLO                ;SAVE TEST ADDRESS LOW
3709         000012                            .REPT   10.
3710                                            ASR     R2                      ;PUT IN PLACE FOR XFER
3711                                            .ENDR
3712 021124  042702  177700                    BIC     #177700,R2              ;DON'T WANT ILA!
3713 021130  010237  003144                    MOV     R2,NXMHI                ;SAVE TEST ADDRESS HIGH
3714 021134  000207                            RTS     PC                      ;RETURN
3715
3716 021136                                    ENDMOD
3725                                           .TITLE  TSV4   MISCELLANEOUS SECTIONS
3726
3727 021136                                    BGNMOD  TSV4
     021136                            TSV4::
3728
3734
3735
3736
3737                                           .SBTTL  PROTECTION TABLE
3738 021136                                    BGNPROT
     021136                            L$PROT::
3739 021136  177777  177777  177777            .WORD    1,  1,  1,  1          ;NO DEVICE PROTECTION REQUIRED.
3740 021146                                    ENDPROT
```

F8

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
                                ;-
021146                                  BGNINIT
021146                          L$INIT::
021146  005037  002226          40$:    CLR     EXTFEA
021152  005037  003140                  CLR     NXMFLG
021156  012737  006362  002200          MOV     #EPRT1,EPRTSW           ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
021164  005037  003156                  CLR     SIFLAG                  ;CLEAR "SOFT INIT" FLAG
021170  005037  003136                  CLR     KTENABLE                ;CLEAR TEST ABOVE 28K FLAG
021174  005037  002304                  CLR     RAMSIZ                  ;CLEAR RAM SIZE FOR RAMERR ROUTINE
021200                                  READEF  #EF.CONTINUE
021200  012700  000036                  MOV     #EF.CONTINUE,R0
021204  104447                          TRAP    C$REFG
021206                                  BNCOMPLETE 1$
021206  103023                          BCC     1$
021210  023737  002202  002012          CMP     UNITN,L$UNIT            ;UNIT IN RANGE?
021216  103070                          BHIS    4$                      ;BR IF NO.
021220  005737  003114                  TST     DUFLG                   ;DROPPED UNIT?
021224  100472                          BMI     NXTU                    ;BR IF YES
021226  013701  002202                  MOV     UNITN,R1
021232  006301                          ASL     R1
021234  005761  003200                  TST     ERTABL(R1)
021240  001516                          BEQ     SETU
021242  032761  040000  003200          BIT     #BIT14,ERTABL(R1)       ;DROPPED?
021250  001060                          BNE     NXTU
021252                                  EXIT    INIT                    ;DO NOTHING IF "CONTINUE".
021252  104432                          TRAP    C$EXIT
021254  000416                          .WORD   L10030 .
021256                          1$:     READEF  #EF.NEW
021256  012700  000035                  MOV     #EF.NEW,R0
021262  104447                          TRAP    C$REFG
021264                                  BNCOMPLETE NXTU                 ;TAKE NEXT UNIT IF NOT NEW PASS.
021264  103052                          BCC     NXTU
021266                                  READEF  #EF.START
021266  012700  000040                  MOV     #EF.START,R0
021272  104447                          TRAP    C$REFG
021274                                  BCOMPLETE 2$
021274  103404                          BCS     2$
021276                                  READEF  #EF.RESTART
021276  012700  000037                  MOV     #EF.RESTART,R0
021302  104447                          TRAP    C$REFG
021304                                  BNCOMPLETE 31$
021304  103031                          BCC     31$
021306                          2$:                                     ;1ST PASS, BUS INIT...
021306                                  BRESET                          ;BUS RESET.
021306  104433                          TRAP    C$RESET
```

INITIALIZE SECTION

```
3783 021310  005037  002214                CLR     TSTCNT                  ;NUMBER OF TESTS RUN IN PASS
3784 021314  005037  002222                CLR     FATFLG                  ;CLEAR FATAL ERROR COUNT
3785 021320  005037  003146                CLR     T23A                    ;CLEAR 11/23A FLAG
3786 021324  005037  003150                CLR     T23B                    ;CLEAR 11/23B FLAG
3787                              ;         MOV     #340, (SP)
3788                              ;         MOV     #20$, (SP)              ;RETURN TO DEBUGGER
3789                              ;         JMP     O.ODT           ;;@@ENTER THE DEBUGGER
3790 021330  005037  003402                CLR     SKIPT           ;CLEAR THE SUBTEST "SKIPPER"
3791 021334                       20$:
3792 021334  012737  177777  002204        MOV     #1,QVP                  ;...QUICK VERIFY...
3793 021342  004737  020152                JSR     PC,ENVIRN               ;SET ENVIRONMENT.
3794 021346  004737  020332                JSR     PC,KTINIT               ;INITIALIZE KT MEMORY MANAGEMENT
3795 021352  012700  003200                MOV     #ERTABL,R0
3796 021356  005020                30$:    CLR     (R0)+                   ;CLEAR THE ERROR TABLE
3797 021360  020027  003400                CMP     R0,#ERTABE
3798 021364  103774                       BLO     30$
3799 021366  000404                       BR      4$
3800 021370  005037  002204        31$:    CLR     QVP
3801 021374  000137  021444                JMP     PASRPT                  ;GO REPORT THE STATUS
3802
3803 021400                       4$:
3804 021400  012737  177777  002202  NEWPAS: MOV     #-1,UNITN               ;INIT UNIT NUMBER...
3805 021406  005037  002220                CLR     DEVCNT                  ;CLEAR COUNT OF DEVICES RUNNING
3806 021412                       NXTU:   BREAK
     021412  104422                       TRAP    C$BRK
3807 021414  005237  002202                INC     UNITN                   ;...AND SET NEXT UNIT NUMBER.
3808 021420  023737  002202  002012        CMP     UNITN,L$UNIT
3809 021426  103423                       BLO     SETU
3810 021430  012737  177777  003114        MOV     #1,DUFLG
3811 021436  000401                       BR      11$
3812 021440                               DOCLN                           ;ABORT, NO MORE UNITS.
     021440  104444                       TRAP    C$DCLN
3813 021442  000240                11$:    NOP
3814 021444                       PASRPT:
3815 021444  023727  002012  000001        CMP     L$UNIT,#1               ;HOW MANY UNITS SELECTED?
3816 021452  101752                       BLOS    NEWPAS                  ;BR IF ONLY 1
3817 021454  005737  002220                TST     DEVCNT                  ;ARE ANY STILL RUNNING?
3818 021460  001747                       BEQ     NEWPAS                  ;BR IF NO
3819 021462                               RFLAGS  R0
     021462  104421                       TRAP    C$RFLA
3820 021464  032700  000100                BIT     #ISR,R0                 ;SHOULD WE PRINT STATISTICS
3821 021470  001343                       BNE     NEWPAS                  ;BR IF NO
3822
3823 021472                               DORPT
     021472  104424                       TRAP    C$DRPT
3824 021474  000741                       BR      NEWPAS
3825 021476                       10$:
3826
3827 021476                       SETU:   GPHARD  UNITN,R0                ;GET UNIT N P-TABLE POINTER.
     021476  013700  002202                MOV     UNITN,R0
     021502  104442                       TRAP    C$GPHRD
3828 021504                               BNCOMPLETE NXTU                 ;BR IF UNIT NOT AVAILABLE.
     021504  103342                       BCC     NXTU
3829 021506  005037  003114                CLR     DUFLG                   ;CLEAR "DROPPED" FLAG.
3830 021512  005237  002220                INC     DEVCNT
3831 021516  012001                       MOV     (R0)+,R1                ;GET 1ST REGISTER ADDRESS.
3832 021520  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.
021526  010137  002210                          MOV     R1,IVEC                 ;SET INTERRUPT VECTOR POINTER...
021532  012721  016256                          MOV     @INTR,(R1)+             ;...VECTOR...
021536  013721  002212                          MOV     IPRI,(R1)+              ;...AND PRIORITY.

021542                                  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.
                                        ;
021542  013701  002202                          MOV     UNITN,R1
021546  006301                                  ASL     R1
021550  052761  100000  003200                  BIS     @BIT15,ERTABL(R1)       ;SAY DEVICE RUNNING
021556  005037  005774                          CLR     EXTA                    ;CLEAR ERROR EXTENSION FLAG.
021562  023727  002012  000001                  CMP     L$UNIT,@1               ;ARE WE TESTING MULTIPLE UNITS?
021570  101416                                  BLOS    10$                     ;BR IF NO.
021572                                          RFLAGS  R0                      ;YES -- GET OPERATOR FLAGS.
021572  104421                                  TRAP    C$RFLA
021574  032700  001000                          BIT     @PNT,R0                 ;SHOULD WE PRINT UNIT @?
021600  001412                                  BEQ     10$                     ;BR IF NOT.
021602                                          PRINTF  @PUNIT,UNITN            ;PRINT THE UNIT @
021602  013746  002202                          MOV     UNITN,-(SP)
021606  012746  021674                          MOV     @PUNIT,-(SP)
021612  012746  000002                          MOV     @2,-(SP)
021616  010600                                  MOV     SP,R0
021620  104417                                  TRAP    C$PNTF
021622  062706  000006                          ADD     @6,SP
021626                                  10$:
021626  005037  003116                          CLR     NODEV
021632  013701  002206                          MOV     CSRADDR,R1              ;ADDRESS OF FIRST REGISTER
021636  010102                                  MOV     R1,R2                   ;START OF REGISTERS
021640  062702  000002                          ADD     @TSSR,R2                ;ADDRESS OF TSSR REGISTER
021644  004737  016436                          JSR     PC,XNXM                 ;TEST BOTH CONTROLLER REGISTERS...
021650  103005                                  BCC     2$                      ;...AND BR IF ALL OK.
021652  010137  003116                          MOV     R1,NODEV                ;FLAG DEVICE AS NON-EXISTENT
021656  012737  177777  003114                  MOV     @-1,DUFLG               ;DROP THIS UNIT.
021664                                  2$:
                                        ;
                                        ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
                                        ;
021664                                  5$:     SETPRI  @PRI00          ;ENABLE INTERRUPTS.
021664  012700  000000                          MOV     @PRI00,R0
021670  104441                                  TRAP    C$SPRI
021672                                          ENDINIT
021672                                  L10030:
021672  104411                                  TRAP    C$INIT

021674     045     116     045  PUNIT:  .ASCIZ  /%N%N%A***** TESTING UNIT %D2%A *****/
                                        .EVEN
```

ADD AND DROP UNITS SECTIONS

```
                                        .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.
                                ; -
021742                                  BGNAU
021742                          L$AU::
021742  010001                          MOV     R0,R1                   ; GET UNIT TO BE ADDED (R0)
021744  006301                          ASL     R1                      ; MAKE IT A WORD INDEX
021746  052761  100000  003200          BIS     #100000,ERTABL(R1)      ; SET THE "ACTIVE" BIT
021754  042761  040000  003200          BIC     #40000,ERTABL(R1)       ; CLEAR THE "DROPPED" BIT
021762                                  PRINTF  #1$,R0
021762  010046                          MOV     R0,-(SP)
021764  012746  022010                  MOV     #1$,-(SP)
021770  012746  000002                  MOV     #2,-(SP)
021774  010600                          MOV     SP,R0
021776  104417                          TRAP    C$PNTF
022000  062706  000006                  ADD     #6,SP
022004                                  EXIT    AU
022004  000167                          .WORD   J$JMP
022006  000026                          .WORD   L10031 2-.
022010     045     116     045  1$:     .ASCIZ  /%N%A  UNIT %D%A ADDED/
                                        .EVEN

                                        ENDAU           ; UNUSED.
022036                          L10031:
022036  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.

022040                                  BGNDU
022040                          L$DU::
022040  012737  177777  003114          MOV     #-1,DUFLG
022046  010001                          MOV     R0,R1
022050  006301                          ASL     R1
022052  052761  140000  003200          BIS     #140000,ERTABL(R1)      ; SAY DROPPED
022060  000240  000240  000240          240,240,240                     ; ?????????
022066                                  PRINTF  #1$,R0
022066  010046                          MOV     R0,-(SP)
022070  012746  022114                  MOV     #1$, (SP)
022074  012746  000002                  MOV     #2,-(SP)
022100  010600                          MOV     SP,R0
022102  104417                          TRAP    C$PNTF
022104  062706  000006                  ADD     #6,SP
022110                                  EXIT    DU
022110  000167                          .WORD   J$JMP
022112  000030                          .WORD   L10032-2 .
```

J8


```
ADD AND DROP UNITS SECTIONS

 3915 022114      045     116     045  1$:     .ASCIZ  /%N%A  UNIT %D%A DROPPED/
 3916                                          .EVEN
 3917 022144                                   ENDDU
      022144                             L10032:
      022144  104453                           TRAP    C$DU
 3918                                   ;**
 3919                                   ; AUTO DROP CODE SECTION.
 3920                                   ;
 3921 022146                                   BGNAUTO
      022146                             L$AUTO::
 3922 022146  013705  002206                    MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
 3923 022152  012703  000550                    MOV     #360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
 3924 022156  004737  016310             10$:   JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 3925 022162  103420                            BCS     20$                     ;LEAVE WHEN SSR IS SET
 3926 022164                                    DELAY   250.                    ;WAIT FOR .25 SECONDS
      022164  012727  000372                    MOV     #250.,(PC)+
      022170  000000                            .WORD   0
      022172  013727  002116                    MOV     L$DLY,(PC)+
      022176  000000                            .WORD   0
      022200  005367  177772                    DEC     -6(PC)
      022204  001375                            BNE     .-4
      022206  005367  177756                    DEC     -22(PC)
      022212  001367                            BNE     .-20
 3927 022214  005303                            DEC     R3                      ;BUMP COUNTER DOWN
 3928 022216  001357                            BNE     10$                     ;KEEP GOING
 3929 022220  004737  017242                    JSR     PC,CKDROP               ;TRY AND DROP UNIT
 3930 022224                             20$:
 3931 022224                                   ENDAUTO                 ; UNUSED.
      022224                             L10033:
      022224  104461                           TRAP    C$AUTO
```

```
                                        .SBTTL  CLEAN UP AND REPORT CODING SECTIONS

                                ;**
                                ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
                                ; EXECUTED AT THE END OF EACH PASS (OR SUB PASS).
                                ; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
                                ;--
022226                                  BGNCLN
022226                          L$CLEAN::
022226  013705  002206                  MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
022232  005737  003114                  TST     DUFLG           ;"DROPPED" FLAG IS SET ON...
022236  100405                          BMI     1$              ;...AND GROSS CONTROLLER FAULT...
                                                                ;...DON'T TRY TO XCT CLEANUP CODE.

022240  012765  000000  000002          MOV     #0,TSSR(R5)     ;DO SOFT INIT
022246  004737  016310                  JSR     PC,WAITF
022252                          1$:
022252                          2$:     ENDCLN
022252                          L10034:
022252  104412                          TRAP    C$CLEAN
                                ;**
                                ; THE REPORT CODING SECTION CONTAINS THE
                                ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;
022254                                  BGNRPT
022254                          L$RPT::
022254                                  PRINTS  #DEVSUM
022254  012746  022516                  MOV     #DEVSUM,-(SP)
022260  012746  000001                  MOV     #1, (SP)
022264  010600                          MOV     SP,R0
022266  104416                          TRAP    C$PNTS
022270  062706  000004                  ADD     #4,SP
022274  010246                          MOV     R2, (SP)
022276  010346                          MOV     R3,-(SP)
022300  010446                          MOV     R4,-(SP)
022302  012704  003200                  MOV     #ERTABL,R4      ; GET START OF ERROR TABLE.
022306  005003                          CLR     R3              ; CLEAR UNIT NUMBER
022310  011402                  1$:     MOV     (R4),R2         ; GET ERROR TABLE ENTRY & TEST IT.
022312  001467                          BEQ     4$              ; ZERO IF UNIT NOT RUN
022314  100066                          BPL     4$
022316  032702  040000                  BIT     #BIT14,R2       ; WAS UNIT DROPPED?
022322  001015                          BNE     2$              ; BR IF YES
022324  042702  170000                  BIC     #^C7777,R2      ; GET ERROR COUNT FIELD
022330                                  PRINTS  #DEVONL,R3,R2   ; PRINT
022330  010246                          MOV     R2,-(SP)
022332  010346                          MOV     R3,-(SP)
022334  012746  022553                  MOV     #DEVONL,-(SP)
022340  012746  000003                  MOV     #3,-(SP)
022344  010600                          MOV     SP,R0
022346  104416                          TRAP    C$PNTS
022350  062706  000010                  ADD     #10,SP
022354  000446                          BR      4$
022356  020227  160000          2$:     CMP     R2,#160000      ; WAS UNIT NON-EXISTENT?
022362  001012                          BNE     3$              ; BR IF NO
022364                                  PRINTS  #DEVNXR,R3
022364  010346                          MOV     R3,-(SP)
022366  012746  022623                  MOV     #DEVNXR,-(SP)
```

CLEAN-UP AND REPORT CODING SECTIONS

```
        022372  012746  000002                  MOV     #2, (SP)
        022376  010600                          MOV     SP,R0
        022400  104416                          TRAP    C$PNTS
        022402  062706  000006                  ADD     #6,SP
  3972  022406  000431                          BR      4$
  3973  022410  020227  160001          3$:     CMP     R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
  3974  022414  001012                          BNE     30$             ; BR IF NO.
  3975  022416                                  PRINTS  #DEVNRD,R3
        022416  010346                          MOV     R3, (SP)
        022420  012746  022705                  MOV     #DEVNRD, (SP)
        022424  012746  000002                  MOV     #2,-(SP)
        022430  010600                          MOV     SP,R0
        022432  104416                          TRAP    C$PNTS
        022434  062706  000006                  ADD     #6,SP
  3976  022440  000414                          BR      4$
  3977  022442  042702  170000          30$:    BIC     #^C7777,R2
  3978  022446                                  PRINTS  #DEVDRO,R3,R2
        022446  010246                          MOV     R2, (SP)
        022450  010346                          MOV     R3,-(SP)
        022452  012746  022766                  MOV     #DEVDRO,-(SP)
        022456  012746  000003                  MOV     #3,-(SP)
        022462  010600                          MOV     SP,R0
        022464  104416                          TRAP    C$PNTS
        022466  062706  000010                  ADD     #10,SP
  3979  022472  062704  000002          4$:     ADD     #2,R4
  3980  022476  005203                          INC     R3
  3981  022500  020427  003400                  CMP     R4,#ERTABE
  3982  022504  103701                          BLO     1$
  3983  022506  012604                          MOV     (SP)+,R4
  3984  022510  012603                          MOV     (SP)+,R3
  3985  022512  012602                          MOV     (SP)+,R2
  3986  022514                                  ENDRPT          ; UNUSED.
        022514                          L10035:
        022514  104425                          TRAP    C$RPT
  3987
  3988  022516      045     116     045 DEVSUM: .ASCIZ  /%N%ADEVICE STATUS SUMMARY:%N/
  3989  022553      045     101     040 DEVONL: .ASCIZ  /%A  UNIT %D3%A    ONLINE,  ERRORS = %D%N/
  3990  022623      045     101     040 DEVNXR: .ASCIZ  /%A  UNIT %D3%A    DROPPED, NON EXISTENT REGISTER%N/
  3991  022705      045     101     040 DEVNRD: .ASCIZ  /%A  UNIT %D3%A    DROPPED, NOT READY AT STARTUP%N/
  3992  022766      045     101     040 DEVDRO: .ASCIZ  /%A  UNIT %D3%A    DROPPED, ERRORS = %D%N/
  3993                                        .EVEN
  3994
  3995  023036                                ENDMOD
  3996
```

CLEAN UP AND REPORT CODING SECTIONS

```
4000                                            .TITLE  TSV5   HARDWARE TESTS
4001
4008
4009 023036                                     BGNMOD  TSV5
     023036                             TSV5::
4015
4023
```

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

```
                                        .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
                                ;--

023036                                  BGNTST
023036                                                                          T1::
023036  012700  023472                  MOV     #TST13ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
023042  004737  016550                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
023046  012737  000012  002216          MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
023054                          T13LOOP:
023054  004737  023746                  JSR     PC,T13REST              ;SET PACKET TO START-UP VALUES

023060  012703  002766                  MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
023064  012704  023430                  MOV     #T13PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
023070  012764  000010  000006          MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
023076                          5$:
023076  004737  016034                  JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
023102  103405                          BCS     10$                     ;BR IF SOFT INIT OKAY
023104  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
023106                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
023106  104455                                                                  TRAP    C$ERDF
023110  000144                                                                  .WORD   100
023112  003654                                                                  .WORD   SFIERR
023114  012074                                                                  .WORD   SFIMSG

                                ;Do WRITE CHARACTERISTICS command.
023116  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
023122  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
023126  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
023132                                  FORCERROR       12$                     ;aaDFORCE ERROR IF FORCER=1
023146  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
023150  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
023152                                  NEXT.ERRNO
023152                          12$:    ERRDF   ERRNO,T13SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
023152  104455                                                                  TRAP    C$ERDF
023154  000145                                                                  .WORD   101
023156  023657                                                                  .WORD   T13SSR
023160  012106                                                                  .WORD   PKTSSR
023162  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
023166                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

```
TEST  1: INITIALIZE AFTER WRITE CHARACTERISTICS

      023166  104406                                                                  TRAP    C$CLP1
 4078 023170  016501  000002          MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 4079 023174  012702  000200          MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
 4080 023200  032701  000100          BIT     #OFL,R1                 ;IS OFF LINE BIT SET ?
 4081 023204  001402                  BEQ     25$                     ;BRANCH IF NOT OFF LINE
 4082 023206  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 4083
 4084                         ;If the NBA bit in the TSSR register is NOT=0 then Print Error.
 4085 023212                  25$:
 4086 023212                          FORCERROR       27$                     ;@@D
 4087 023226  020201                  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 4088 023230  001404                  BEQ     30$                     ;OKAY IF MATCH
 4089 023232                          NEXT.ERRNO
 4090 023232                  27$:    ERRHRD  ERRNO,T13NBA,PKTSSR     ;NBA NOT ZERO
      023232  104456                                                                  TRAP    C$ERHRD
      023234  000146                                                                  .WORD   102
      023236  023604                                                                  .WORD   T13NBA
      023240  012106                                                                  .WORD   PKTSSR
 4091 023242                  30$:    CKLOOP                          ;LOOP ON ERROR ?
      023242  104406                                                                  TRAP    C$CLP1
 4092
 4093                         ;Write to TSSR register to soft initialize the controller
 4094 023244                  40$:
 4095 023244  004737  016034          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
 4096 023250                          FORCERROR       42$                     ;@@D
 4097 023264  103405                  BCS     50$                     ;BR IF SOFT INIT OKAY
 4098 023266  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4099 023270                          NEXT.ERRNO
 4100 023270                  42$:    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      023270  104455                                                                  TRAP    C$ERDF
      023272  000147                                                                  .WORD   103
      023274  003654                                                                  .WORD   SFIERR
      023276  012074                                                                  .WORD   SFIMSG
 4101
 4102                         ;If controller RAM 310-377 NOT=0 then Print Error
 4103 023300  012704  000310  50$:    MOV     #310,R4                 ;START WITH LOC 310
 4104 023304  005002                  CLR     R2                      ;MEMORY EXPECTED SHOULD BE 000000
 4105 023306  105065  000000          CLRB    TSDB(R5)                ;SET MAINTENANCE MODE
 4106 023312  004737  016376          JSR     PC,CHKTSSR              ;WAIT FOR SSR READY
 4107 023316  010465  000000  60$:    MOV     R4,TSDB(R5)             ;SELECT RAM ADDRESS
 4108 023322  004737  016376          JSR     PC,CHKTSSR              ;WAIT FOR SSR READY
 4109 023326  116501  000000          MOVB    TSBA(R5),R1             ;READ LOC CONTENTS
 4110 023332                          FORCERROR       62$,NOTSSR              ;@@D
 4111 023342  120102                  CMPB    R1,R2                   ;CHECK MEMORY FOR 000000
 4112 023344  001406                  BEQ     70$                     ;BRANCH IF DATA OKAY
 4113 023346                          NEXT.ERRNO
 4114 023346                  62$:    ERRDF   ERRNO,T13MEM,RAMEXP     ;MEMORY NOT ZERO AFTER INIT.
      023346  104455                                                                  TRAP    C$ERDF
      023350  000150                                                                  .WORD   104
      023352  023545                                                                  .WORD   T13MEM
      023354  015570                                                                  .WORD   RAMEXP
 4115 023356  005237  002222          INC     FATFLG                  ;SET THE FATAL ERROR FLAG
 4116 023362                  70$:    CKLOOP
      023362  104406                                                                  TRAP    C$CLP1
 4117 023364                          ESCAPE  TST                     ;EXIT ON FATAL ERROR
      023364  104410                                                                  TRAP    C$ESCAPE
      023366  000426                                                                  .WORD   L10036-.
```

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

```
4118
4119 023370  005204                  82$:    INC     R4                      ;LOOK AT NEXT RAM LOC.
4120 023372  020427  000400                  CMP     R4,#400                 ;AT TOP OF RAM ADDRESS SPACE
4121 023376  001347                          BNE     60$                     ;BRANCH TILL ALL MEMORY TESTED
4122
4123
4124 023400  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
4125 023404  001402                          BEQ     160$                    ;BRANCH IF NOT
4126 023406  004737  017242                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4127 023412  004737  016516          160$:   JSR     PC,TSTLOOP              ;DONE ALL ITERATIONS?
4128 023416  103002                          BCC     165$                    ;BR IF YES
4129 023420  000137  023054                  JMP     T13LOOP                 ;LOOP UNTIL ITERATION COUNT DONE
4130 023424                          165$:
4131 023424                          EXIT    TST
     023424  104432                                                          TRAP    C$EXIT
     023426  000366                                                          .WORD   L10036 .
4132
4133
4134                                 ;+
4135                                 ;LOCAL STORAGE FOR THIS TEST
4136                                 ;-
4137
4141 023430                          T13PACKET:                              ;COMMAND PACKET FOR TEST
4142 023430  100004                          .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
4143 023432  023440                          .WORD   T13DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
4144 023434  000000                          .WORD   0
4145 023436  000010                          .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
4146
4147 023440                          T13DATA:                                        ;CHARACTERISTICS DATA BLOCK
4148 023440  023452                          .WORD   T13BFR                  ;ADDRESS OF MESSAGE BUFFER
4149 023442  000000                          .WORD   0
4150 023444  000016                          .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
4151 023446  000000  000000                  .WORD   0,0
4152
4153 023452                          T13BFR: .BLKW   8.                      ;MESSAGE BUFFER
4154                                 ;LOCAL TEXT MESSAGES FOR TEST
4155                                 ;-
4156
4157 023472     111     156     151  TST13ID:        .ASCIZ  'Initialization After WRITE CHARACTERISTICS'
4158 023545     111     156     143  T13MEM: .ASCIZ  'Incorrect RAM Data After Init'
4159                                         .EVEN
4160 023604     127     122     111  T13NBA: .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
4161 023657     103     157     156  T13SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
4162
4163
4164                                 ;+
4165                                 ;
4166                                 ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
4167                                 ;
4168                                 ;-
4169
4170                                 .EVEN
4171
4172 023746                          T13REST:
4173 023746                                  SAVREG                          ;SAVE THE REGISTERS
4174 023752  012701  023430                  MOV     #T13PACKET,R1           ;START OF THE PACKET
4175 023756  012721  100004                  MOV     #100004,(R1)+           ;WRITE CHARACTERISTICS WITH ACK
```

```
TEST  1: INITIALIZE AFTER WRITE CHARACTERISTICS

 4176 023762  012721  023440                  MOV     #T13DATA,(R1)+  ;ADDRESS OF CHAR DATA BLOCK
 4177 023766  005021                          CLR     (R1)+           ;EXTENDED ADDRESS
 4178 023770  012721  000010                  MOV     #8.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
 4179 023774  012721  023452                  MOV     #T13BFR,(R1)+   ;ADDRESS OF MESSAGE BUFFER
 4180 024000  005021                          CLR     (R1)+
 4181 024002  012721  000016                  MOV     #14.,(R1)+      ;LENGTH OF MESSAGE BUFFER
 4182 024006  005021                          CLR     (R1)+
 4183 024010  005011                          CLR     (R1)
 4184 024012  000207                          RTS     PC              ;RETURN
 4185 024014                                  ENDTST
      024014                                                                  L10036:
      024014  104401                                                                 TRAP    C$ETST
 4186
 4187                                         .SBTTL  TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND
 4188                               ;+
 4189                               ;
 4190                               ;THIS TEST VERIFIES THAT THE WRITE SUBSYSTEM MEMORY COMMAND WITH A
 4191                               ;BSELO SELECT CODE OF 0 (NO-OP) EXECUTES CORRECTLY. IT ALSO
 4192                               ;VERIFIES THAT A WRITE SUBSYSTEM MEMORY COMMAND WITH A NON ZERO
 4193                               ;MODE FIELD IS REJECTED. THE TEST FURTHER VERIFIES MICROPROGRAM
 4194                               ;COMMAND DECODING AND HANDLING SEQUENCES.
 4195                               ;
 4196                               ;-
 4197 024016                                  BGNTST
      024016                                                                  T2::
 4202 024016  012700  025631                  MOV     #TST14ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
 4203 024022  004737  016550                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
 4204 024026  012737  000024  002216          MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
 4205 024034                       T14LOOP:
 4206 024034                                  BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
      024034                                                                  T2.1:
      024034  104402                                                                 TRAP    C$BSUB
 4207 024036  004737  025676                  JSR     PC,T14REST              ;SET PACKET TO INITIAL VALUES
 4208 024042  004737  025734                  JSR     PC,T14RST               ;SET PACKET TO INITIAL VALUES
 4209 024046                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      024046  012700  000000                                                          MOV     #PRI00,R0
      024052  104441                                                                 TRAP    C$SPRI
 4210 024054                            5$:
 4211 024054                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      024054  104404                                                                 TRAP    C$BSEG
 4212 024056  004737  016034                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 4213 024062  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
 4217 024064  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4218 024066                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      024066  104455                                                                 TRAP    C$ERDF
      024070  000311                                                                 .WORD   201
      024072  003654                                                                 .WORD   SFIERR
      024074  012074                                                                 .WORD   SFIMSG
 4219 024076                            10$:
 4220 024076  012704  025170                  MOV     #T14PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
 4221 024102  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 4222 024106  103405                          BCS     11$                     ;BR, IF COMMAND ISSUED OK
 4226 024110  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4227 024112                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      024112  104456                                                                 TRAP    C$ERHRD
      024114  000312                                                                 .WORD   202
      024116  005060                                                                 .WORD   WRTMSG
```

E9

SEQ 0108

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
        024120  012074                                                                  .WORD   SFIMSG
4228    024122                  11$:
4229    024122  005037  002222          CLR     FATFLG          ;CLEAR FATAL ERROR FLAG
4230    024126  005037  002224          CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
4231    024132  012704  024550          MOV     #T14PACKET,R4   ;SET UP NEW WRT. SUBSYS MEM. COMMAND
4232    024136  010465  000000          MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
4233    024142  004737  016376          JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
4234    024146  103407                  BCS     15$             ;BR IF CARRY SET (GOOD RETURN)
4235    024150  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
4239    024152                          ERRDF   ERRNO,T14SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        024152  104455                                                                 TRAP    C$ERDF
        024154  000313                                                                  .WORD   203
        024156  025363                                                                  .WORD   T14SSR
        024160  012106                                                                  .WORD   PKTSSR
4240    024162  005237  002222          INC     FATFLG          ;SET FATAL ERROR FLAG
4241    024166                  15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
        024166  104406                                                                 TRAP    C$CLP1
4242    024170                          ESCAPE  SEG             ;BY PASS SUBTEST IF FATAL ERROR
        024170  104410                                                                 TRAP    C$ESCAPE
        024172  000074                                                                  .WORD   10000$ .
4243    024174  005737  002224          TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
4244    024200  001004                  BNE     22$             ;BRANCH IF YES
4248    024202                          ERRHRD  ERRNO,T14NINT,PKTSSR
        024202  104456                                                                 TRAP    C$ERHRD
        024204  000314                                                                  .WORD   204
        024206  025453                                                                  .WORD   T14NINT
        024210  012106                                                                  .WORD   PKTSSR
4249    024212  016501  000002  22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
4250    024216  012702  000200          MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
4251    024222  032701  000100          BIT     #OFL,R1         ;IS OFF LINE BIT SET ?
4252    024226  001402                  BEQ     25$             ;BRANCH IF NOT OFF-LINE
4253    024230  052702  000100          BIS     #OFL,R2         ;SET OFF LINE IN EXPECTED DATA
4254    024234  020201          25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
4255    024236  001404                  BEQ     30$             ;OKAY IF MATCH
4259    024240                          ERRHRD  ERRNO,T14NBA,PKTSSR     ;NBA NOT ZERO
        024240  104456                                                                 TRAP    C$ERHRD
        024242  000315                                                                  .WORD   205
        024244  025212                                                                  .WORD   T14NBA
        024246  012106                                                                  .WORD   PKTSSR
4260    024250                  30$:
4261    024250  004737  011154  35$:    JSR     PC,CKRAM        ;CHECK RAM TO MEMORY
4262    024254  103405                  BCS     59$             ;RAM OK GO ON
4266    024256                          ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
        024256  104456                                                                 TRAP    C$ERHRD
        024260  000316                                                                  .WORD   206
        024262  004747                                                                  .WORD   PKTRAM
        024264  015550                                                                  .WORD   RAMERR
4267    024266                          ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        024266                                                                  10000$:
        024266  104405                                                                 TRAP    C$ESEG
4268
4269
4270    024270                  59$:    ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        024270                                                                  L10040:
        024270  104403                                                                 TRAP    C$ESUB
4271
4272    024272  005737  002222          TST     FATFLG          ;ANY FATAL ERRORS ?
```

F9

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
4273 024276  001402                        BEQ     60$                     ;BRANCH IF NOT
4274 024300  004737  017242                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4275 024304                         60$:
4276                                ;*
4277                                ;
4278                                ;TEST 2, SUBTEST 2
4279                                ;
4280                                ;CHECK THAT NON ZERO MODE BITS BEING SET CAUSES
4281                                ;WRITE SUBSYSTEM MEMORY COMMAND TO BE REJECTED
4282                                ;
4283                                ;
4284
4285 024304                         BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     024304                                                                              T2.2.
     024304  104402                                                                     TRAP    C$BSUB
4286
4287 024306                         SETPRI  @PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     024306  012700  000000                                                             MOV     @PRI00,R0
     024312  104441                                                                     TRAP    C$SPRI
4288 024314  012704  025170         5$:     MOV     @T14PK2,R4      ;GET THE ADDRESS OF COMMAND PACKET
4289 024320  004737  025676                 JSR     PC,T14REST      ;RESTORE PACKET TO STARTING VALUES
4290 024324  004737  025734                 JSR     PC,T14RST       ;RESTORE PACKET TO STARTING VALUES
4291 024330                                 BGNSEG                  ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     024330  104404                                                                     TRAP    C$BSEG
4292
4293 024332  004737  016034                 JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
4294 024336  103405                         BCS     10$             ;BR IF SOFT INIT = OK
4298 024340  010001                         MOV     R0,R1           ;SAVE CONTENTS OF TSSR
4299 024342                                 ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     024342  104455                                                                     TRAP    C$ERDF
     024344  000317                                                                     .WORD   207
     024346  003654                                                                     .WORD   SFIERR
     024350  012074                                                                     .WORD   SFIMSG
4300 024352                         10$:
4301 024352  004737  010662                 JSR     PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
4302 024356  103405                         BCS     11$             ;BR, IF COMMAND ISSUED OK
4306 024360  010001                         MOV     R0,R1           ;SAVE CONTENTS OF TSSR
4307 024362                                 ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE SUBSYSTEM MEMORY FAILED
     024362  104456                                                                     TRAP    C$ERHRD
     024364  000320                                                                     .WORD   208
     024366  005060                                                                     .WORD   WRTMSG
     024370  012074                                                                     .WORD   SFIMSG
4308 024372                         11$:
4309 024372  005037  002224                 CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
4310 024376  012704  024550                 MOV     @T14PACKET,R4   ;SET UP WITH WRT SUBSYS MEM PACKET
4311 024402  052714  007000                 BIS     @007000,(R4)    ;NON ZERO COMMAND MODE BITS
4312 024406  010465  000000                 MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
4313 024412  004737  016310                 JSR     PC,WAITF        ;WAIT FOR SSR TO SET
4314 024416  103405                         BCS     15$             ;BR IF CARRY SET (GOOD RETURN)
4315 024420  010001                         MOV     R0 R1           ;SAVE CONTENTS OF TSSR
4319 024422                                 ERRDF   ERRNO,T14SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     024422  104455                                                                     TRAP    C$ERDF
     024424  000321                                                                     .WORD   209
     024426  025363                                                                     .WORD   T14SSR
     024430  012106                                                                     .WORD   PKTSSR
4320 024432                         15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
     024432  104406                                                                     TRAP    C$CLP1
```

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
4321 024434                                ESCAPE  SEG                     ;BY PASS CHECKS IF FATAL ERROR
     024434  104410                                                                TRAP    C$ESCAPE
     024436  000074                                                                .WORD   10000$ .
4322 024440  005737  002224                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
4323 024444  001004                        BNE     22$                     ;BRANCH IF YES
4327 024446                                ERRHRD  ERRNO,T14NINT,PKTSSR
     024446  104456                                                                TRAP    C$ERHRD
     024450  000322                                                                .WORD   210
     024452  025453                                                                .WORD   T14NINT
     024454  012106                                                                .WORD   PKTSSR
4328 024456  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
4329 024462  012702  100206                MOV     #SC!SSR!TSREJ,R2        ;EXPECTED CONTENTS OF TSSR
4330 024466  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
4331 024472  001402                        BEQ     25$                     ;BRANCH IF NOT OFF LINE
4332 024474  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
4333 024500  020201          25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
4334 024502  001404                        BEQ     30$                     ;OKAY IF MATCH
4338 024504                                ERRHRD  ERRNO,T142REJ,PKTSSR    ;COMMAND NOT REJECTED
     024504  104456                                                                TRAP    C$ERHRD
     024506  000323                                                                .WORD   211
     024510  025266                                                                .WORD   T142REJ
     024512  012106                                                                .WORD   PKTSSR
4339 024514                          30$:
4340 024514  004737  011154          35$:    JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
4341 024520  103405                        BCS     59$                     ;RAM OK GO ON
4345 024522                                ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
     024522  104456                                                                TRAP    C$ERHRD
     024524  000324                                                                .WORD   212
     024526  004747                                                                .WORD   PKTRAM
     024530  015550                                                                .WORD   RAMERR
4346 024532                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     024532                                                                 10000$:
     024532  104405                                                                TRAP    C$ESEG
4347
4348 024534                          59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     024534                                                                 L10041:
     024534  104403                                                                TRAP    C$ESUB
4349 024536                                EXIT    TST                     ;ALL DONE THIS TEST
     024536  104432                                                                TRAP    C$EXIT
     024540  001246                                                                .WORD   L10037 .
4350
4351                                 ;+
4352                                 ;LOCAL STORAGE FOR THIS TEST
4353                                 ;
4354
4356         024550                          .=<.+10>&177770
4358 024550                          T14PACKET:                              ;COMMAND PACKET FOR TEST
4359 024550  100206                        .WORD   100206                  ;WRITE SUBSYSTEM MEM. COMMAND, WITH IE, ACK
4360 024552  024560                        .WORD   T14DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
4361 024554  000000                        .WORD   0
4362 024556  000006                        .WORD   6.                      ;STARTING VALUE OF BLOCK SIZE
4363 024560                          T14DATA:                                ;CHARACTERISTICS DATA BLOCK
4364 024560     000                 T14BS0: .BYTE   0                       ;BSEL0 BYTE
4365 024561     000                 T14BS1: .BYTE   0                       ;BSEL1 BYTE
4366 024562  000000                T14BS2: .WORD   0                       ;BSEL1 WORD
4367 024564  000000                        .WORD   0                       ;DATA
4368 024566                          T14BFR: .BLKW   128.                    ;MESSAGE BUFFER
```

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
4369
4370
4372            025170                          .=<.+10>&177770
4374 025170                             T14PK2:                         ;COMMAND PACKET FOR TEST
4375 025170  100204                             .WORD   100204          ;WRITE CHARA. MEM. CMND., WITH IE, ACK
4376 025172  025200                             .WORD   T14DTA          ;ADDRESS OF SELECT DATA BLOCK
4377 025174  000000                             .WORD   0
4378 025176  000010                             .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
4379
4380
4381 025200                             T14DTA:                         ;SELECT DATA BLOCK
4382 025200  024566                             .WORD   T14BFR          ;ADDRESS OF MESSAGE BUFFER
4383 025202  000000                             .WORD   0
4384 025204  000400                             .WORD   256.            ;LENGTH OF MESSAGE BUFFER
4385 025206  000000  000000                     .WORD   0,0
4386
4387
4388                                    ;+
4389                                    ;LOCAL TEXT MESSAGES FOR TEST
4390                                    ;-
4391
4392 025212     127     122     111    T14NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
4393 025266     127     122     111    T142REJ:        .ASCIZ  'WRITE SUBSYSTEM MEMORY Not Rejected With Non Zero Mode Field'
4394 025363     103     157     156    T14SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
4395 025453     105     170     160    T14NINT:        .ASCIZ  'Expected Interrupt Not Received On WRITE SUBSYSTEM MEMORY'
4396 025545     111     156     143    T14TSBA:        .ASCIZ  'Incorrect TSBA Address After WRITE SUBSYSTEM MEMORY'
4397 025631     102     141     163    TST14ID:        .ASCIZ  'Basic WRITE SUBSYSTEM MEMORY Command'
4398                                            .EVEN
4399
4400
4401                                    ;+
4402                                    ;
4403                                    ;ROUTINE TO RESTORE COMMAND PACKET TO START UP (DEFAULT) VALUES
4404                                    ;WRITE SUBSYSTEM MEMORY COMMAND
4405                                    ;
4406                                    ;-
4407
4408 025676                             T14REST:
4409 025676                                     SAVREG                  ;SAVE THE REGISTERS
4410 025702  012701  024550                     MOV     #T14PACKET,R1   ;START OF THE PACKET
4411 025706  012721  100206                     MOV     #100206,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK, IE
4412 025712  012721  024560                     MOV     #T14DATA,(R1)+  ;ADDRESS OF DATA BLOCK
4413 025716  005021                             CLR     (R1)+           ;EXTENDED ADDRESS
4414 025720  012721  000006                     MOV     #6.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
4415 025724  005021                             CLR     (R1)+           ;CLEAR BSEL0 AND BSEL1
4416 025726  005021                             CLR     (R1)+           ;CLEAR SEL2
4417 025730  005011                             CLR     (R1)            ;CLEAR DATA AREA
4418 025732  000207                             RTS     PC              ;RETURN
4419
4420
4421 025734                             T14RST:
4422 025734                                     SAVREG                  ;SAVE THE REGISTERS
4423 025740  012701  025170                     MOV     #T14PK2,R1      ;START OF THE PACKET
4424 025744  012721  100204                     MOV     #100204,(R1)+   ;WRITE CHARA. WITH ACK, IE
4425 025750  012721  025200                     MOV     #T14DTA,(R1)+   ;ADDRESS OF CHARAISTICS DATA BLOCK
4426 025754  005021                             CLR     (R1)+           ;EXTENDED ADDRESS
4427 025756  012721  000010                     MOV     #8.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
```

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
 4428 025762  012721  024566          MOV     #T14BFR,(R1)+       ;MESSAGE BUFFER ADDRESS
 4429 025766  005021                  CLR     (R1)+
 4430 025770  012721  000400          MOV     #256.,(R1)+         ;LENGTH OF MESSAGE BUFFER
 4431 025774  005021                  CLR     (R1)+
 4432 025776  005011                  CLR     (R1)
 4433 026000  005037  024566          CLR     T14BFR              ;CLEAR 1ST LOC IN MESSAGE BUFFER
 4434 026004  000207                  RTS     PC                  ;RETURN
 4435 026006                          ENDTST
      026006                                                          L10037:
      026006  104401                                                          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
                                ;
                                ;

026010                                  BGNTST
026010                                                                          T3::
026010  012700  030060                  MOV     #TST12ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
026014  004737  016550                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
026020  012737  000012  002216          MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
026026  005237  003152                  INC     T3BFLG                  ;SET TEST FLAG
026032  004737  020666                  JSR     PC,MEMCK                ;CHECK MEMORY

026036                          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
                                        ;
026036                                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
026036                                                                                  T3.1:
026036  104402                                                                            TRAP    C$BSUB

                                        ;Write to TSSR to soft initialize
026040  004737  016034                          JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
026044  103405                                  BCS     15$                     ;BR IF SOFT INIT = OK
026046                                          NEXT.ERRNO
026046  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
026050                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
026050  104455                                                                          TRAP    C$ERDF
026052  000455                                                                          .WORD   301
026054  003654                                                                          .WORD   SFIERR
026056  012074                                                                          .WORD   SFIMSG

                                        ;Do a WRITE CHARACTERISTICS to setup a message buffer
026060                                  15$:
026060  012704  027650                          MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
026064  004737  031230                          JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
026070  005037  003136                          CLR     KTENABLE                ;TURN OFF KT-11
026074  010465  000000                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
026100  004737  016376                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
026104                                          FORCERROR       17$
026120  103405                                  BCS     20$                     ;BR IF SSR SET IN CHKTSSR
026122  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
026124                                          NEXT.ERRNO
026124                                  17$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
026124  104455                                                                          TRAP    C$ERDF
026126  000456                                                                          .WORD   302
026130  030162                                                                          .WORD   T12WRTSSR
026132  012106                                                                          .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
026134  005037  002222                  20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
026140  005037  027720                          CLR     T12KT                   ;TEST ABOVE 28K SWITCH
026144  012702  027724                          MOV     #T12BLK,R2              ;POINT TO TEST PATTERN TABLE
026150                                  T121LOOP:
026150  005037  003136                          CLR     KTENABLE                ;TURN OFF ABOVE 28K TEST FLAG
026154  012201                                  MOV     (R2)+,R1                ;GET TEST PATTERN ADDRESS
026156  005000                                  CLR     R0                      ;ASSUME NO TEST ABOVE 28K
026160  005737  027720                          TST     T12KT                   ;TEST ABOVE 28K THIS TIME?
026164  001407                                  BEQ     25$                     ;BR IF NO
026166  016200  177776                          MOV     -2(R2),R0               ;GET TEST PATTERN AGAIN
026172  042700  177774                          BIC     #^C<A1716>,R0           ;SAVE 18 BIT ADDRESS ONLY
026176  012737  000001  003136                  MOV     #1,KTENABLE             ;TURN ON ABOVE 28K TEST FLAG
026204  004737  030726                  25$:    JSR     PC,T12CONVERT           ;CONVERT TEST PATTERN TO TEST ADDRESS
```

```
TEST 3: SUBTEST 1: GET STATUS SELECTED LOCATIONS

4544 026210 103034                      BCC     65$                     ;BR IF INVALID PACKET ADDRESS
4545 026212 013704 027714               MOV     T12LOADD,R4             ;COPY CURRENT PACKET LOW ADDRESS
4546 026216 013703 027712               MOV     T12HIADD,R3             ;COPY CURRENT PACKET HIGH ADDRESS
4547 026222 004737 031276               JSR     PC,T12SETGET            ;SETUP CURRENT PACKET TO GET STATUS
4548 026226 042703 177774               BIC     #^C<A1716>,R3           ;SAVE ADDRESS BITS 17-16
4549 026232 050304                      BIS     R3,R4                   ;SETUP 18 BIT PACKET ADDRESS
4550 026234 004737 017334               JSR     PC,KTOFF                ;TURN OFF KT-11
4551 026240 010465 000000               MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4552 026244 004737 016376               JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4553 026250                             FORCERROR       32$
4554 026264 103405                      BCS     40$                     ;BR IF SSR SET IN CHKTSSR
4555 026266 010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4556 026270                             NEXT.ERRNO
4557 026270                     32$:    ERRDF   ERRNO,T12GETSSR,PKTGETS ;DEVICE FATAL SSR FAILED TO SET
     026270 104455                                                                  TRAP    C$ERDF
     026272 000457                                                                  .WORD   303
     026274 030106                                                                  .WORD   T12GETSSR
     026276 012124                                                                  .WORD   PKTGETS
4558 026300                     40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     026300 104406                                                                  TRAP    C$CLP1
4559 026302                     65$:
4560 026302                             FORCEXIT        80$
4561 026312 020227 030056               CMP     R2,#T12TBE              ;DONE ALL TSTBLK TEST PATTERNS?
4562 026316 103002                      BHIS    70$                     ;BR IF YES
4563 026320 000137 026150               JMP     T121LOOP                ;DO ANOTHER MODULO  4 ADDRESS
4564 026324 005737 027720       70$:    TST     T12KT                   ;DONE ABOVE 28K TESTING TOO?
4565 026330 003012                      BGT     80$                     ;BR IF YES
4566 026332 005737 003134               TST     KTFLG                   ;ANY MEMORY ABOVE 28K ON SYSTEM?
4567 026336 001407                      BEQ     80$                     ;BR IF NO
4568 026340 012737 000001 027720        MOV     #1,T12KT                ;SET SWITCH
4569 026346 012702 027124               MOV     #T12BLK,R2              ;RESET TEST PATTERN TABLE
4570 026352 000137 026150               JMP     T121LOOP                ;DO ABOVE 28K TESTING
4571 026356 004737 017334       80$:    JSR     PC,KTOFF                ;TURN OFF KT11
4572 026362                             ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     026362                                                                     L10043:
     026362 104403                                                                  TRAP    C$ESUB
4573 026364 005737 002222               TST     FATFLG                  ;ANY FATAL ERRORS ?
4574 026370 001402                      BEQ     100$                    ;BRANCH IF NOT
4575 026372 004737 017242               JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4576 026376                     100$:
4577
4578                                    .SBTTL  TEST 3: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS
4579                            ;**
4580                            ; TEST 3: SUBTEST 2:
4581                            ;
4582                            ; SUBTEST DESCRIPTION:
4583                            ;
4584                            ;       This subtest verifies the controller can deposit message packets
4585                            ;       to all available memory locations.
4586                            ;       Write Characteristics commands are executed with message
4587                            ;       buffer addresses set to various addresses in each available
4588                            ;       memory location.
4589                            ;       The various addresses are determined by floating a 1 then a 0
4590                            ;       through the address bits.
4591                            ;
4592                            ; TEST STEPS:
4593                            ;
```

```
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
                                        ;
 026376                                    BGNSUB                          ;////////////// BEGIN SUBTEST //////////
 026376                                                                                      T3.2:
 026376  104402                                                                                     TRAP    C$BSUB

                                        ;Write to TSSR to soft initial'ze
 026400  004737  016034                            JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
 026404  103405                            BCS     15$                     ;BR IF SOFT INIT = OK
 026406                                    NEXT.ERRNO
 026406  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 026410                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
 026410  104455                                                                             TRAP    C$ERDF
 026412  000460                                                                             .WORD   304
 026414  003654                                                                             .WORD   SFIERR
 026416  012074                                                                             .WORD   SFIMSG

                                        ;Do a WRITE CHARACTERISTICS to setup a message buffer to compare
 026420                                 15$:
 026420  012704  027650                            MOV     #T12PACKET,R4   ;GET THE ADDRESS OF COMMAND PACKET
 026424  004737  031230                            JSR     PC,T12SWRT      ;SET PACKET TO WRITE CHARACTERISTICS
 026430  004737  017334                            JSR     PC,KTOFF        ;TURN OFF KT 11
 026434  010465  000000                            MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
 026440  004737  016376                            JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
 026444                                            FORCERROR       17$
 026460  103405                                    BCS     20$             ;BR IF SSR SET IN CHKTSSR
 026462  010001                                    MOV     R0,R1           ;SAVE CONTENTS OF TSSR
 026464                                            NEXT.ERRNO
 026464                                 17$:       ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
 026464  104455                                                                             TRAP    C$ERDF
 026466  000461                                                                             .WORD   305
 026470  030162                                                                             .WORD   T12WRTSSR
 026472  012106                                                                             .WORD   PKTSSR

                                        ;Get a valid modulo-4 test address
                                        ;Set the packet message buffer to the test address
                                        ;Do a WRITE CHARACTERISTICS
 026474  005037  002222                 20$:       CLR     FATFLG          ;CLEAR FATAL ERROR FLAG
 026500  012703  027724                            MOV     #T12BLK,R3      ;POINT TO TEST PATTERN TABLE
 026504                                 T122LOOP:
 026504  012301                                    MOV     (R3)+,R1        ;GET TEST PATTERN ADDRESS
 026506  010100                                    MOV     R1,R0           ;GET ADDRESS ALL "18 BITS"
 026510  042700  177774                            BIC     #177774,R0      ;LEAVE ONLY A17 AND A16
 026514  042701  000003                            BIC     #3,R1           ;GET RID OF A17 AND A16
```

TEST 3: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS

```
4641 026520  004737  030726                JSR     PC,T12CONVERT           ;CONVERT TEST PATTERN TO TEST ADDRESS
4642 026524  103402                        BCS     25$                     ;BR IF VALID MESSAGE BUFFER ADDRESS
4643 026526  000137  026624                JMP     150$                    ;GET ANOTHER TEST PATTERN TO TRY
4644 026532  012704  027650          25$:  MOV     #T12PACKET,R4           ;SET THE COMMAND PACKET ADDRESS
4645 026536  004737  031230                JSR     PC,T12SWRT              ;SETUP T12PACKET TO WRITE CHAR.
4646 026542  013737  027714  027660        MOV     T12LOADD,T12DATA        ;SETUP LOW ORDER MESSAGE BUFFER ADD.
4647 026550  013737  027712  027662        MOV     T12HIADD,T12DATA+2      ;SETUP HIGH ORDER MESSAGE BUFFER ADD.
4648 026556  004737  017334                JSR     PC,KTOFF                ;TURN OFF KT-11
4649 026562  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4650 026566  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4651 026572                                FORCERROR       32$
4652 026606  103405                        BCS     50$                     ;BR IF SSR SET IN CHKTSSR
4653 026610  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4654 026612                                NEXT.ERRNO
4655 026612                          32$:  ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
     026612  104455                                                              TRAP    C$ERDF
     026614  000462                                                              .WORD   306
     026616  030162                                                              .WORD   T12WRTSSR
     026620  012106                                                              .WORD   PKTSSR
4656 026622                          50$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     026622  104406                                                              TRAP    C$CLP1
4657 026624                          150$:
4658 026624                                FORCEXIT        160$
4659 026634  020327  030056                CMP     R3,#T12T8E              ;DONE ALL TST12BLK TEST PATTERNS?
4660 026640  103002                        BHIS    160$                    ;BR IF YES
4661 026642  000137  026504                JMP     T122LOOP                ;DO ANOTHER MODULO- 4 ADDRESS
4662 026646  004737  017334          160$: JSR     PC,KTOFF                ;TURN OFF KT11
4663 026652                                ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     026652                                                                   L10044:
     026652  104403                                                              TRAP    C$ESUB
4664 026654  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
4665 026660  001402                        BEQ     180$                    ;BRANCH IF NOT
4666 026662  004737  017242                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4667 026666                          180$:
4668
4669
4670
4671                                       .SBTTL  TEST 3: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS
4672                                ;**
4673                                ; TEST 3: SUBTEST 3:
4674                                ;
4675                                ; SUBTEST DESCRIPTION:
4676                                ;
4677                                ;       This subtest verifies the controller can fetch a
4678                                ;       Write Characteristics data block from all available
4679                                ;       memory locations.
4680                                ;       Write Characteristics commands are executed with
4681                                ;       characteristic data blocks at various memory addresses.
4682                                ;       The various memory addresses are determined by floating
4683                                ;       a 1 then a 0 through the address bits.
4684                                ;
4685                                ; TEST STEPS:
4686                                ;
4687                                ;    BEGIN
4688                                ;       Write to TSSR to soft initialize
4689                                ;
4690                                ;       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 //////////
                                                                                              T3.3:
026666  104402                                                                                    TRAP    C$BSUB

                                        ;Write to TSSR to soft initialize
                                            JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
                                            BCS     20$                     ;BR IF SOFT INIT = OK
                                            NEXT.ERRNO
                                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                            ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
026700  104455                                                                                    TRAP    C$ERDF
026702  000463                                                                                    .WORD   307
026704  003654                                                                                    .WORD   SFIERR
026706  012074                                                                                    .WORD   SFIMSG

                                        ;Get a valid test address
                                        20$:    CLR     FATFLG              ;CLEAR FATAL ERROR FLAG
                                            CLR     T12KT                   ;TEST ABOVE 28K SWITCH
                                            MOV     #T12BLK,R3              ;POINT TO TEST PATTERN TABLE
                                        T123LOOP:
                                            CLR     KTENABLE                ;TURN OFF ABOVE 28K TEST FLAG
                                            MOV     (R3)+,R1                ;GET TEST PATTERN ADDRESS
                                            MOV     R1,R0                   ;GET ADDRESS ALL "18 BITS"
                                            BIC     #177774,R0              ;LEAVE ONLY A17 AND A16
                                            BIC     #3,R1                   ;GET RID OF A17 AND A16
                                            TST     T12KT                   ;TEST ABOVE 28K THIS TIME?
                                            BEQ     25$                     ;BR IF NO
                                            MOV     -2(R3),R0               ;GET TEST PATTERN AGAIN
                                            BIC     #^C<A1716>,R0           ;SAVE 18 BIT ADDRESS ONLY
                                            MOV     #1,KTENABLE             ;TURN ON ABOVE 28K TEST FLAG
                                        25$:    JSR     PC,T12CONVERT       ;CONVERT TEST PATTERN TO TEST ADDRESS
                                            BCS     30$                     ;BR IF VALID TEST ADDRESS
                                            JMP     60$                     ;GET NEXT TEST PATTERN
                                        ;Set the test packet characteristics data pointer to the test address
                                        30$:    MOV     #T12PACKET,R4       ;GET THE ADDRESS OF COMMAND PACKET
                                            JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
                                            MOV     T12LOADD,PKLOW(R4)      ;STORE CHAR. DATA PTR LOW ADDRESS
                                            MOV     T12HIADD,PKHI(R4)       ;STORE CHAR. DATA PTR HIGH ADDRESS
                                            JSR     PC,T12CHAR              ;STORE EXPECTED DATA IN DATA BLOCK
                                        ;Do a WRITE CHARACTERISTIC command
                                            JSR     PC,KTOFF                ;TURN OFF KT-11
                                            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
                                            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
                                            FORCERROR       32$
                                            BCS     40$                     ;BR IF SSR SET IN CHKTSSR
                                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

```
TEST 3: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS

                                        NEXT.ERRNO
                              32$:      ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
027066  104455                                                                TRAP    C$ERDF
027070  000464                                                                .WORD   308
027072  030162                                                                .WORD   T12WRTSSR
027074  012106                                                                .WORD   PKTSSR
                              40$:      CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
027076  104406                                                                TRAP    C$CLP1
                              60$:
027100  020327  030056                    CMP     R3,#T12TBE              ;DONE ALL TSTBLK TEST PATTERNS?
027104  103002                            BHIS    65$                     ;BR IF YES
027106  000137  026724                    JMP     T123LOOP                ;DO ANOTHER MODULO- 4 ADDRESS
027112  005737  027720              65$:  TST     T12KT                   ;DONE ABOVE 28K TESTING TOO?
027116  003012                            BGT     70$                     ;BR IF YES
027120  005737  003134                    TST     KTFLG                   ;ANY MEMORY ABOVE 28K ON SYSTEM?
027124  001407                            BEQ     70$                     ;BR IF NO
027126  012737  000001  027720            MOV     #1,T12KT                ;SET SWITCH
027134  012703  027724                    MOV     #T12BLK,R3              ;RESET TEST PATTERN TABLE
027140  000137  026724                    JMP     T123LOOP                ;DO ABOVE 28K TESTING
027144  004737  017334              70$:  JSR     PC,KTOFF                ;TURN OFF KT11
                                        ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\
                                                                              L10045:
027150  104403                                                                TRAP    C$ESUB
027152  005737  002222                    TST     FATFLG                  ;ANY FATAL ERRORS ?
027156  001402                            BEQ     75$                     ;BRANCH IF NOT
027160  004737  017242                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
027164                                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 driver 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 11/238 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
```

```
                                        ;               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
                                        ;

027164                                          BGNSUB                                  ;///////////// BEGIN SUBTEST //////////
027164                                                                                                  T3.4:
027164  104402                                                                                                  TRAP    C$BSUB


027166  005737  003146                          TST     T23A                            ;26-APR-83 REV B - CHK FOR 23A CPU
027172  001406                                  BEQ     5$                              ;26 APR-83 REV B   BR, IF NOT 23A
027174  023727  002120  007777                  CMP     L$HIME,#7777                    ;26-APR-83 REV B - CHK FOR > 256KB
027202  103402                                  BLO     5$                              ;26 APR 83 REV B   BR, IF < 256KB
027204  000137  027576                          JMP     NOEXTF                          ;26-APR 83 REV B - JMP OVER 256KB
027210                                  5$:
027210  005737  003140                          TST     NXMFLG                          ;GOT ENOUGH MEMORY?
027214  001002                                  BNE     10$                             ;IF SET STAY
027216  000137  027576                          JMP     NOEXTF                          ;LEAVE IF NOT SET

                                        ;Write to TSSR to soft initialize

027222  004737  016034                  10$:    JSR     PC,SOFINIT                      ;DO SOFT INIT OF CONTROLLER
027226  103405                                  BCS     11$                             ;BR IF SOFT INIT = OK
027230                                          NEXT.ERRNO
027230  010001                                  MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
027232                                          ERRDF   ERRNO,SFIERR,SFIMSG             ;DEVICE FATAL ERROR DURING INIT
027232  104455                                                                                                  TRAP    C$ERDF
027234  000465                                                                                                  .WORD   309
027236  003654                                                                                                  .WORD   SFIERR
027240  012074                                                                                                  .WORD   SFIMSG

                                        ;Do a WRITE CHARACTERISTIC command so to invert switch

027242                                  11$:    CKLOOP                                  ;LOOP IF SELECTED
027242  104406                                                                                                  TRAP    C$CLP1
027244  012704  027650                          MOV     #T12PACKET,R4                   ;GET THE ADDRESS OF COMMAND PACKET
027250  004737  031230                          JSR     PC,T12SWRT                      ;RESTORE PACKET TO STARTING VALUES
027254  005037  003136                          CLR     KTENABLE                        ;TURN OFF KT-11
027260  010465  000000                          MOV     R4,TSDB(R5)                     ;SET THE PACKET ADDRESS
027264  004737  016376                          JSR     PC,CHKTSSR                      ;WAIT FOR SSR TO SET
027270                                          FORCERROR       15$
027304  103405                                  BCS     17$                             ;BR IF SSR SET IN CHKTSSR
027306  010001                                  MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
027310                                          NEXT.ERRNO
027310                                  15$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR          ;DEVICE FATAL SSR FAILED TO SET
027310  104455                                                                                                  TRAP    C$ERDF
027312  000466                                                                                                  .WORD   310
027314  030162                                                                                                  .WORD   T12WRTSSR
027316  012106                                                                                                  .WORD   PKTSSR
027320                                  17$:    CKLOOP                                  ;LOOP IF SELECTED
027320  104406                                                                                                  TRAP    C$CLP1
```

```
TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

4837 027322  004737  020504                    JSR     PC,INVERT               ;INVERT THE SWITCH
4838
4839                                 ;Get an invalid test address
4840
4841 027326  005037  002222          20$:      CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
4842 027332                          25$:
4843 027332  013737  003144  027712            MOV     NXMHI,T12HIADD          ;SAVE TEST ADDRESS HIGH
4844 027340  013737  003142  027714            MOV     NXMLO,T12LOADD          ;SAVE TEST ADDRESS LOW
4845 027346                          T124LOOP:
4846
4847                                 ;Set the test packet characteristics data pointer to the
4848                                 ;  test address.
4849
4850 027346  012704  027650          30$:      MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
4851 027352  004737  031230                    JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
4852 027356  013764  027714  000002            MOV     T12LOADD,PKLOW(R4)      ;STORE CHAR. DATA PTR LOW ADDRESS
4853 027364  013764  027712  000004            MOV     T12HIADD,PKHI(R4)       ;STORE CHAR. DATA PTR HIGH ADDRESS
4854
4855                                 ;Do a WRITE CHARACTERISTIC command
4856 027372  004737  017334                    JSR     PC,KTOFF                ;TURN OFF KT 11
4857 027376  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4858 027402  004737  016310                    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
4859 027406                                    FORCERROR       32$
4860 027422  103407                            BCS     40$                     ;BR IF SSR SET IN CHKTSSR
4861 027424  010001                            MOV     R0,R1                   ;SAVE CONTENTS Of TSSR
4862 027426                                    NEXT.ERRNO
4863 027426                          32$:      ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
     027426  104455                                                                    TRAP    C$ERDF
     027430  000467                                                                    .WORD   311
     027432  030162                                                                    .WORD   T12WRTSSR
     027434  012106                                                                    .WORD   PKTSSR
4864 027436  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4865 027442                          40$:      CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     027442  104406                                                                    TRAP    C$CLP1
4866 027444                                    FORCERROR       45$,NOTSSR
4867 027454                                    ESCAPE  SUB                     ;BY PASS SUBTEST IF FATAL ERROR
     027454  104410                                                                    TRAP    C$ESCAPE
     027456  000124                                                                    .WORD   L10046-.
4868                                 ;If TSSR register NXM bit not set then print error message
4869 027460                          45$:
4870 027460  016501  000002                    MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
4871 027464                                    FORCERROR       52$
4872 027500  032701  004000                    BIT     #NXM,R1                 ;NXM SET?
4873 027504  001012                            BNE     60$                     ;BR IF YES
4874 027506                                    NEXT.ERRNO
4875 027506  013737  027714  002242  52$:      MOV     T12LOADD,ERRLO          ;MEMORY TEST ADDRESS LOW
4876 027514  013737  027712  002240            MOV     T12HIADD,ERRHI          ;MEMORY TEST ADDRESS HIGH
4877 027522                                    ERRHRD  ERRNO,T12NXM,ADDSSR     ;REPORT ADDRESS AND TSSR ERROR
     027522  104456                                                                    TRAP    C$ERHRD
     027524  000470                                                                    .WORD   312
     027526  030617                                                                    .WORD   T12NXM
     027530  012166                                                                    .WORD   ADDSSR
4878
4879 027532                          60$:      CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     027532  104406                                                                    TRAP    C$CLP1
4880 027534                                    FORCFXIT        90$
4881 027544  005737  003146                    TST     T23A                    ;IS IT A 11/23A?
```

F10

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
4882 027550  001012                    BNE     90$                 ;YES WERE DONE
4883 027552  013700  027712            MOV     T12HIADD,R0         ;GET CURRENT HIGH ADDRESS
4884 027556  005200              65$:  INC     R0                  ;GET NEXT ADDRESS
4885 027560  020027  000077            CMP     R0,#77              ;DONE A21 A16?
4886 027564  101004                    BHI     90$                 ;BR IF YES
4887 027566  010037  027712      75$:  MOV     R0,T12HIADD         ;SETUP NEW HIGH ORDER ADDRESS
4888 027572  000137  027346            JMP     T124LOOP            ;DO ANOTHER NON EXISTENT ADDRESS
4889 027576              90$:
4890 027576              NOEXTF:
4891 027576  004737  017334            JSR     PC,KTOFF            ;TURN OFF KT11
4892 027602                            ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\
     027602                                                                L10046:
     027602  104403                                                          TRAP    C$ESUB
4893 027604  005737  002222            TST     FATFLG              ;ANY FATAL ERRORS ?
4894 027610  001402                    BEQ     100$                ;BRANCH IF NOT
4895 027612  004737  017242            JSR     PC,CKDROP           ;TRY TO DROP THE UNIT
4896 027616  004737  016516      100$: JSR     PC,TSTLOOP          ;SHOULD WE DO ITERATIONS?
4897 027622  103002                    BCC     105$                ;BR IF NO
4898 027624  000137  026036            JMP     T12LOOP             ;LOOP UNTIL ITERATION COUNT DONE
4899 027630              105$:
4900 027630  004737  017334            JSR     PC,KTOFF            ;TURN OFF MEMORY MANAGEMENT
4901 027634  005037  003152            CLR     T3BFLG              ;CLEAR TEST FLAG
4902 027640                            EXIT    TST                 ;ALL DONE THIS TEST
     027640  104432                                                          TRAP    C$EXIT
     027642  001540                                                          .WORD   L10042 .
4903
4904
4905                                   ;*
4906                                   ;LOCAL STORAGE FOR THIS TEST
4907                                   ;
4908
4910         027650                            .=<.+10>&177770
4912 027650              T12PACKET:                                ;COMMAND PACKET FOR TEST
4913 027650  100004                    .WORD   100004              ;WRITE CHARACTERISTICS COMMAND, WITH ACK
4914 027652  027660                    .WORD   T12DATA             ;ADDRESS OF CHARACTERISTICS BLOCK
4915 027654  000000                    .WORD   0
4916 027656  000010                    .WORD   8.                  ;STARTING VALUE OF BLOCK SIZE
4917
4918 027660              T12DATA:                                  ;CHARACTERISTICS DATA BLOCK
4919 027660  027672                    .WORD   T12BFR              ;LOW ADDRESS OF MESSAGE BUFFER
4920 027662  000000                    .WORD   0                   ;HIGH ORDER OF MESSAGE BUFFER
4921 027664  000016                    .WORD   14.                 ;LENGTH OF MESSAGE BUFFER
4922 027666  000000  000000            .WORD   0,0
4923
4924 027672              T12BFR: .BLKW   8.                        ;MESSAGE BUFFER
4925
4926 027712  000000              T12HIADD:       .WORD   0         ;HIGH ADDRESS
4927 027714  000000              T12LOADD:       .WORD   0         ;LOW ADDRESS
4928 027716  000000              T12PAR6:        .WORD   0         ;ADDRESS IN PAR FORMAT
4929 027720  000000              T12KT:          .WORD   0         ;TEST ABOVE 28K SWITCH
4930 027722  000000              T124TST:        .WORD   0         ;ADDRESS TEST BIT
4931                                   ;*
4932                                   ;
4933                                   ;TABLE OF ADDRESSES
4934                                   ;
4935                                   ;-
4936 027724  000001              T12BLK: .WORD   000001
```

G10

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
 4937 027726  000002                          .WORD   000002
 4938 027730  000003                          .WORD   000003
 4939 027732  000005                          .WORD   000005
 4940 027734  000006                          .WORD   000006
 4941 027736  000007                          .WORD   000007
 4942 027740  000011                          .WORD   000011
 4943 027742  000012                          .WORD   000012
 4944 027744  000013                          .WORD   000013
 4945 027746  000021                          .WORD   000021
 4946 027750  000022                          .WORD   000022
 4947 027752  000023                          .WORD   000023
 4948 027754  000041                          .WORD   000041
 4949 027756  000042                          .WORD   000042
 4950 027760  000043                          .WORD   000043
 4951 027762  000101                          .WORD   000101
 4952 027764  000102                          .WORD   000102
 4953 027766  000103                          .WORD   000103
 4954 027770  000201                          .WORD   000201
 4955 027772  000202                          .WORD   000202
 4956 027774  000203                          .WORD   000203
 4957 027776  000401                          .WORD   000401
 4958 030000  000402                          .WORD   000402
 4959 030002  000403                          .WORD   000403
 4960 030004  001001                          .WORD   001001
 4961 030006  001002                          .WORD   001002
 4962 030010  001003                          .WORD   001003
 4963 030012  002001                          .WORD   002001
 4964 030014  002002                          .WORD   002002
 4965 030016  002003                          .WORD   002003
 4966 030020  004001                          .WORD   004001
 4967 030022  004002                          .WORD   004002
 4968 030024  004003                          .WORD   004003
 4969 030026  010001                          .WORD   010001
 4970 030030  010002                          .WORD   010002
 4971 030032  010003                          .WORD   010003
 4972 030034  020001                          .WORD   020001
 4973 030036  020002                          .WORD   020002
 4974 030040  020003                          .WORD   020003
 4975 030042  040001                          .WORD   040001
 4976 030044  040002                          .WORD   040002
 4977 030046  040003                          .WORD   040003
 4978 030050  100001                          .WORD   100001
 4979 030052  100002                          .WORD   100002
 4980 030054  100003                          .WORD   100003
 4981 030056  177777                T12TBE:   .WORD           177777
 4982                                ;+
 4983                                ;LOCAL TEXT MESSAGES FOR TEST
 4984                                ;-
 4985
 4986 030060      104     115     101  TST12ID:          .ASCIZ  'DMA Memory Addressing'
 4987 030106      103     157     156  T12GET  ?:        .ASCIZ  'Contents of TSSR Incorrect After GET STATUS'
 4988 030162      103     157     156  T12WRTSSR:        .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
 4989 030251      115     145     163  T12MSGBUF:        .ASCIZ  'Message Buffer Contents Incorrect After WRITE CHARACTERISTICS'
 4990 030347      102     141     143  T12BKGND:         .ASCIZ  'Background Pattern Disturbed By WRITE CHARACTERISTICS'
 4991 030435      105     170     160  T12NINT:          .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
 4992 030526      127     162     151  T12DPR:           .ASCIZ  'Write Characteristic data in ram does not match expected'
 4993 030617      124     123     123  T12[illegible]1:  .ASCIZ  'TSSR NXM bit failed to set when non-existent memory address specified'
```

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
                                ;-
030726                          T12CONVERT:
030726                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
030732  005037  027714                  CLR     T12LOADD                ;CLEAR LOW ADDRESS
030736  005037  027712                  CLR     T12HIADD                ;CLEAR HIGH ADDRESS
030742  005037  027716                  CLR     T12PAR6                 ;CLEAR PAR6 BIASED ADDRESS
030746  042701  170000                  BIC     #^C<7777>,R1            ;FORCE TO LOWER 12 BITS OF ADDRESS
030752  010005                          MOV     R0,R5                   ;SAVE HIGH ORDER ADDRESS BITS
030754  004737  017334                  JSR     PC,KTOFF                ;SHUTOFF MEMORY MANAGEMENT
030760  013702  003126                  MOV     FREE,R2                 ;GET FIRST FREE ADDRESS
030764  062702  000020                  ADD     #16.,R2                 ;IN CASE TEST PATTERN=0
030770  060102                          ADD     R1,R2                   ;ADD IN TEST PATTERN
030772  042702  000003                  BIC     #3,R2                   ;MAKE IT MODULO-4
030776  013703  003132          25$:    MOV     FREEHI,R3               ;GET LAST FREE ADDRESS
031002  162703  000020                  SUB     #16.,R3                 ;SAVE AT LEAST 8 WORDS (IN CASE MESSAGE BUFFER)
031006  010237  027714                  MOV     R2,T12LOADD             ;SAVE POSSIBLE LOW ADDRESS
031012  010237  027716                  MOV     R2,T12PAR6              ;SAVE IT IN PAR6 BIASED TOO
031016  020203                          CMP     R2,R3                   ;IS THIS ADDRESS ABOVE FREE SPACE?
031020  101007                          BHI     35$                     ;BR IF YES
031022  020237  003126                  CMP     R2,FREE                 ;IS IT IN FREE SPACE?
031026  103007                          BHIS    50$                     ;BR IF YES- ITS GOOD
031030  005737  003136                  TST     KTENABLE                ;TESTING ABOVE 28K?
031034  001004                          BNE     50$                     ;BR IF YES
031036  000424                          BR      90$                     ;BR IF NOT IN FREE SPACE
031040  162702  000020          35$:    SUB     #16.,R2                 ;FORCE FIT THE TEST PATTERN
031044  000754                          BR      25$                     ;TRY THIS TEST PATTERN ADDRESS
031046                          50$:
031046  005737  003136                  TST     KTENABLE                ;TESTING ABOVE 28K?
031052  001420                          BEQ     100$                    ;BR IF NO
031054  005737  003134                  TST     KTFLG                   ;ANY MEMORY ABOVE 28K?
031060  001413                          BEQ     90$                     ;BR IF NO
031062  004737  017316                  JSR     PC,KTON                 ;TURN ON MEMORY MANAGEMENT
031066  010500                          MOV     R5,R0                   ;GET HIGH ORDER ADDRESS
031070  010037  027712                  MOV     R0,T12HIADD             ;SAVE POSSIBLE HIGH ADDRESS
031074  010201                          MOV     R2,R1                   ;GET COMPUTED LOW ORDER ADDRESS
031076  004737  017356                  JSR     PC,SETMAP               ;RETURN PAR6 BIASED ADDRESS IN R0
```

I10

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
5051 031102  010037  027716                    MOV     R0,T12PAR6      ;COPY PAR6 BIASED ADDRESS
5052 031106  103403                            BCS     105$            ;BR IF VALID ADDRESS
5053 031110  000241                  90$:      CLC                     ;CLR C BIT FOR FAILURE
5054 031112  000401                            BR      105$            ;
5055 031114  000261                  100$:     SEC                     ;SET SUCCESS
5056 031116  000207                  105$:     RTS     PC              ;RETURN
5057
5058
5059                                 ;*
5060                                 ;
5061                                 ;ROUTINE TO READ THE FIRST 2 BYTES FROM RAM
5062                                 ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
5063                                 ;
5064                                 ;INPUT:
5065                                 ;
5066                                 ;       R4      ADDRESS OF THE COMMAND PACKET
5067                                 ;       R5      FIRST DEVICE UNIBUS ADDRESS
5068                                 ;
5069                                 ;OUTPUT:
5070                                 ;
5071                                 ;       CARRY   SET   RAM MATCHES PACKET
5072                                 ;               CLR   RAM DOES NOT MATCH PACKET
5073                                 ;
5074                                 ;IMPLICIT OUTPUT:
5075                                 ;
5076                                 ;       THE TABLE RAMDATA IS FILLED WITH THE
5077                                 ;       DATA HELD IN RAM.
5078                                 ;       RAMSIZ  SET TO 2 FOR PRAMPKT ROUTINE
5079                                 ;
5080                                 ;SIDE EFFECTS:
5081                                 ;
5082                                 ;       THE SUBSYS #4 IS LEFT IN MAINTENANCE MODE
5083                                 ;
5084                                 ;
5085 031120                         T12CKRAM::
5086 031120                                 SAVREG                          ;SAVE THE GENERAL REGISTERS
5087 031124  012701  002244                MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
5088 031130  012702  000201                MOV     #RMPKTBEG,R2            ;BYTE ADDRESS OF FIRST RAM DATA
5089 031134  005003                        CLR     R3                      ;CLEAR THE ERROR FLAG
5090 031136  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR
5091 031142  112765  000000  000000        MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
5092 031150  004737  016376          10$:  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5093 031154  010265  000000                MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
5094 031160  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5095 031164  116511  000000                MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
5096 031170  122124                        CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
5097 031172  001401                        BEQ     20$                     ;BRANCH IF OK
5098 031174  005203                        INC     R3                      ;SET ERROR FLAG
5099 031176  005202                  20$:  INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
5100 031200  020227  000203                CMP     R2,#RMPKTBEG+2          ;DONE 2 BYTES?
5101 031204  002761                        BLT     10$                     ;BR IF NO
5102 031206  005703                        TST     R3                      ;WAS AN ERROR FOUND ?
5103 031210  001402                        BEQ     30$                     ;BRANCH IF NOT
5104 031212  000241                        CLC                             ;CLEAR CARRY TO SHOW ERROR
5105 031214  000401                        BR      50$                     ;AND EXIT
5106 031216  000261                  30$:  SEC                             ;SHOW GOOD COMPARE
5107 031220  012737  000002  002304  50$:  MOV     #2,RAMSIZ               ;SETUP RAMSIZ
```

```
TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

 5108 031226  000207                          RTS     PC                      ;RETURN
 5109
 5110                                ;*
 5111                                ;
 5112                                ;ROUTINE TO SETUP PACKET TO WRITE CHARACTERISTICS
 5113                                ;-
 5114
 5115 031230                         T12SWRT:
 5116 031230                                 SAVREG                          ;SAVE THE REGISTERS
 5117 031234  012701  027650                MOV     #T12PACKET,R1           ;START OF THE PACKET
 5118 031240  012721  100004                MOV     #100004,(R1)+           ;WRITE CHARACTERISTICS WITH ACK
 5119 031244  012721  027660                MOV     #T12DATA,(R1)+          ;ADDRESS OF CHAR DATA BLOCK
 5120 031250  005021                        CLR     (R1)+                   ;EXTENDED ADDRESS
 5121 031252  012721  000010                MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
 5122 031256  012721  027672                MOV     #T12BFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
 5123 031262  005021                        CLR     (R1)+
 5124 031264  012721  000016                MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
 5125 031270  005021                        CLR     (R1)+
 5126 031272  005011                        CLR     (R1)
 5127 031274  000207                        RTS     PC                      ;RETURN
 5128                                ;*
 5129                                ;
 5130                                ;ROUTINE TO SETUP A GET STATUS COMMAND PACKET AT CURRENT PACKET ADDRESS
 5131                                ;
 5132                                ;       R3      HIGH ORDER PACKET ADDRESS
 5133                                ;       R4      LOW ORDER PACKET ADDRESS
 5134                                ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG CLEAR
 5135                                ;
 5136                                ;
 5137
 5138 031276                         T12SETGET:
 5139 031276                                 SAVREG                          ;SAVE THE REGISTERS
 5140 031302  010401                        MOV     R4,R1                   ;GET LOW ORDER ADDRESS
 5141 031304  005737  003136                TST     KTENABLE                ;TESTING ABOVE 28K?
 5142 031310  001404                        BEQ     10$                     ;BR IF NO
 5143 031312  010300                        MOV     R3,R0                   ;GET HIGH ORDER ADDRESS
 5144 031314  004737  017356                JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
 5145 031320  010001                        MOV     R0,R1                   ;GET ADDRESS
 5146 031322  012700  000017         10$:   MOV     #P.GETSTATUS,R0         ;GET STATUS COMMAND CODE NO IE
 5147 031326  052700  100000                BIS     #P.ACK,R0               ;SET ACK
 5148 031332  010021                        MOV     R0,(R1)+                ;STORE GET STATUS IN PACKET
 5149 031334  005021                        CLR     (R1)+                   ;CLEAR UNUSED WORD
 5150 031336  000207                        RTS     PC                      ;RETURN
 5151
 5152
 5153                                ;*
 5154                                ;
 5155                                ;ROUTINE TO SETUP A CHARACTERISTIC DATA BLOCK AT A TEST ADDRESS
 5156                                ;
 5157                                ;-
 5158 031340                         T12CHAR:
 5159 031340                                 SAVREG                          ;SAVE R1 R5 UNTIL NEXT RETURN
 5160 031344  012700  027660                MOV     #T12DATA,R0             ;GET T12PACKET DATA POINTER
 5161 031350  013701  027714                MOV     T12LOADD,R1             ;ASSUME NOT ABOVE 28K
 5162 031354  005737  003136                TST     KTENABLE                ;TESTING ABOVE 28K?
 5163 031360  001402                        BEQ     10$                     ;BR IF NO
 5164 031362  013701  027716                MOV     T12PAR6,R1              ;SET TEST ADDRESS ABOVE 28K
```

K10

SEQ 0127

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
 5165 031366  012021                10$:    MOV     (R0)+,(R1)+     ;STORE DATA WORD 1
 5166 031370  012021                        MOV     (R0)+,(R1)+     ;STORE DATA WORD 2
 5167 031372  012021                        MOV     (R0)+,(R1)+     ;STORE DATA WORD 3
 5168 031374  012021                        MOV     (R0)+,(R1)+     ;STORE DATA WORD 4
 5169 031376  012021                        MOV     (R0)+,(R1)+     ;STORE DATA WORD 5
 5170 031400  000207                        RTS     PC              ;RETURN
 5171
 5172 031402                                ENDTST
      031402                                                     L10042:
      031402  104401                                                    TRAP    C$ETST
```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
                                        .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
                                ;
                                ;-
031404                                  BGNTST
031404                                                                  T4::
031404  005737  002214                  TST     TSTCNT                  ;CHECK FOR RUN MODE
031410  001402                          BEQ     10$                     ;BR, IF NOT ONLY PROGRAM RUN
031412  005237  003402                  INC     SKIPT                   ;SET SKIP SW
031416  012700  034043          10$:    MOV     #TST15ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
031422  004737  016550                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
031426  012737  000005  002216          MOV     #5,LOOPCNT              ;PERFORM 5 ITERATIONS
031434                          T15LOOP:
                                ;
                                ;
                                ;TEST 4, SUBTEST 1
                                ;
                                ;       THIS SUBTEST WRITES THE ADDRESS (8 BITS) INTO THE
                                ;       RAM MEMORY SINGLE WORD (8 BITS) MODE
                                ;
                                ;
031434                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
031434                                                                          T4.1:
031434  104402                                                                  TRAP    C$BSUB
031436                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
031436  012700  000000                                                          MOV     #PRI00,R0
031442  104441                                                                  TRAP    C$SPRI
031444  005737  003402                  TST     SKIPT                   ;SHOULD WE SKIP THIS SUBTEST
031450  001402                          BEQ     10$                     ;BR, IF NOW SKIP REQUIRED
031452  000137  031734                  JMP     50$                     ;SKIP SUBTEST
031456  004737  034062          10$:    JSR     PC,T15REST              ;SET COMMAND PACKET
031462  004737  034134                  JSR     PC,T15RT2               ;SET UP OTHER COMMAND PACKET
031466  004737  016034                  JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
031472  103405                          BCS     20$                     ;BR IF INIT WAS OK
031474  010001                          MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
031476                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
031476  104455                                                                  TRAP    C$ERDF
031500  000621                                                                  .WORD   401
031502  003654                                                                  .WORD   SFIERR
031504  012074                                                                  .WORD   SFIMSG
031506                          20$:
031506  012704  032760                  MOV     #T15PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
031512  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
031516  103405                          BCS     23$                     ;BR, IF COMMAND ISSUED OK
031520  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
031522                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
031522  104456                                                                  TRAP    C$ERHRD
031524  000622                                                                  .WORD   402
031526  005060                                                                  .WORD   WRTMSG
031530  012074                                                                  .WORD   SFIMSG
```

M10

TEST 4: RAM EXERCISER TEST

```
5228 031532  012703  000400          23$:    MOV     #256.,R3                ;STARTING ADDRESS FOR RAM WRITE
5229 031536  112737  000001  033471          MOVB    #1,T15BS1               ;SIZE OF TRANSFER
5230 031544  112737  000002  033470          MOVB    #2,T15BS0               ;WRITE RAM "COMMAND"
5231 031552                             25$:
5232 031552  010337  033472                  MOV     R3,T15S2                ;ADDRESS FOR RAM
5233 031556  012704  033460                  MOV     #T15PK2,R4              ;WRITE SUBSYS MEM PACKET
5234 031562  110337  033474                  MOVB    R3,T15S3                ;DATA FOR WRITE (ADDRESS)
5235 031566  010465  000000                  MOV     R4,TSDB(R5)             ;ISSUE COMMAND
5236 031572  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
5237 031576  103407                         BCS     30$                     ;BR, IF NO ERROR
5238 031600  010001                         MOV     R0,R1                   ;ERROR, SAVE TSSR
5242 031602                                  ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT AFTER WRITE SUB MEM
     031602  104456                                                                 TRAP    C$ERHRD
     031604  000623                                                                 .WORD   403
     031606  033476                                                                 .WORD   T15SSR
     031610  012106                                                                 .WORD   PKTSSR
5243 031612                                  ESCAPE  SUB                     ;DON'T CONTINUE IF ERROR ON WRITE
     031612  104410                                                                 TRAP    C$ESCAPE
     031614  000122                                                                 .WORD   L10050-.
5244 031616                             30$:    CKLOOP                          ;SCOPE LOOP
     031616  104406                                                                 TRAP    C$CLP1
5245
5246
5247 031620  005203                         INC     R3                      ;NEXT ADDRESS
5248 031622  020327  010000                  CMP     R3,#10000               ;END OF RAM MEMORY CHECK
5249 031626  001351                         BNE     25$                     ;LOOP TILL ALL RAM WRITTEN
5250 031630  005002                         CLR     R2                      ;CLEAR OUT R2 HIGH BITS
5251 031632  005303                         DEC     R3                      ;SET BACK TO 7777
5252 031634  110337  033474          40$:    MOVB    R3,T15S3                ;GET DATA PATTERN BACK IN SHAPE
5253 031640  010337  033472                  MOV     R3,T15S2                ;ADDRESS FOR RAM READ
5254 031644  112737  000001  033470          MOVB    #1,T15BS0               ;READ RAM COMMAND
5255 031652  010465  000000                  MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS TO CONTR.
5256 031656  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
5257 031662  103405                         BCS     43$                     ;BR, IF NO PROBLEM
5258 031664  010001                         MOV     R0,R1                   ;SAVE TSSR
5262 031666                                  ERRDF   ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     031666  104455                                                                 TRAP    C$ERDF
     031670  000624                                                                 .WORD   404
     031672  033476                                                                 .WORD   T15SSR
     031674  012106                                                                 .WORD   PKTSSR
5263 031676                             43$:    CKLOOP                          ;SCOPE LOOP
     031676  104406                                                                 TRAP    C$CLP1
5264 031700  013701  033022                  MOV     T15BFR+20,R1            ;GET RAM READ DATA
5265 031704  010302                         MOV     R3,R2                   ;SET UP FOR COMPARE
5266 031706  120102                         CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
5267 031710  001404                         BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
5271 031712                                  ERRHRD  ERRNO,T15AM4,EXPBREC    ;WRITTEN DATA NOT = TO READ
     031712  104456                                                                 TRAP    C$ERHRD
     031714  000625                                                                 .WORD   405
     031716  033755                                                                 .WORD   T15AM4
     031720  015542                                                                 .WORD   EXPBREC
5272 031722                             45$:    CKLOOP                          ;SCOPE LOOP
     031722  104406                                                                 TRAP    C$CLP1
5273 031724  005303                         DEC     R3                      ;DROP DATA COUNTER (PATTERN)
5274 031726  020327  000377                  CMP     R3,#255.                ;AT BOTTOM YET
5275 031732  001340                         BNE     40$                     ;BR, IF MORE TO CHECK
5276 031734                             50$:    CKLOOP                          ;SCOPE LOOP
```

```
TEST  4: RAM EXERCISER TEST

      031734  104406                                                          TRAP    C$CLP1
 5277 031736                          ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      031736                                                                L10050:
      031736  104403                                                          TRAP    C$ESUB
 5278
 5279
 5280 031740                          BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
      031740                                                                T4.2:
      031740  104402                                                          TRAP    C$BSUB
 5281
 5282                           ;*
 5283                           ;
 5284                           ;TEST 4, SUBTEST 2
 5285                           ;
 5286                           ;
 5287                           ;       THIS SUBTEST WRITES RAM WITH ALL ZEROS
 5288                           ;       THEN WALKS AN ALL ONES WORD DOWN THROUGH MEMORY
 5289                           ;
 5290 031742  004737  034062            JSR     PC,T15REST              ;RESTORE PACKET FOR WRITE CHARA
 5291 031746  004737  034134            JSR     PC,T15RT2               ;RESTORE PACKET FOR WRT SUB SYS MEM
 5292 031752  004737  016034            JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
 5293 031756  103405                    BCS     20$                     ;BR IF INIT WAS OK
 5297 031760  010001                    MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
 5298 031762                            ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
      031762  104455                                                          TRAP    C$ERDF
      031764  000626                                                          .WORD   406
      031766  003654                                                          .WORD   SFIERR
      031770  012074                                                          .WORD   SFIMSG
 5299 031772                    20$:
 5300 031772  012704  032760            MOV     #T15PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
 5301 031776  004737  010662            JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 5302 032002  103405                    BCS     25$                     ;BR, IF COMMAND ISSUED OK
 5306 032004  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5307 032006                            ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      032006  104456                                                          TRAP    C$ERHRD
      032010  000627                                                          .WORD   407
      032012  005060                                                          .WORD   WRTMSG
      032014  012074                                                          .WORD   SFIMSG
 5308 032016                    25$:
 5309 032016  112737  000001  033471        MOVB    #1,T15BS1               ;SET SIZE OF TRANSFER ! BYTE
 5310 032024  012704  033460            MOV     #T15PK2,R4              ;SET NEW PACKET ADDRESS
 5311 032030  012703  000400            MOV     #256.,R3                ;STARTING ADDRESS IN RAM
 5312 032034  112737  000002  033470        MOVB    #2,T15BS0               ;WRITE RAM COMMAND
 5313 032042  105037  033474            CLRB    T15S3                   ;SET DATA TO 000
 5314 032046  010337  033472    30$:    MOV     R3,T15S2                ;ADDRESS TO PACKET DATA AREA
 5315 032052  010465  000000            MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
 5316 032056  004737  016376            JSR     PC,CHKTSSR              ;WAIT FOR SSR
 5317 032062  103405                    BCS     33$                     ;BR, IF NO PROBLEM
 5318 032064  010001                    MOV     R0,R1                   ;SAVE TSSR
 5322 032066                            ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
      032066  104456                                                          TRAP    C$ERHRD
      032070  000630                                                          .WORD   408
      032072  033476                                                          .WORD   T15SSR
      032074  012106                                                          .WORD   PKTSSR
 5323 032076                    33$:    CKLOOP                          ;SCOPE LOOP
      032076  104406                                                          TRAP    C$CLP1
 5324
```

TEST 4: RAM EXERCISER TEST

```
5325
5326 032100  005203                        INC     R3                      ;NEXT ADDRESS
5327 032102  020327  010000                CMP     R3,#10000               ;END OF RAM MEMORY CHECK
5328 032106  001357                        BNE     30$                     ;BR, MORE RAM TO GO
5329 032110  005303                35$:    DEC     R3                      ;SET BACK TO 7777
5330 032112  005002                40$:    CLR     R2                      ;SET TO ALL ZEROS
5331 032114  112737  000001  033470        MOVB    #1,T15BS0               ;READ RAM COMMAND
5332 032122  010337  033472                MOV     R3,T15S2                ;ADDRESS TO BE READ TO PACKET DATA
5333 032126  010465  000000                MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
5334 032132  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5335 032136  103405                        BCS     41$                     ;BR, IF ALL IS WELL
5336 032140  010001                        MOV     R0,R1                   ;SAVE TSSR
5340 032142                                ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032142  104456                                                        TRAP    C$ERHRD
     032144  000631                                                        .WORD   409
     032146  033476                                                        .WORD   T15SSR
     032150  012106                                                        .WORD   PKTSSR
5341 032152                        41$:    CKLOOP                          ;SCOPE LOOP
     032152  104406                                                        TRAP    C$CLP1
5342 032154  013701  033022                MOV     T15BFR+20,R1            ;PICK UP READ DATA
5343 032160  120102                        CMPB    R1,R2                   ;BOTH SHOULD BE 00000000 BINARY
5344 032162  001404                        BEQ     42$                     ;BR, IF DATA IS GOOD
5348 032164                                ERRHRD  ERRNO,T15AM3,EXPBREC    ;CHARACTERISTICS DATA NOT CORRECT
     032164  104456                                                        TRAP    C$ERHRD
     032166  000632                                                        .WORD   410
     032170  033653                                                        .WORD   T15AM3
     032172  015542                                                        .WORD   EXPBREC
5349 032174                        42$:    CKLOOP                          ;SCOPE LOOPER
     032174  104406                                                        TRAP    C$CLP1
5350 032176  012702  000377                MOV     #000377,R2              ;SET ALL ONES WORD
5351 032202  112737  000002  033470        MOVB    #2,T15BS0               ;WRITE RAM COMMAND
5352 032210  112737  000377  033474        MOVB    #000377,T15S3           ;ALL ONES PATTERN
5353 032216  010465  000000                MOV     R4,TSDB(R5)             ;PASS PACKET ADDRESS TO CONTR.
5354 032222  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR
5355 032226  103405                        BCS     43$                     ;BR, IF OK (NO ERROR)
5356 032230  010001                        MOV     R0,R1                   ;SAVE TSSR
5360 032232                                ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032232  104456                                                        TRAP    C$ERHRD
     032234  000633                                                        .WORD   411
     032236  033476                                                        .WORD   T15SSR
     032240  012106                                                        .WORD   PKTSSR
5361 032242                        43$:    CKLOOP                          ;SCOPE LOOP
     032242  104406                                                        TRAP    C$CLP1
5362 032244  112737  000001  033470        MOVB    #1,T15BS0               ;SET UP FOR RAM READ
5363 032252  010465  000000                MOV     R4,TSDB(R5)             ;ISSUE RAM READ
5364 032256  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5365 032262  103405                        BCS     44$                     ;BR, IF OK (NO ERROR)
5366 032264  010001                        MOV     R0,R1                   ;SAVE TSSR
5370 032266                                ERRDF   ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032266  104455                                                        TRAP    C$ERDF
     032270  000634                                                        .WORD   412
     032272  033476                                                        .WORD   T15SSR
     032274  012106                                                        .WORD   PKTSSR
5371 032276  013701  033022        44$:    MOV     T15BFR+20,R1            ;PICK UP REC'D DATA
5372 032302  120102                        CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
5373 032304  001404                        BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
5377 032306                                ERRHRD  ERRNO,T15AM2,EXPBREC    ;WRITTEN DATA NOT = TO READ
```

```
TEST  4: RAM EXERCISER TEST

        032306  104456                                                          TRAP    C$ERHRD
        032310  000635                                                          .WORD   413
        032312  033552                                                          .WORD   T15AM2
        032314  015542                                                          .WORD   EXP8REC
 5378   032316                          45$:    CKLOOP                  ;SCOPE LOOP
        032316  104406                                                          TRAP    C$CLP1
 5379   032320  005303                  DEC     R3                      ;DROP RAM ADDRESS POINTER
 5380   032322  020327  000377         CMP     R3,#255.                ;AT START YET
 5381   032326  001271                  BNE     40$                     ;BR, IF MORE RAM TO CHECK
 5382
 5383   032330                          ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
        032330                                                                  L10051:
        032330  104403                                                          TRAP    C$ESUB
 5384
 5385   032332                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
        032332                                                                  T4.3:
        032332  104402                                                          TRAP    C$BSUB
 5386                                   ;*
 5387                                   ;
 5388                                   ;TEST 4, SUBTEST 3
 5389                                   ;
 5390                                   ;
 5391                                   ;       THIS SUBTEST WRITES RAM WITH ALL ONES
 5392                                   ;       THEN WALKS AN ALL ZEROS WORD DOWN THROUGH MEMORY
 5393                                   ;
 5394   032334  005737  003402          TST     SKIPT                   ;CHECK RUN MODE
 5395   032340  001402                  BEQ     10$                     ;BR, IF NO SKIP
 5396   032342  000137  032736          JMP     50$                     ;SKIP SUBTEST
 5397   032346  004737  034062  10$:    JSR     PC,T15REST              ;RESTORE PACKET FOR WRITE CHARA
 5398   032352  004737  034134          JSR     PC,T15RT2               ;RESTORE PACKET FOR WRT SUB SYS MEM
 5399   032356  004737  016034          JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
 5400   032362  103405                  BCS     20$                     ;BR IF INIT WAS OK
 5404   032364  010001                  MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
 5405   032366                          ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
        032366  104455                                                          TRAP    C$ERDF
        032370  000636                                                          .WORD   414
        032372  003654                                                          .WORD   SFIERR
        032374  012074                                                          .WORD   SFIMSG
 5406   032376                  20$:
 5407   032376  012704  032760          MOV     #T15PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
 5408   032402  004737  010662          JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 5409   032406  103405                  BCS     25$                     ;BR, IF COMMAND ISSUED OK
 5413   032410  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5414   032412                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
        032412  104456                                                          TRAP    C$ERHRD
        032414  000637                                                          .WORD   415
        032416  005060                                                          .WORD   WRTMSG
        032420  012074                                                          .WORD   SFIMSG
 5415   032422                  25$:
 5416   032422  112737  000001  033471  MOVB    #1,T15BS1               ;SET SIZE TO 1 BYTE
 5417   032430  012704  033460          MOV     #T15PK2,R4              ;SET NEW PACKET ADDRESS
 5418   032434  012703  000400          MOV     #256.,R3                ;STARTING ADDRESS IN RAM
 5419   032440  112737  000002  033470  MOVB    #2,T15BS0               ;WRITE RAM COMMAND
 5420   032446  112737  000377  033474  MOVB    #377,T15S3              ;SET DATA TO 377
 5421   032454  010337  033472  30$:    MOV     R3,T15S2                ;ADDRESS TO PACKET DATA AREA
 5422   032460  010465  000000          MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
 5423   032464  004737  016376          JSR     PC,CHKTSSR              ;WAIT FOR SSR
```

SEQ 0133

TEST 4: RAM EXERCISER TEST

```
5424 032470  103405                    BCS     33$                     ;BR, IF NO PROBLEM
5425 032472  010001                    MOV     R0,R1                   ;SAVE TSSR
5429 032474                            ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032474  104456                                                    TRAP    C$ERHRD
     032476  000640                                                    .WORD   416
     032500  033476                                                    .WORD   T15SSR
     032502  012106                                                    .WORD   PKTSSR
5430 032504                    33$:    CKLOOP                          ;SCOPE LOOP
     032504  104406                                                    TRAP    C$CLP1
5431
5432
5433 032506  005203                    INC     R3                      ;NEXT ADDRESS
5434 032510  020327  010000            CMP     R3,#010000              ;END OF RAM MEMORY CHECK
5435 032514  001357                    BNE     30$                     ;BR, MORE RAM TO GO
5436 032516  005303            35$:    DEC     R3                      ;SET BACK TO 7777
5437 032520  112702  000377    40$:    MOVB    #377,R2                 ;SET TO ALL ONES
5438 032524  112737  000001  033470    MOVB    #1,T15BS0               ;READ RAM COMMAND
5439 032532  010337  033472            MOV     R3,T15S2                ;ADDRESS TO BE READ TO PACKET DATA
5440 032536  010465  000000            MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
5441 032542  004737  016376            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5442 032546  103405                    BCS     41$                     ;BR, IF ALL IS WELL
5443 032550  010001                    MOV     R0,R1                   ;SAVE TSSR
5447 032552                            ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032552  104456                                                    TRAP    C$ERHRD
     032554  000641                                                    .WORD   417
     032556  033476                                                    .WORD   T15SSR
     032560  012106                                                    .WORD   PKTSSR
5448 032562                    41$:    CKLOOP                          ;SCOPE LOOP
     032562  104406                                                    TRAP    C$CLP1
5449 032564  013701  033022            MOV     T15BFR+20,R1            ;PICK UP READ DATA
5450 032570  120102                    CMPB    R1,R2                   ;BOTH SHOULD BE 11111111 BINARY
5451 032572  001404                    BEQ     42$                     ;BR, IF DATA IS GOOD
5455 032574                            ERRHRD  ERRNO,T15AM3,EXPBREC    ;CHARACTERISTICS DATA NOT CORRECT
     032574  104456                                                    TRAP    C$ERHRD
     032576  000642                                                    .WORD   418
     032600  033653                                                    .WORD   T15AM3
     032602  015542                                                    .WORD   EXPBREC
5456 032604  012702  000377    42$:    MOV     #000377,R2              ;SET ALL ONES WORD
5457 032610  012737  000002  033470    MOV     #2,T15BS0               ;WRITE RAM COMMAND
5458 032616  112737  000377  033474    MOVB    #000377,T15S3           ;ALL ONES PATTERN
5459 032624  010465  000000            MOV     R4,TSDB(R5)             ;PASS PACKET ADDRESS TO CONTR.
5460 032630  004737  016376            JSR     PC,CHKTSSR              ;WAIT FOR SSR
5461 032634  103405                    BCS     43$                     ;BR, IF OK (NO ERROR)
5462 032636  010001                    MOV     R0,R1                   ;SAVE TSSR
5466 032640                            ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032640  104456                                                    TRAP    C$ERHRD
     032642  000643                                                    .WORD   419
     032644  033476                                                    .WORD   T15SSR
     032646  012106                                                    .WORD   PKTSSR
5467 032650                    43$:    CKLOOP                          ;SCOPE LOOP
     032650  104406                                                    TRAP    C$CLP1
5468 032652  112737  000001  033470    MOVB    #1,T15BS0               ;SET UP FOR RAM READ
5469 032660  010465  000000            MOV     R4,TSDB(R5)             ;ISSUE RAM READ
5470 032664  004737  016376            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5471 032670  103405                    BCS     44$                     ;BR, IF OK (NO ERROR)
5472 032672  010001                    MOV     R0,R1                   ;SAVE TSSR
5476 032674                            ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
```

```
TEST  4: RAM EXERCISER TEST

        032674  104456                                                          TRAP    C$ERHRD
        032676  000644                                                          .WORD   420
        032700  033476                                                          .WORD   T15SSR
        032702  012106                                                          .WORD   PKTSSR
  5477  032704  013701  033022          44$:    MOV     T15BFR+20,R1          ;PICK UP REC'D DATA
  5478  032710  120102                  CMPB    R1,R2                 ;CHECK WITH DATA WRITTEN
  5479  032712  001404                  BEQ     45$                   ;BR IF OK, DATA IN = DATA OUT
  5483  032714                          ERRHRD  ERRNO,T15AM2,EXPBREC  ;WRITTEN DATA NOT = TO READ
        032714  104456                                                          TRAP    C$ERHRD
        032716  000645                                                          .WORD   421
        032720  033552                                                          .WORD   T15AM2
        032722  015542                                                          .WORD   EXPBREC
  5484  032724                  45$:    CKLOOP                        ;SCOPE LOOP
        032724  104406                                                          TRAP    C$CLP1
  5485  032726  005303                  DEC     R3                    ;DROP RAM ADDRESS POINTER
  5486  032730  020327  000377          CMP     R3,#255.              ;AT START YET
  5487  032734  001271                  BNE     40$                   ;BR, IF MORE RAM TO CHECK
  5488
  5489  032736                  50$:
  5490  032736                          ENDSUB                        ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        C32736                                                                  L10052:
        032736  104403                                                          TRAP    C$ESUB
  5491
  5492  032740  004737  016516          JSR     PC,TSTLOOP            ;DO WE NEED TO ITERATE TEST ?
  5493  032744  103002                  BCC     63$                   ;BRANCH IF NOT
  5494  032746  000137  031434          JMP     T15LOOP               ;EXECUTE AGAIN
  5495  032752                  63$:    EXIT    TST                   ;ALL DONE THIS TEST
        032752  104432                                                          TRAP    C$EXIT
        032754  001216                                                          .WORD   L10047-.
  5496                          ;+
  5497                          ;LOCAL STORAGE FOR THIS TEST
  5498                          ;-
  5499
  5501          032760                  .=<.+10>&177770
  5503  032760                  T15PACKET:                            ;COMMAND PACKET FOR TEST
  5504  032760  100204                  .WORD   100204                ;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
  5505  032762  032770                  .WORD   T15DATA               ;ADDRESS OF CHARACTERISTICS BLOCK
  5506  032764  000000                  .WORD   0
  5507  032766  000010                  .WORD   8.                    ;STARTING VALUE OF BLOCK SIZE
  5508  032770                  T15DATA:                              ;CHARACTERISTICS DATA BLOCK
  5509  032770  033002                  .WORD   T15BFR                ;ADDRESS OF MESSAGE BUFFER
  5510  032772  000000                  .WORD   0
  5511  032774  000400                  .WORD   256.                  ;LENGTH OF MESSAGE BUFFER
  5512  032776  000000  000000          .WORD   0,0
  5513  033002                  T15BFR: .BLKW   150.                  ;MESSAGE BUFFER
  5514                          ;
  5515                          ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
  5516                          ;
  5518          033460                  .=<.+10>&177770
  5520  033460                  T15PK2:
  5521  033460  100206                  .WORD   100206                ;WRITE SUB SYS MEM COMMAND, IE AND ACK
  5522  033462  033470                  .WORD   T15BF2                ;ADDRESS OF SELECT BLOCK DATA
  5523  033464  000000                  .WORD   0
  5524  033466  000006                  .WORD   6.                    ;SIZE OF DATA PACKET
  5525
  5526                                  .EVEN
  5527  033470                  T15BF2:
```

F11

TEST 4: RAM EXERCISER TEST

```
5528 033470     000                    T15BS0: .BYTE   0                       ;BSEL0 AREA
5529 033471     000                    T15BS1: .BYTE   0                       ;BSEL1 AREA
5530 033472  000000                    T15S2:  .WORD   0                       ;SEL 2 AREA
5531 033474  000000                    T15S3:  .WORD   0                       ;DATA AREA
5532
5533
5534
5535
5536                                   ;+
5537                                   ;LOCAL TEXT MESSAGES FOR TEST
5538                                   ;
5539
5540 033476     127     122     111   T15SSR: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
5541 033552     127     122     111   T15AM2: .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Ones Word Read Back'
5542 033653     127     122     111   T15AM3: .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Zeros Word Read Back'
5543 033755     127     122     111   T15AM4: .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On Address Test'
5544 034043     122     101     115   TST15ID:        .ASCIZ  'RAM Exerciser'
5545                                           .EVEN
5546                                   ;+
5547                                   ;
5548                                   ;ROUTINE TO RESTORE COMMAND PACKET TO START UP (DEFAULT) VALUES
5549                                   ;WRITE SUBSYSTEM MEMORY COMMAND
5550                                   ;
5551                                   ;-
5552
5553 034062                            T15REST:
5554 034062                                    SAVREG                          ;SAVE THE REGISTERS
5555 034066  012701  032760                     MOV     #T15PACKET,R1           ;START OF THE PACKET
5556 034072  012721  100204                     MOV     #100204,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK, IE
5557 034076  012721  032770                     MOV     #T15DATA,(R1)+          ;ADDRESS OF CHARAISTICS DATA BLOCK
5558 034102  005021                             CLR     (R1)+                   ;EXTENDED ADDRESS
5559 034104  012721  000010                     MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
5560 034110  012721  033002                     MOV     #T15BFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
5561 034114  005021                             CLR     (R1)+
5562 034116  012721  000400                     MOV     #256.,(R1)+             ;LENGTH OF MESSAGE BUFFER
5563 034122  005021                             CLR     (R1)+
5564 034124  005011                             CLR     (R1)
5565 034126  005037  033002                     CLR     T15BFR                  ;CLEAR 1ST LOC IN MESSAGE BUFFER
5566 034132  000207                             RTS     PC                      ;RETURN
5567
5568
5569 034134                            T15RT2:
5570 034134                                    SAVREG                          ;SAVE THE REGISTERS
5571 034140  012701  033460                     MOV     #T15PK2,R1              ;START OF THE PACKET
5572 034144  012721  100206                     MOV     #100206,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK, IE
5573 034150  012721  033470                     MOV     #T15BF2,(R1)+           ;ADDRESS OF DATA BLOCK
5574 034154  005021                             CLR     (R1)+                   ;EXTENDED ADDRESS
5575 034156  012721  000006                     MOV     #6.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
5576 034162  005021                             CLR     (R1)+
5577 034164  005021                             CLR     (R1)+
5578 034166  005011                             CLR     (R1)
5579 034170  000207                             RTS     PC                      ;RETURN
5580 034172                                     ENDTST
     034172
     034172  104401                                                     L10047:
                                                                                TRAP    C$ETST
```

```
                                        .SBTTL  TEST  5: EXTENDED FEATURES SWITCH AND TIMERS A,B
                                ;**
                                ; TEST DESCRIPTION:
                                ;
                                ;       This test verifies the Invert Extended Features funct on
                                ;       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
                                ;


                                        BGNTST
034174                                                                          T5::
034174  012700  036252                  MOV     @TST16ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
034200  004737  016550                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
034204  012737  000012  002216          MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
034212                          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
                                        ;--
034212                                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
034212                                                                                  T5.1:
034212  104402                                                                                  TRAP    C$BSUB

034214                                  5$:
                                        ;       Write to TSSR register to soft initialize the controller
034214  004737  016034                          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
034220  103405                                  BCS     10$                     ;BR IF SOFT INIT OKAY
034222  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
034224                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
034224  104455                                                                          TRAP    C$ERDF
034226  000764                                                                          .WORD   500
034230  003654                                                                          .WORD   SFIERR
034232  012074                                                                          .WORD   SFIMSG
                                        ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
034234  004737  037420          10$:            JSR     PC,T16REST              ;RESTORE PACKET DEFAULTS
034240  005037  002222                          CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
034244  012704  037600                          MOV     @T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
034250  004737  010662                          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
034254                                          FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
034270  103407                                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
034272  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
034274                                          NEXT.ERRNO
034274                          12$:            ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
034274  104455                                                                          TRAP    C$ERDF
034276  000765                                                                          .WORD   501
034300  036322                                                                          .WORD   T16SSR
034302  012106                                                                          .WORD   PKTSSR
034304  005237  002222                          INC     FATFLG                  ;SET FATAL ERROR FLAG
034310                          15$:            CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
034310  104406                                                                          TRAP    C$CLP1

                                        ;       If Extended Features Hardware Switch Clear then:
                                        ;         (* Verify Extended Features switch can be Inverted to SET *)
                                        ;         REPEAT FOR TEST PATTERNS IN TSTBLK TABLE
034312  012701  037622                          MOV     @T16BFR,R1              ;MESSAGE BUFFER ADDRESS
034316  032761  000200  000012                  BIT     @X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH CLEAR?
034324  001402                                  BEQ     20$                     ;BR IF YES
034326  000137  034676                          JMP     200$                    ;
034332  012703  002766          20$:            MOV     @TSTBLK+10.,R3          ;START OF TEST DATA
```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```
5688                                    ;       Do Write Subsystem Write Miscellaneous to SET Extended Features.
5689
5690 034336  004737  037560                      JSR     PC,T16SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
5691 034342  012704  037650                      MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5692 034346  010465  000000                      MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5693 034352  004737  016376                      JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5694 034356                                      FORCERROR       32$                     ;@@FORCE ERROR IF FORCER=1
5695 034372  103407                              BCS     40$                     ;BR IF CARRY SET (GOOD RETURN)
5696 034374  010001                              MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5697 034376                                      NEXT.ERRNO
5698 034376                              32$:    ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     034376  104455                                                                      TRAP    C$ERDF
     034400  000766                                                                      .WORD   502
     034402  036357                                                                      .WORD   T162SSR
     034404  012106                                                                      .WORD   PKTSSR
5699 034406  005237  002222                      INC     FATFLG                  ;SET FATAL ERROR FLAG
5700 034412                              40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034412  104406                                                                      TRAP    C$CLP1
5701
5702                                    ;         DO a WRITE CHARACTERISTICS with an extended characteristic word
5703 034414  012737  125252  002314              MOV     #125252,DATA            ;SETUP TEST DATA FOR EXTENDED WORD
5704 034422  012704  037600                      MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
5705 034426  012764  000020  000006              MOV     #16.,PKBCNT(R4)         ;STORE MESSAGE PACKET SIZE
5706 034434  013737  002314  037620              MOV     DATA,T16DATA+10         ;STORE TEST DATA IN EXTENDED WORD
5707 034442  004737  010662                      JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
5708 034446                                      FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
5709 034462  103407                              BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
5710 034464  010001                              MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5711 034466                                      NEXT.ERRNO
5712 034466                              42$:    ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     034466  104455                                                                      TRAP    C$ERDF
     034470  000767                                                                      .WORD   503
     034472  036322                                                                      .WORD   T16SSR
     034474  012106                                                                      .WORD   PKTSSR
5713 034476  005237  002222                      INC     FATFLG                  ;SET FATAL ERROR FLAG
5714 034502                              50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034502  104406                                                                      TRAP    C$CLP1
5715                                    ;         If the TSBA Address Register NOT= Expected  Then Print Error
5716 034504  016501  000000                      MOV     TSBA(R5),R1             ;GET TSBA REGI TER CONTENTS
5717 034510  012702  037622                      MOV     #T16BFR,R2              ;START OF THE DATA BUFFER
5718 034514  062702  000020              62$:    ADD     #16.,R2                 ;EXPECTED CONTENTS OF TSBA
5719 034520                                      FORCERROR       72$,NOTSSR              ;@@FORCE ERROR IF FORCER=1
5720 034530  020102                              CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
5721 034532  001404                              BEQ     80$                     ;ERROR IF NOT EQUAL
5722 034534                                      NEXT.ERRNO
5723 034534                              72$:    ERRHRD  ERRNO,T16TSBA,EXPREC    ;PRINT THE ERROR & EXPD/RECV
     034534  104456                                                                      TRAP    C$ERHRD
     034536  000770                                                                      .WORD   504
     034540  036470                                                                      .WORD   T16TSBA
     034542  015534                                                                      .WORD   EXPREC
5724 034544                              80$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034544  104406                                                                      TRAP    C$CLP1
5725                                    ;         Compare the controller ram to the extended characteristic word
5726                                    ;         If Data word in controller ram NOT= to word sent  Then Print Error
5727 034546  012704  037610                      MOV     #T16DATA,R4             ;GET CHARACTERISTIC DATA ADDRESS
5728 034552  004737  011264                      JSR     PC,CKRAM2               ;DOES RAM DATA EQUAL DATA SENT?
5729 034556                                      FORCERROR       92$                     ;@@FORCE ERROR IF FORCER=1
```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```
5730 034572  103404                        BCS     100$                    ;BR IF YES
5731 034574                                 NEXT.ERRNO
5732 034574                          92$:   ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
     034574  104456                                                         TRAP    C$ERHRD
     034576  000771                                                         .WORD   505
     034600  004747                                                         .WORD   PKTRAM
     034602  015550                                                         .WORD   RAMERR
5733 034604                          100$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034604  104406                                                         TRAP    C$CLP1
5734                                 ;        If Message Buffer Data Length NOT= 12.  Then Print Error
5735 034606  012702  037622                  MOV     @T16BFR,R2             ;GET MESSAGE BUFFER ADDRESS
5736 034612  016201  000002                  MOV     2(R2),R1               ;GET RECV DATA FIELD LENGTH
5737 034616  012702  000014                  MOV     #12.,R2                ;GET EXPD DATA FIELD LENGTH
5738 034622                                  FORCERROR       112$,NOTSSR            ;@@FORCE ERROR IF FORCER=1
5739 034632  020102                        CMP     R1,R2                  ;COMPARE EXPECTED TO RECEIVED
5740 034634  001404                        BEQ     120$                   ;ERROR IF NOT EQUAL
5741 034636                                 NEXT.ERRNO
5742 034636                          112$:  ERRHRD  ERRNO,T16LEN,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     034636  104456                                                         TRAP    C$ERHRD
     034640  000772                                                         .WORD   506
     034642  036572                                                         .WORD   T16LEN
     034644  015534                                                         .WORD   EXPREC
5743 034646                          120$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034646  104406                                                         TRAP    C$CLP1
5744
5745 034650  004737  016034                  JSR     PC,SOFINIT             ;WRITE TO TSSR TO SOFT INITIALIZE
5746 034654  103405                        BCS     125$                   ;BR IF SOFT INIT OKAY
5747 034656  010001                        MOV     R0,R1                  ;SAVE CONTENTS OF TSSR
5748 034660                                 ERRDF   ERRNO,SFIERR,SFIMSG    ;DEVICE FATAL DURING INIT
     034660  104455                                                         TRAP    C$ERDF
     034662  000772                                                         .WORD   506
     034664  003654                                                         .WORD   SFIERR
     034666  012074                                                         .WORD   SFIMSG
5749 034670                          125$:  CKLOOP                          ;LOOP IF SELECTED
     034670  104406                                                         TRAP    C$CLP1
5750 034672  000137  035056                  JMP     300$                   ;
5751
5752                                 ;    (* Verify Extended Features switch can be Inverted to CLEAR *)
5753 034676                          200$:
5754                                 ;        Do Write Subsystem Write Miscellaneous to CLEAR Extended Features.
5755 034676  004737  037560                  JSR     PC,T16SEXT             ;SETUP PACKET FOR WRITE MISC INVERT
5756 034702  012704  037650                  MOV     @T16PK2,R4             ;GET WRITE SUBSYSTEM COMMAND PACKET
5757 034706  010465  000000                  MOV     R4,TSDB(R5)            ;SET THE PACKET ADDRESS TO EXECUTE
5758 034712  004737  016376                  JSR     PC,CHKTSSR             ;WAIT FOR SSR TO SET
5759 034716                                  FORCERROR       232$                    ;@@FORCE ERROR IF FORCER=1
5760 034732  103407                        BCS     240$                   ;BR IF CARRY SET (GOOD RETURN)
5761 034734  010001                        MOV     R0,R1                  ;SAVE CONTENTS OF TSSR
5762 034736                                 NEXT.ERRNO
5763 034736                          232$:  ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     034736  104455                                                         TRAP    C$ERDF
     034740  000773                                                         .WORD   507
     034742  036357                                                         .WORD   T162SSR
     034744  012106                                                         .WORD   PKTSSR
5764 034746  005237  002222                  INC     FATFLG                 ;SET FATAL ERROR FLAG
5765 034752                          240$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034752  104406                                                         TRAP    C$CLP1
5766
```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```
5767                                    ;       DO a WRITE CHARACTERISTICS without an extended characteristic word
5768 034754  012704  037600             MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
5769 034760  012764  000016  000006     MOV     #14.,PKBCNT(R4)         ;STORE MESSAGE PACKET SIZE
5770 034766  004737  010662             JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
5771 034772                             FORCERROR       242$                    ;@@DFORCE ERROR IF FORCER=1
5772 035006  103407                     BCS     250$                    ;BR IF CARRY SET (GOOD RETURN)
5773 035010  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5774 035012                             NEXT.ERRNO
5775 035012                     242$:   ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     035012  104455                                                                     TRAP    C$ERDF
     035014  000774                                                                     .WORD   508
     035016  036322                                                                     .WORD   T16SSR
     035020  012106                                                                     .WORD   PKTSSR
5776 035022  005237  002222             INC     FATFLG                  ;SET FATAL ERROR FLAG
5777 035026                     250$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035026  104406                                                                     TRAP    C$CLP1
5778                                    ;       If Message Buffer Data Length NOT= 10.  Then Print Error
5779 035030  013701  037624             MOV     T16BFR+2,R1             ;GET RECV DATA FIELD LENGTH
5780 035034  012702  000012             MOV     #10.,R2                 ;GET EXPD DATA FIELD LENGTH
5781 035040  020102                     CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
5782 035042  001404                     BEQ     270$                    ;ERROR IF NOT EQUAL
5783 035044                             NEXT.ERRNO
5784 035044                     262$:   ERRHRD  ERRNO,T16LEN,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     035044  104456                                                                     TRAP    C$ERHRD
     035046  000775                                                                     .WORD   509
     035050  036572                                                                     .WORD   T16LEN
     035052  015534                                                                     .WORD   EXPREC
5785 035054                     270$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035054  104406                                                                     TRAP    C$CLP1
5785
5787
5788                                    ;       (* Verify Function Reject when Message Buffer 21 19 are non-zero *)
5789                                    ;       Write to TSSR register to soft initialize the controller
5790 035056                     300$:
5791                                    ;       REPEAT FOR MESSAGE BUFFER ADDRESS bits <21:19> FROM 0 TO 7
5792 035056  012737  000001  002314     320$:   MOV     #1,DATA                 ;START AT BITS<21:19>=001
5793                                    ;       DO a WRITE CHARACTERISTICS with a message address bit<21:19> non zero
5794 035064                     325$:
5795 035064  012704  037600             MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
5796 035070  012764  000016  000006     MOV     #14.,PKBCNT(R4)         ;STORE MESSAGE PACKET SIZE
5797 035076  013700  002314             MOV     DATA,R0                 ;GET TEST DATA
5798         000003                     .REPT   3
5799                                    ASL     R0                      ;SHIFT INTO BITS 21:19
5800                                    .ENDR
5801 035110  010037  037612             MOV     R0,T16DATA+2            ;STORE BUFFER ADDRESS BITS 21:19
5802 035114  010465  000000             MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5803 035120  004737  016310             JSR     PC,WAITF                ;WAIT FOR SSR
5804 035124                             FORCERROR       342$                    ;@@DFORCE ERROR IF FORCER=1
5805 035140  103407                     BCS     350$                    ;BR IF CARRY SET (GOOD RETURN)
5806 035142  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5807 035144                             NEXT.ERRNO
5808 035144                     342$:   ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     035144  104455                                                                     TRAP    C$ERDF
     035146  000776                                                                     .WORD   510
     035150  036322                                                                     .WORD   T16SSR
     035152  012106                                                                     .WORD   PKTSSR
5809 035154  005237  002222             INC     FATFLG                  ;SET FATAL ERROR FLAG
```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```
 5810 035160                          350$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      035160  104406                                                                TRAP    C$CLP1
 5811
 5812                                 ;      If TSSR termination code NOT= Function Reject  Then Print Error
 5813 035162  016501  000002                 MOV     TSSR(R5),R1             ;GET RECV TSSR
 5814 035166  010102                         MOV     R1,R2                   ;COPY RECV TSSR
 5815 035170  042702  000016                 BIC     #TERCLS,R2              ;CLEAR TC<2:0> EXPD
 5816 035174  052702  000006                 BIS     #TSREJ,R2               ;SET EXPD TC<2:0>= FUNCTION REJECT
 5817 035200                                 FORCERROR       352$,NOTSSR             ;@@FORCE ERROR IF FORCER=1
 5818 035210  020102                         CMP     R1,R2                   ;EXPD EQUAL RECV?
 5819 035212  001404                         BEQ     360$                    ;BR IF YES
 5820 035214                                 NEXT.ERRNO
 5821 035214                          352$:  ERRHRD  ERRNO,T16REJ,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      035214  104456                                                                TRAP    C$ERHRD
      035216  000777                                                                .WORD   511
      035220  036704                                                                .WORD   T16REJ
      035222  012106                                                                .WORD   PKTSSR
 5822 035224                          360$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      035224  104406                                                                TRAP    C$CLP1
 5823 035226                                 FORCEXIT        370$
 5824 035236  005237  002314                 INC     DATA                    ;GET NEXT TST PATTERN
 5825 035242  023727  002314  000007         CMP     DATA,#7                 ;DONE ALL DATA?
 5826 035250  101002                         BHI     370$                    ;BR IF YES
 5827 035252  000137  035064                 JMP     325$                    ;DO ANOTHER TEST PATTERN
 5828                                 ;      END-REPEAT
 5829 035256                          370$:
 5830 035256                                 ENDSUB                          ;////////////// END SUBTEST //////////
      035256                                                                    L10054:
      035256  104403                                                                TRAP    C$ESUB
 5831
 5832 035260  005737  002222                 TST     FATFLG                  ;ANY FATAL ERRORS ?
 5833 035264  001402                         BEQ     460$                    ;BRANCH IF NOT
 5834 035266  004737  017242                 JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 5835 035272                          460$:
 5836
 5837
 5838
 5839
 5840
 5841                                        .SBTTL  TEST  5: SUBTEST 2: VERIFY TIMERS A,B
 5842
 5843                                 ;**
 5844                                 ; TEST 5: SUBTEST 2:
 5845                                 ;
 5846                                 ; SUBTEST DESCRIPTION:
 5847                                 ;
 5848                                 ;      This subtest verifies that timers A,B can be reset
 5849                                 ;      and that Timer A is twice the frequency of Timer B.
 5850                                 ;      Timer A has a period of 25 microseconds and Timer B
 5851                                 ;      has a period of 50 microseconds. The timers are
 5852                                 ;      checked at 1, 28, 53, and 78 micoseconds.
 5853                                 ;
 5854                                 ; TEST STEPS:
 5855                                 ;
 5856                                 ;
 5857                                 ;      Write to TSSR register to soft initialize the controller
 5858                                 ;      Do WRITE CHARACTERISTICS to setup a Message Buffer
```

TEST 5: SUBTFST 2: VERIFY TIMERS A,B

```
5859                                    ;       (* Verify Timers A,B after RESET TIMER with 0 microsecond delay *)
5860                                    ;       Do a Write Control RESET TIMER with 1 microsecond delay
5861                                    ;       Do a Write Subsystem READ STATUS
5862                                    ;       If Timer A NOT= 0  Then Print Error
5863                                    ;       If Timer B NOT= 0  Then Print Error
5864                                    ;       (* Verify Timers A,B after RESET TIMER with 28 microsecond delay *)
5865                                    ;       Do a Write Control RESET TIMER with 28 microsecond delay
5866                                    ;       If Timer A NOT= 1  Then Print Error
5867                                    ;       If Timer B NOT= 1  Then Print Error
5868                                    ;       Do a Write Control RESET TIMER with 53 microsecond delay
5869                                    ;       If Timer A NOT= 0  Then Print Error
5870                                    ;       If Timer B NOT= 1  Then Print Error
5871                                    ;       Do a Write Control RESET TIMER with 78 microsecond delay
5872                                    ;       If Timer A NOT= 1  Then Print Error
5873                                    ;       If Timer B NOT= 0  Then Print Error
5874                                    ;
5875 035272                                    BGNSUB                          ;////////////// BEGIN SUBTEST //////////
      035272                                                                                  T5.2:
      035272  104402                                                                            TRAP    C$BSUB
5876                                    ;       Write to TSSR register to soft initialize the controller
5877 035274                             5$:
5878 035274  004737  016034                    JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
5879 035300  103405                            BCS     10$                     ;BR IF SOFT INIT OKAY
5880 035302  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5881 035304                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      035304  104455                                                                          TRAP    C$ERDF
      035306  000777                                                                          .WORD   511
      035310  003654                                                                          .WORD   SFIERR
      035312  012074                                                                          .WORD   SFIMSG
5882                                    ;       Do WRITE CHARACTERISTICS to setup a Message Buffer
5883 035314  004737  037420            10$:    JSR     PC,T16REST              ;RESTORE PACKET DEFAULTS
5884 035320  005037  002222                    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
5885 035324  012704  037600                    MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
5886 035330  012764  000010  000006            MOV     #8.,PKBCNT(R4)          ;MESSAGE PACKET SIZE NO EXTEND
5887 035336  004737  010662                    JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
5888 035342                                    FORCERROR       12$                     ;@@FORCE ERROR IF FORCER=1
5889 035356  103407                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
5890 035360  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5891 035362                                    NEXT.ERRNO
5892 035362                             12$:   ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      035362  104455                                                                          TRAP    C$ERDF
      035364  001000                                                                          .WORD   512
      035366  036322                                                                          .WORD   T16SSR
      035370  012106                                                                          .WORD   PKTSSR
5893 035372  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
5894 035376                             15$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      035376  104406                                                                          TRAP    C$CLP1
5895
5896                                    ;       (* Verify Timers A,B after RESET TIMER with 1 microsecond delay *)
5897                                    ;       Do a Write Control RESET TIMER with 1 microsecond delay
5898 035400  012700  000001                    MOV     #MS.RSD,R0              ;RESET TIMER COMMAND
5899 035404  013701  036242                    MOV     T16D01,R1               ;1 MICROSECOND DELAY
5900 035410  004737  037532                    JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
5901 035414  012704  037650                    MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5902 035420  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5903 035424  004737  016376                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5904 035430                                    FORCERROR       32$                     ;@@FORCE ERROR IF FORCER=1
```

```
TEST  5: SUBTEST 2: VERIFY TIMERS A,B

 5905 035444  103407                      BCS     40$                     ;BR IF CARRY SET (GOOD RETURN)
 5906 035446  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5907 035450                              NEXT.ERRNO
 5908 035450                      32$:    ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      035450  104455                                                              TRAP    C$ERDF
      035452  001001                                                              .WORD   513
      035454  036357                                                              .WORD   T162SSR
      035456  012106                                                              .WORD   PKTSSR
 5909 035460  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
 5910 035464                      40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      035464  104406                                                              TRAP    C$CLP1
 5911                             ;       If Timer A NOT= 0  Then Print Error
 5912                             ;       If Timer B NOT= 0  Then Print Error
 5913 035466  005002                      CLR     R2                      ;INIT EXPD
 5914 035470  042702  000010              BIC     @S2.ATIM,R2             ;TIMER A EXPD=0
 5915 035474  042702  000004              BIC     @S2.BTIM,R2             ;TIMER B EXPD=0
 5916 035500  012700  037642              MOV     @T16BFSTA,R0            ;GET RECV READ STATUS
 5917 035504  016001  000002              MOV     2(R0),R1                ;GET RECV BYTE 2
 5918 035510  042701  177763              BIC     @+C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
 5919 035514                              FORCERROR       72$,NOTSSR      ;@@D
 5920 035524  020201                      CMP     R2,R1                   ;EXPD EQUAL RECV?
 5921 035526  001404                      BEQ     80$                     ;BR IF YES
 5922 035530                              NEXT.ERRNO
 5923 035530                      72$:    ERRHRD  ERRNO,T16T01,TIMEXP     ;REPORT ERROR
      035530  104456                                                              TRAP    C$ERHRD
      035532  001002                                                              .WORD   514
      035534  037021                                                              .WORD   T16T01
      035536  015612                                                              .WORD   TIMEXP
 5924 035540                      80$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      035540  104406                                                              TRAP    C$CLP1
 5925
 5926                             ;       Do a Write Control RESET TIMER with 28 microsecond delay
 5927 035542  012700  000001              MOV     @MS.RSD,R0              ;RESET TIMER COMMAND
 5928 035546  013701  036244              MOV     T16D28,R1               ;28 MICROSECOND DELAY
 5929 035552  004737  037532              JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
 5930 035556  012704  037650              MOV     @T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 5931 035562  010465  000000              MOV     R4,TSD8(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 5932 035566  004737  016376              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 5933 035572                              FORCERROR       112$                    ;@@DFORCE ERROR IF FORCER=1
 5934 035606  103407                      BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
 5935 035610  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5936 035612                              NEXT.ERRNO
 5937 035612                      112$:   ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      035612  104455                                                              TRAP    C$ERDF
      035614  001003                                                              .WORD   515
      035616  036357                                                              .WORD   T162SSR
      035620  012106                                                              .WORD   PKTSSR
 5938 035622  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
 5939 035626                      120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      035626  104406                                                              TRAP    C$CLP1
 5940                             ;       If Timer A NOT= 1  Then Print Error
 5941                             ;       If Timer B NOT= 1  Then Print Error
 5942 035630  005002                      CLR     R2                      ;INIT EXPD
 5943 035632  052702  000010              BIS     @S2.ATIM,R2             ;TIMER A EXPD=1
 5944 035636  052702  000004              BIS     @S2.BTIM,R2             ;TIMER B EXPD=1
 5945 035642  012700  037642              MOV     @T16BFSTA,R0            ;GET RECV READ STATUS
 5946 035646  016001  000002              MOV     2(R0),R1                ;GET RECV BYTE 2
```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```
5947 035652  042701  177763                BIC     @^C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
5948 035656                                FORCERROR       172$,NOTSSR     ;@@D
5949 035666  020201                        CMP     R2,R1                   ;EXPD EQUAL RECV?
5950 035670  001404                        BEQ     180$                    ;BR IF YES
5951 035672                                NEXT.ERRNO
5952 035672                        172$:   ERRHRD  ERRNO,T16T28,TIMEXP     ;REPORT ERROR
     035672  104456                                                                TRAP    C$ERHRD
     035674  001004                                                                .WORD   516
     035676  037120                                                                .WORD   T16T28
     035700  015612                                                                .WORD   TIMEXP
5953 035702                        180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035702  104406                                                                TRAP    C$CLP1
5954
5955                               ;       Do a Write Control RESET TIMER with 53 microsecond delay
5956 035704  012700  000001                MOV     @MS.RSD,R0              ;RESET TIMER COMMAND
5957 035710  013701  036246                MOV     T16D53,R1               ;53 MICROSECOND DELAY
5958 035714  004737  037532                JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
5959 035720  012704  037650                MOV     @T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5960 035724  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5961 035730  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5962 035734                                FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
5963 035750  103407                        BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
5964 035752  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5965 035754                                NEXT.ERRNO
5966 035754                        212$:   ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     035754  104455                                                                TRAP    C$ERDF
     035756  001005                                                                .WORD   517
     035760  036357                                                                .WORD   T162SSR
     035762  012106                                                                .WORD   PKTSSR
5967 035764  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
5968 035770                        220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035770  104406                                                                TRAP    C$CLP1
5969                               ;       If Timer A NOT= 0  Then Print Error
5970                               ;       If Timer B NOT= 1  Then Print Error
5971 035772  005002                        CLR     R2                      ;INIT EXPD
5972 035774  042702  000010                BIC     @S2.ATIM,R2             ;TIMER A EXPD=0
5973 036000  052702  000004                BIS     @S2.BTIM,R2             ;TIMER B EXPD=1
5974 036004  012700  037642                MOV     @T16BFSTA,R0            ;GET RECV READ STATUS
5975 036010  016001  000002                MOV     2(R0),R1                ;GET RECV BYTE 2
5976 036014  042701  177763                BIC     @^C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
5977 036020                                FORCERROR       272$,NOTSSR     ;@@D
5978 036030  020201                        CMP     R2,R1                   ;EXPD EQUAL RECV?
5979 036032  001404                        BEQ     280$                    ;BR IF YES
5980 036034                                NEXT.ERRNO
5981 036034                        272$:   ERRHRD  ERRNO,T16T53,TIMEXP     ;REPORT ERROR
     036034  104456                                                                TRAP    C$ERHRD
     036036  001006                                                                .WORD   518
     036040  037220                                                                .WORD   T16T53
     036042  015612                                                                .WORD   TIMEXP
5982 036044                        280$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036044  104406                                                                TRAP    C$CLP1
5983                               ;       Do a Write Control RESET TIMER with 78 microsecond delay
5984 036046  012700  000001                MOV     @MS.RSD,R0              ;RESET TIMER COMMAND
5985 036052  013701  036250                MOV     T16D78,R1               ;78 MICROSECOND DELAY
5986 036056  004737  037532                JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
5987 036062  012704  037650                MOV     @T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5988 036066  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```
 5989 036072  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 5990 036076                                FORCERROR       312$                    ;@@DFORCE ERROR IF FORCER=1
 5991 036112  103407                        BCS     320$                    ;BR IF CARRY SET (GOOD RETURN)
 5992 036114  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5993 036116                                NEXT.ERRNO
 5994 036116                        312$:   ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      036116  104455                                                                 TRAP    C$ERDF
      036120  001007                                                                 .WORD   519
      036122  036357                                                                 .WORD   T162SSR
      036124  012106                                                                 .WORD   PKTSSR
 5995 036126  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 5996 036132                        320$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      036132  104406                                                                 TRAP    C$CLP1
 5997                               ;       If Timer A NOT= 1  Then Print Error
 5998                               ;       If Timer B NOT= 0  Then Print Error
 5999 036134  005002                        CLR     R2                      ;INIT EXPD
 6000 036136  052702  000010                BIS     @S2.ATIM,R2             ;TIMER A EXPD=1
 6001 036142  042702  000004                BIC     @S2.BTIM,R2             ;TIMER B EXPD=0
 6002 036146  012700  037642                MOV     @T16BFSTA,R0            ;GET RECV READ STATUS
 6003 036152  016001  000002                MOV     2(R0),R1                ;GET RECV BYTE 2
 6004 036156  042701  177763                BIC     @^C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
 6005 036162                                FORCERROR       372$,NOTSSR     ;@@D
 6006 036172  020201                        CMP     R2,R1                   ;EXPD EQUAL RECV?
 6007 036174  001404                        BEQ     380$                    ;BR IF YES
 6008 036176                                NEXT.ERRNO
 6009 036176                        372$:   ERRHRD  ERRNO,T16T78,TIMEXP     ;REPORT ERROR
      036176  104456                                                                 TRAP    C$ERHRD
      036200  001010                                                                 .WORD   520
      036202  037320                                                                 .WORD   T16T78
      036204  015612                                                                 .WORD   TIMEXP
 6010 036206                        380$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      036206  104406                                                                 TRAP    C$CLP1
 6011
 6012 036210                                ENDSUB                          ;////////////// END SUBTEST //////////
      036210                                                                     L10055:
      036210  104403                                                                 TRAP    C$ESUB
 6013
 6014 036212  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
 6015 036216  001402                        BEQ     460$                    ;BRANCH IF NOT
 6016 036220  004737  017242                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 6017 036224  004737  016516        460$:   JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS?
 6018 036230  103002                        BCC     465$                    ;BR IF NO
 6019 036232  000137  034212                JMP     T16LOOP                 ;LOOP UNTIL ITERATIONS DONE
 6020 036236                        465$:
 6021
 6022
 6023 036236                                EXIT    TST                     ;////////// EXIT TEST //////////
      036236  104432                                                                 TRAP    C$EXIT
      036240  001524                                                                 .WORD   L10053-.
 6024
 6025
 6026                                       ;*
 6027                               ;LOCAL STORAGE FOR THIS TEST
 6028                                       ;-
 6029 036242  000001                T16D01:         .WORD   1               ;1 MICROSECOND DELAY (ACTUALLY .8 MIC)
 6030 036244  000040                T16D28:         .WORD   40              ;28 MICROSECOND DELAY (.8 MICROS PER)
 6031 036246  000076                T16D53:         .WORD   76              ;53 MICROSECOND
```

```
TEST  5: SUBTEST 2: VERIFY TIMERS A,B

6032 036250  000142                    T16D78:         .WORD   142             ;78 MICROSECOND
6033                                    ;*
6034                                    ;LOCAL TEXT MESSAGES FOR TEST
6035                                    ;
6036
6037 036252      105     170     164   TST16ID:        .ASCIZ  'Extended Features Switch and Timers A,B'
6038 036322      127     122     111   T16SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
6039 036357      127     122     111   T162SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
6040 036423      127     122     111   T163SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
6041 036470      102     165     163   T16TSBA:.ASCIZ  'Bus Address Register (TSBA) Incorrect after Write Characteristics'
6042 036572      104     141     164   T16LEN:.ASCIZ   'Data Field Length in Message Buffer Incorrect after Write Characteristics'
6043 036704      124     123     123   T16REJ:.ASCIZ   'TSSR Function Reject Not Returned When Non-Existent Buffer Address Specifie
d'
6044 037021      124     151     155   T16T01:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 1 microsecond Delay'
6045 037120      124     151     155   T16T28:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 28 microsecond Delay'
6046 037220      124     151     155   T16T53:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 53 microsecond Delay'
6047 037320      124     151     155   T16T78:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 78 microsecond Delay'
6048                                            .EVEN
6049
6050                                    ;*
6051                                    ; SET DEFAULT PACKET
6052                                    ;-
6053 037420                             T16REST:
6054 037420  012700  037600                        MOV     #T16PACKET,R0           ;PACKET ADDRESS
6055 037424  012720  100004                        MOV     #100004,(R0)+           ;WRITE CHARACTERISTICS WITH ACK
6056 037430  012720  037610                        MOV     #T16DATA,(R0)+          ;ADDRESS OF CHAR DATA BLOCK
6057 037434  005020                                CLR     (R0)+                   ;EXTENDED ADDRESS
6058 037436  012720  000012                        MOV     #10.,(R0)+              ;SIZE OF MESSAGE PACKET
6059 037442  012720  037622                        MOV     #T16BFR,(R0)+           ;MESSAGE BUFFER ADDRESS
6060 037446  005020                                CLR     (R0)+                   ;CLEAR EXTENDED BUFFER ADDRESS
6061 037450  012720  000024                        MOV     #20.,(R0)+              ;LENGTH OF MESSAGE BUFFER
6062 037454  005020                                CLR     (R0)+                   ;CLEAR ESS,ENB,EAI,ERI
6063 037456  005010                                CLR     (R0)                    ;CLEAR EXTENDED FEATURES WORD
6064 037460  005037  037622                        CLR     T16BFR                  ;CLEAR 1ST LOCATION IN MESSAGE BUFFER
6065 037464  000207                                RTS     PC                      ;
6066
6067                                    ;*
6068                                    ; CLEAR MESSAGE BUFFER
6069                                    ;
6070 037466                             T16CLRBUF:                              ;
6071 037466                                        SAVREG                      ;SAVE R1-R5 UNTIL NEXT RETURN
6072 037472  012701  037622                        MOV     #T16BFR,R1          ;GET MESSAGE BUFFER ADDRESS
6073 037476  012702  000026                        MOV     #T16BEND-T16BFR,R2  ;SIZE OF MESSAGE BUFFER IN BYTES
6074 037502  105021                     10$:       CLRB    (R1)+               ;CLEAR A BYTE
6075 037504  005302                                DEC     R2                  ;DONE?
6076 037506  003375                                BGT     10$                 ;BR IF NO
6077 037510  000207                                RTS     PC                  ;RETURN
6078
6079                                    ;*
6080                                    ; SETUP T16PK2 PACKET FOR READ STATUS
6081                                    ;-
6082 037512                             T16SRD:
6083 037512  004737  037466                        JSR     PC,T16CLRBUF        ;CLEAR MESSAGE BUFFER
6084 037516  012700  037660                        MOV     #T16DT2,R0          ;WRITE SUBSYSTEM DATA BUFFER
6085 037522  112720  000005                        MOVB    #PW.RDSTATUS,(R0)+  ;STORE READ STATUS COMMAND IN BSEL0
6086 037526  105010                                CLRB    (R0)                ;CLEAR BSEL1
6087 037530  000207                                RTS     PC                  ;RETURN
6088
```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```
                                        ;*
                                        ; SETUP T16PK2 PACKET FOR WRITE MISC.
                                        ;
                                        ; INPUT:
                                        ;       R0      CONTAINS WRITE MISC FUNCTION CODE (BSEL1)
                                        ;       R1      CONTAINS DELAY (TIMES 800 NS) FOR BSEL2
                                        ;
037532                                  T16WMISC:
037532                                          SAVREG                          ;SAVE R1-R5 UNITL NEXT RETURN
037536  004737  037466                          JSR     PC,T16CLRBUF            ;CLEAR MESSAGE BUFFER
037542  012702  037660                          MOV     #T16DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
037546  112722  000010                          MOVB    #PW.WMISC,(R2)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
037552  110022                                  MOVB    R0,(R2)+                ;STORE WRITE MISC CODE IN BSEL1
037554  110112                                  MOVB    R1,(R2)                 ;STORE DELAY (RESET TIMER) IN BSEL2
037556  000207                                  RTS     PC                      ;RETURN
                                        ;*
                                        ; SETUP T16PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
                                        ;
037560                                  T16SEXT:
037560  012700  037660                          MOV     #T16DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
037564  112720  000010                          MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
037570  112710  000200                          MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
037574  000207                                  RTS     PC                      ;RETURN


                        037600                  .=<.+10>&177770
                                        ;
                                        ;WRITE CHARACTERISTICS COMMAND PACKET
                                        ;
037600                                  T16PACKET:                              ;COMMAND PACKET FOR TEST
037600  100004                                  .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
037602  037610                                  .WORD   T16DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
037604  000000                                  .WORD   0
037606  000012                                  .WORD   10.                     ;MESSAGE PACKET SIZE

037610                                  T16DATA:                                ;CHARACTERISTICS DATA BLOCK
037610  037622                                  .WORD   T16BFR                  ;ADDRESS OF MESSAGE BUFFER
037612  000000                                  .WORD   0
037614  000024                                  .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
037616  000000                                  .WORD   0                       ;ESS,ENB,EAI,ERI
037620  000000                                  .WORD   0                       ;EXTENDED FEATURES WORD


                                        ;MESSAGE BUFFER

037622                                  T16BFR:                                 ;BEGIN MESSAGE BUFFER
037622  000000                                  .WORD   0                       ;MESSAGE TYPE
037624  000000                                  .WORD   0                       ;DATA FIELD LENGTH
037626  000000                                  .WORD   0                       ;RBPCR
037630  000000                                  .WORD   0                       ;XST0
037632  000000                                  .WORD   0                       ;XST1
037634  000000                                  .WORD   0                       ;XST2
037636  000000                                  .WORD   0                       ;XST3
037640  000000                                  .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
```

F12


```
TEST  5: SUBTEST 2: VERIFY TIMERS A,B

   6148 037642                          T16BFSTA: .BLKB 6.                      ;READ STATUS AND WRITE FIFO BUFFER
   6149 037650                          T16BEND:                                ;END OF MESSAGE BUFFER
   6150                                 ;
   6151                                 ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
   6152                                 ;
   6156 037650                          T16PK2:
   6157 037650  100006                          .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
   6158 037652  037660                          .WORD   T16DT2                  ;LOW ADDRESS OF DATA BLOCK
   6159 037654  000000                          .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
   6160 037656  000012                          .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
   6161
   6162 037660                          T16DT2:                                 ;DATA BLOCK
   6163 037660     000                          .BYTE   0                       ;BSEL0
   6164 037661     000                          .BYTE   0                       ;BSEL1
   6165 037662  000000                          .WORD   0                       ;SEL2
   6166 037664                                  .BLKB   64.                     ;WRITE FIFO DATA OUTPUT BUFFER
   6167
   6168
   6169 037764                          ENDTST
        037764                                                          L10053:
        037764  104401                                                          TRAP    C$ETST
```

TEST 6: FIFO EXERCISER

```
                                        .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
                                ;

037766                                  BGNTST
037766                                                                          ;;
037766  012700  046216                  MOV     @TST17ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
037772  004737  016550                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
037776  012737  000012  002216          MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
040004  004737  017334                  JSR     PC,KTOFF                ;SHUT OFF MEMORY MANAGEMENT
040010  005037  003136                  CLR     KTENABLE                ;REALLY SHUT DOWN KT-11
040014                          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

```
6231                                    ;
6232 040014                             BGNSUB                          ;////////////// BEGIN SUBTEST /////////
     040014                                                                     T6.1:
     040014  104402                                                             TRAP    C$BSUB
6233
6234                                    ;       Write to TSSR register to soft initialize the controller
6235 040016                             5$:
6236 040016  004737  016034                    JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
6237 040022  103405                            BCS     10$                     ;BR IF SOFT INIT OKAY
6238 040024  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6239 040026                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     040026  104455                                                             TRAP    C$ERDF
     040030  001130                                                             .WORD   600
     040032  003654                                                             .WORD   SFIERR
     040034  012074                                                             .WORD   SFIMSG
6240                                    ;       Do a WRITE CHARACTERISTICS to setup a message buffer
6241 040036  005037  002222             10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
6242 040042  012704  047610                    MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
6243 040046  004737  010662                    JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
6244 040052                                    FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
6245 040066  103407                            BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
6246 040070  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6247 040072                                    NEXT.ERRNO
6248 040072                             42$:   ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     040072  104455                                                             TRAP    C$ERDF
     040074  001131                                                             .WORD   601
     040076  046235                                                             .WORD   T17SSR
     040100  012106                                                             .WORD   PKTSSR
6249 040102  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
6250 040106                             50$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     040106  104406                                                             TRAP    C$CLP1
6251
6252                                    ;       Do a Write Subsystem READ STATUS
6253 040110  004737  047374                    JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
6254 040114  012704  047760                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6255 040120  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6256 040124  004737  016376                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6257 040130                                    FORCERROR       62$                     ;@@FORCE ERROR IF FORCER=1
6258 040144  103407                            BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
6259 040146  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6260 040150                                    NEXT.ERRNO
6261 040150                             62$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     040150  104455                                                             TRAP    C$ERDF
     040152  001132                                                             .WORD   602
     040154  046336                                                             .WORD   T173SSR
     040156  012106                                                             .WORD   PKTSSR
6262 040160  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
6263 040164                             70$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     040164  104406                                                             TRAP    C$CLP1
6264                                    ;       Set WORDS 0-7  of expd message buffer = to recv since not testing
6265 040166  004737  047556                    JSR     PC,T17SETEXP            ;SET WORDS 0 7 EXPD=RECV
6266 040172  012701  046012                    MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
6267 040176  012702  047652                    MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
6268 040202  012221                            MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
6269 040204  011211                            MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
6270 040206  052711  000020                    BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= TRUE
6271 040212  042711  000040                    BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= FALSE
```

```
TEST  6: SUBTEST 1: FIFO INITIALIZE STATUS TEST

 6272 040216  042711  000200                  BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = FALSE
 6273                                ;        If Input Ready NOT=1 then Print Error
 6274                                ;        If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
 6275 040222  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 6276 040224  012701  047632                  MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 6277 040230  012702  045772                  MOV     #T17EXP,R2              ;EXPD ADDRESS
 6278 040234  012703  000024                  MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 6279 040240  004737  011540                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 6280 040244                                  FORCERROR       82$,NOTSSR      ;ADD
 6281 040254  103404                          BCS     90$                     ;BR IF YES
 6282 040256                                  NEXT.ERRNO
 6283 040256                          82$:    ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
      040256  104456                                                                  TRAP    C$ERHRD
      040260  001133                                                                  .WORD   603
      040262  046555                                                                  .WORD   T171CMP
      040264  012410                                                                  .WORD   MSGSTAT
 6284 040266                          90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      040266  104406                                                                  TRAP    C$CLP1
 6285
 6286 040270                                  ENDSUB                          ;////////////// END SUBTEST //////////
      040270                                                                      L10057:
      040270  104403                                                                  TRAP    C$ESUB
 6287
 6288 040272  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
 6289 040276  001402                          BEQ     160$                    ;BRANCH IF NOT
 6290 040300  004737  017242                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 6291 040304                          160$:
 6292
 6293                                         .SBTTL  TEST  6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST
 6294
 6295                                ;**
 6296                                ; TEST 6: SUBTEST 2:
 6297                                ;
 6298                                ; SUBTEST DESCRIPTION:
 6299                                ;
 6300                                ;       This subtest verifies the ability of the FIFO to correctly
 6301                                ;       pass a single data byte from input to output. For each
 6302                                ;       of 256 data values (0 377 octal) the following is done:
 6303                                ;               1. Initial FIFO status is checked
 6304                                ;               2. The Write FIFO function, specifying a count of
 6305                                ;                  one byte to be written is executed.
 6306                                ;               3. Read Status is executed and FIFO status is checked.
 6307                                ;               4. Read FIFO is executed and the data and final status
 6308                                ;                  is checked.
 6309                                ;
 6310                                ; TEST STEPS:
 6311                                ;
 6312                                ; BEGIN
 6313                                ;       Write to TSSR to soft initialize
 6314                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
 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                                ;
 6320                                ;  REPEAT FOR DATA FROM 0 TO 377 OCTAL
 6321                                ;    BEGIN
```

J12

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```
                                        ;       Do a Write Subsystem WRITE NPR to set tape direction out
                                        ;       Do a Write Subsystem WRITE FIFO with byte count equal to 1
                                        ;       Do a Write Subsystem READ STATUS
                                        ;       If Input Ready  NOT=1   Then Print Error
                                        ;       If Output Ready NOT=1   Then Print Error
                                        ;       If Data In Miss NOT=0   Then Print Error
                                        ;       Do Write Subsystem READ FIFO with byte count equal to 1
                                        ;       If Data read from FIFO NOT= to Data sent Then Print Error
                                        ;       Do a Write Subsystem READ STATUS
                                        ;       If Input Ready  NOT=1   Then Print Error
                                        ;       If Output Ready NOT=0   Then Print Error
                                        ;       If Data In Miss NOT=0   Then Print Error
                                        ;   END
                                        ;  END
                                        ;
040304                                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
040304                                                                                  T6.2:
040304  104402                                                                          TRAP    C$BSUB

                                        ;       Write to TSSR register to soft initialize the controller
040306                                  5$:
040306  004737  016034                          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
040312  103405                                  BCS     10$                     ;BR IF SOFT INIT OKAY
040314  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040316                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
040316  104455                                                                          TRAP    C$ERDF
040320  001133                                                                          .WORD   603
040322  003654                                                                          .WORD   SFIERR
040324  012074                                                                          .WORD   SFIMSG

                                        ;       Do a WRITE CHARACTERISTICS to setup a message buffer
040326  005037  002222                  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
040332  012704  047610                          MOV     @T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
040336  004737  010662                          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
040342                                          FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
040356  103407                                  BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
040360  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040362                                          NEXT.ERRNO
040362                                  42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
040362  104455                                                                          TRAP    C$ERDF
040364  001134                                                                          .WORD   604
040366  046235                                                                          .WORD   T17SSR
040370  012106                                                                          .WORD   PKTSSR
040372  005237  002222                          INC     FATFLG                  ;SET FATAL ERROR FLAG
040376                                  50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
040376  104406                                                                          TRAP    C$CLP1

                                        ;       Do a Write Subsystem READ STATUS
040400  004737  047374                          JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
040404  012704  047760                          MOV     @T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
040410  010465  000000                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
040414  004737  016376                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
040420                                          FORCERROR       62$                     ;@@FORCE ERROR IF FORCER=1
040434  103407                                  BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
040436  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040440                                          NEXT.ERRNO
040440                                  62$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
040440  104455                                                                          TRAP    C$ERDF
040442  001135                                                                          .WORD   605
```

K12

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```
                040444  046336                                                                  .WORD   T173SSR
                040446  012106                                                                  .WORD   PKTSSR
6366            040450  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
6367            040454                          70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                040454  104406                                                                  TRAP    C$CLP1
6368                                            ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
6369            040456  004737  047556                  JSR     PC,T17SETEXP            ;SET WORDS 0 7 EXPD=RECV
6370            040462  012701  046012                  MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
6371            040466  012702  047652                  MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
6372            040472  012221                          MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
6373            040474  011211                          MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
6374            040476  052711  000020                  BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= TRUE
6375            040502  042711  000040                  BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= FALSE
6376            040506  042711  000200                  BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = FALSE
6377                                            ;       If Input Ready NOT=1 then Print Error
6378                                            ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
6379            040512  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
6380            040514  012701  047632                  MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
6381            040520  012702  045772                  MOV     #T17EXP,R2              ;EXPD ADDRESS
6382            040524  012703  000024                  MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
6383            040530  004737  011540                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
6384            040534                                  FORCERROR       82$,NOTSSR      ;@@D
6385            040544  103404                          BCS     90$                     ;BR IF YES
6386            040546                                  NEXT.ERRNO
6387            040546                          82$:    ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
                040546  104456                                                                  TRAP    C$ERHRD
                040550  001136                                                                  .WORD   606
                040552  046555                                                                  .WORD   T171CMP
                040554  012410                                                                  .WORD   MSGSTAT
6388            040556                          90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                040556  104406                                                                  TRAP    C$CLP1
6389
6390                                            ;  Repeat for DATA from 0 to 377
6391            040560  012737  000000  002314          MOV     #0,DATA                 ;GET FIRST DATA
6392            040566                          100$:                                   ;REPEAT LABEL
6393                                            ;       Do a Write Subsystem WRITE NPR to set tape direct on out
6394            040566  012700  000100                  MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
6395            040572  004737  047436                  JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
6396            040576  012704  047760                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6397            040602  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6398            040606  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6399            040612                                  FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
6400            040626  103407                          BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
6401            040630  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6402            040632                                  NEXT.ERRNO
6403            040632                          102$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                040632  104455                                                                  TRAP    C$ERDF
                040634  001137                                                                  .WORD   607
                040636  046403                                                                  .WORD   T174SSR
                040640  012106                                                                  .WORD   PKTSSR
6404            040642  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
6405            040646                          105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                040646  104406                                                                  TRAP    C$CLP1
6406                                            ;       Do a Write Subsystem WRITE FIFO with byte count equal to 1
6407            040650  012700  000001                  MOV     #1,R0                   ;WRITE 1 BYTE
6408            040654  012701  002314                  MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
6409            040660  004737  047462                  JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
```

L12


```
TEST  6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

 6410 040664  012704  047760                 MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 6411 040670  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 6412 040674  004737  016376                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 6413 040700                                 FORCERROR       107$                    ;@@FORCE ERROR IF FORCER=1
 6414 040714  103407                         BCS     110$                    ;BR IF CARRY SET (GOOD RETURN)
 6415 040716  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6416 040720                                 NEXT.ERRNO
 6417 040720                         107$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      040720  104455                                                                 TRAP    C$ERDF
      040722  001140                                                                 .WORD   608
      040724  046446                                                                 .WORD   T175SSR
      040726  012106                                                                 .WORD   PKTSSR
 6418 040730  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 6419 040734                         110$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      040734  104406                                                                 TRAP    C$CLP1
 6420
 6421                                ;       Do a Write Subsystem READ STATUS
 6422 040736  004737  047374                 JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
 6423 040742  012704  047760                 MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 6424 040746  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 6425 040752  004737  016376                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 6426 040756                                 FORCERROR       112$                    ;@@FORCE ERROR IF FORCER=1
 6427 040772  103407                         BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
 6428 040774  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6429 040776                                 NEXT.ERRNO
 6430 040776                         112$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      040776  104455                                                                 TRAP    C$ERDF
      041000  001141                                                                 .WORD   609
      041002  046336                                                                 .WORD   T173SSR
      041004  012106                                                                 .WORD   PKTSSR
 6431 041006  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 6432 041012                         120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      041012  104406                                                                 TRAP    C$CLP1
 6433                                ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
 6434 041014  004737  047556                 JSR     PC,T17SETEXP            ;SET WORDS 0 7 EXPD=RECV
 6435 041020  012701  046012                 MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
 6436 041024  012702  047652                 MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
 6437 041030  012221                         MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
 6438 041032  011211                         MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
 6439 041034  052711  000020                 BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
 6440 041040  052711  000040                 BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
 6441 041044  042711  000200                 BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
 6442                                ;       If Input Ready NOT=1 then Print Error
 6443                                ;       If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
 6444 041050  005000                         CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 6445 041052  012701  047632                 MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 6446 041056  012702  045772                 MOV     #T17EXP,R2              ;EXPD ADDRESS
 6447 041062  012703  000024                 MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 6448 041066  004737  011540                 JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 6449 041072                                 FORCERROR       132$,NOTSSR             ;@@D
 6450 041102  103404                         BCS     140$                    ;BR IF YES
 6451 041104                                 NEXT.ERRNO
 6452 041104                         132$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
      041104  104456                                                                 TRAP    C$ERHRD
      041106  001142                                                                 .WORD   610
      041110  046733                                                                 .WORD   T173CMP
      041112  012410                                                                 .WORD   MSGSTAT
```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```
6453 041114                          140$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041114  104406                                                                  TRAP    C$CLP1
6454
6455                                 ;       Do Write Subsystem READ FIFO with byte count equal to 1
6456 041116  012700  000001                  MOV     #1,R0                   ;SET READ BYTE COUNT
6457 041122  004737  047516                  JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
6458 041126  012704  047760                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6459 041132  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6460 041136  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6461 041142                                  FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
6462 041156  103407                          BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
6463 041160  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6464 041162                                  NEXT.ERRNO
6465 041162                          142$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041162  104455                                                                  TRAP    C$ERDF
     041164  001143                                                                  .WORD   611
     041166  046512                                                                  .WORD   T176SSR
     041170  012106                                                                  .WORD   PKTSSR
6466 041172  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
6467 041176                          150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041176  104406                                                                  TRAP    C$CLP1
6468                                 ;       Set WORDS 0 7 of expd message buffer = to recv since not testing
6469 041200  004737  047556                  JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
6470 041204  012701  046012                  MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
6471 041210  012702  047652                  MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
6472 041214  013721  002314                  MOV     DATA,(R1)+              ;SET EXPD WORD #8 = COUNT DATA
6473 041220  011211                          MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV (NOT TESTING)
6474                                 ;       If Data read from FIFO NOT= to Data sent Then Print Error
6475                                 ;       The data is in WORD #8 of the message buffer
6476 041222  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
6477 041224  012701  047632                  MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
6478 041230  012702  045772                  MOV     #T17EXP,R2              ;EXPD ADDRESS
6479 041234  012703  000022                  MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
6480 041240  004737  011540                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
6481 041244                                  FORCERROR       152$,NOTSSR             ;@@D
6482 041254  103404                          BCS     160$                    ;BR IF YES
6483 041256                                  NEXT.ERRNO
6484 041256                          152$:   ERRHRD  ERRNO,T172CMP,MSGSUB    ;REPORT ERROR
     041256  104456                                                                  TRAP    C$ERHRD
     041260  001144                                                                  .WORD   612
     041262  046637                                                                  .WORD   T172CMP
     041264  014002                                                                  .WORD   MSGSUB
6485 041266                          160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041266  104406                                                                  TRAP    C$CLP1
6486
6487                                 ;       Do a Write Subsystem READ STATUS
6488 041270  004737  047374                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
6489 041274  012704  047760                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6490 041300  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6491 041304  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6492 041310                                  FORCERROR       162$                    ;@@DFORCE ERROR IF FORCER=1
6493 041324  103407                          BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
6494 041326  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6495 041330                                  NEXT.ERRNO
6496 041330                          162$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041330  104455                                                                  TRAP    C$ERDF
     041332  001145                                                                  .WORD   613
```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```
        041334  046336                                                          .WORD   T173SSR
        041336  012106                                                          .WORD   PKTSSR
  6497  041340  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
  6498  041344                        170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        041344  104406                                                          TRAP    C$CLP1
  6499                                  ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
  6500  041346  004737  047556                JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
  6501  041352  012701  046012                MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
  6502  041356  012702  047652                MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
  6503  041362  012221                        MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
  6504  041364  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
  6505  041366  052711  000020                BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
  6506  041372  042711  000040                BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
  6507  041376  042711  000200                BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
  6508                                  ;       If Input Ready NOT=1 then Print Error
  6509                                  ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
  6510  041402  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
  6511  041404  012701  047632                MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
  6512  041410  012702  045772                MOV     #T17EXP,R2              ;EXPD ADDRESS
  6513  041414  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
  6514  041420  004737  011540                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
  6515  041424                                FORCERROR       172$,NOTSSR     ;@@D
  6516  041434  103404                        BCS     180$                    ;BR IF YES
  6517  041436                                NEXT.ERRNO
  6518  041436                        172$:   ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
        041436  104456                                                          TRAP    C$ERHRD
        041440  001146                                                          .WORD   614
        041442  047017                                                          .WORD   T174CMP
        041444  012410                                                          .WORD   MSGSTAT
  6519  041446                        180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        041446  104406                                                          TRAP    C$CLP1
  6520  041450                                FORCEXIT        205$            ;@@D
  6521  041460  005237  002314                INC     DATA                    ;GET NEXT TEST DATA
  6522  041464  023727  002314  000377        CMP     DATA,#377               ;DONE 0 TO 377?
  6523  041472  101002                        BHI     205$                    ;BR IF YES
  6524  041474  000137  040566                JMP     100$                    ;DO ANOTHER TEST PATTERN
  6525  041500                        205$:
  6526
  6527  041500                                ENDSUB                          ;////////////// END SUBTEST //////////
        041500                                                                  L10060:
        041500  104403                                                          TRAP    C$ESUB
  6528
  6529  041502  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
  6530  041506  001402                        BEQ     260$                    ;BRANCH IF NOT
  6531  041510  004737  017242                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
  6532  041514                        260$:
  6533
  6534                                  .SBTTL  TEST  6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST
  6535
  6536                                  ;**
  6537                                  ; TEST 6: SUBTEST 3:
  6538                                  ;
  6539                                  ; SUBTEST DESCRIPTION:
  6540                                  ;
  6541                                  ;       This subtest verifies the ability of the FIFO to correctly
  6542                                  ;       pass a multiple data bytes from input to output.
  6543                                  ;       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
                                ;
041514                                  BGNSUB                  ;////////////// BEGIN SUBTEST //////////
041514                                                                          T6.3:
041514  104402                                                                          TRAP    C$BSUB
                                ;
                                ;       Write to TSSR register to soft initialize the controller
041516                          5$:
041516  004737  016034                  JSR     PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
041522  103405                          BCS     10$             ;BR IF SOFT INIT OKAY
041524  010001                          MOV     R0,R1           ;SAVE CONTENTS OF TSSR
041526                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
041526  104455                                                                  TRAP    C$ERDF
041530  001146                                                                  .WORD   614
041532  003654                                                                  .WORD   SFIERR
041534  012074                                                                  .WORD   SFIMSG
                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
041536  005037  002222          10$:    CLR     FATFLG          ;CLEAR FATAL ERROR FLAG
041542  012704  047610                  MOV     #T17PACKET,R4   ;GET THE ADDRESS OF COMMAND PACKET
041546  004737  010662                  JSR     PC,WRTCHR       ;DO WRITE CHARACTERISTICS COMMAND
041552                                  FORCERROR       42$             ;@@DFORCE ERROR IF FORCER=1
041566  103407                          BCS     50$             ;BR IF CARRY SET (GOOD RETURN)
```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```
6595 041570  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6596 041572                                NEXT.ERRNO
6597 041572                          42$:  ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     041572  104455                                                              TRAP    C$ERDF
     041574  001147                                                              .WORD   615
     041576  046235                                                              .WORD   T17SSR
     041600  012106                                                              .WORD   PKTSSR
6598 041602  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
6599 041606                          50$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041606  104406                                                              TRAP    C$CLP1
6600                                 ;     Do a Write Subsystem READ STATUS
6601 041610  004737  047374                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
6602 041614  012704  047760                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6603 041620  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6604 041624  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6605 041630                                  FORCERROR       62$             ;@@FORCE ERROR IF FORCER=1
6606 041644  103407                          BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
6607 041646  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6608 041650                                  NEXT.ERRNO
6609 041650                          62$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041650  104455                                                              TRAP    C$ERDF
     041652  001150                                                              .WORD   616
     041654  046336                                                              .WORD   T173SSR
     041656  012106                                                              .WORD   PKTSSR
6610 041660  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
6611 041664                          70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041664  104406                                                              TRAP    C$CLP1
6612                                 ;     Set WORDS 0 7 of expd message buffer = to recv since not testing
6613 041666  004737  047556                  JSR     PC,T17SETEXP            ;SET WORDS 0 7 EXPD=RECV
6614 041672  012701  046012                  MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
6615 041676  012702  047652                  MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
6616 041702  012221                          MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
6617 041704  011211                          MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
6618 041706  052711  000020                  BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY=  1
6619 041712  042711  000040                  BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY=  0
6620 041716  042711  000200                  BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
6621 041722  042711  000100                  BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
6622                                 ;     If Input Ready NOT=1 then Print Error
6623                                 ;     If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
6624                                 ;     If Last Word    NOT=0   Then Print Error
6625 041726  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
6626 041730  012701  047632                  MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
6627 041734  012702  045772                  MOV     #T17EXP,R2              ;EXPD ADDRESS
6628 041740  012703  000024                  MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
6629 041744  004737  011540                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
6630 041750                                  FORCERROR       82$,NOTSSR      ;@@D
6631 041760  103404                          BCS     90$                     ;BR IF YES
6632 041762                                  NEXT.ERRNO
6633 041762                          82$:    ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
     041762  104456                                                              TRAP    C$ERHRD
     041764  001151                                                              .WORD   617
     041766  046555                                                              .WORD   T171CMP
     041770  012410                                                              .WORD   MSGSTAT
6634 041772                          90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041772  104406                                                              TRAP    C$CLP1
6635
6636
```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```
6637
6638                                    ;  REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
6639                                    ;  TSTFLAG =1 FOR INCREMENT TEST PATTERN
6640                                    ;            =2 FOR DECREMENT TEST PATTERN
6641                                    ;            =3 FOR TSTBLK TABLE PATTERN
6642 041774  012737  000001  002316             MOV     #1,TSTFLAG              ;TEST PATTERN FLAG
6643 042002                             95$:
6644 042002  012737  000002  002312             MOV     #2,COUNT                ;GET FIRST BYTE COUNT
6645 042010                             100$:
6646                                    ;        Do a Write Subsystem WRITE NPR to set tape direction out
6647 042010  012700  000100                     MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
6648 042014  004737  047436                     JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
6649 042020  012704  047760                     MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6650 042024  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6651 042030  004737  016376                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6652 042034                                     FORCERROR       102$                    ;@@FORCE ERROR IF FORCER=1
6653 042050  103407                             BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
6654 042052  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6655 042054                                     NEXT.ERRNO
6656 042054                             102$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042054  104455                                                                     TRAP    C$ERDF
     042056  001152                                                                     .WORD   618
     042060  046403                                                                     .WORD   T174SSR
     042062  012106                                                                     .WORD   PKTSSR
6657 042064  005237  002222                     INC     FATFLG                  ;SET FATAL ERROR FLAG
6658 042070                             105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042070  104406                                                                     TRAP    C$CLP1
6659                                    ;        Do a Write Subsystem WRITE FIFO
6660 042072  004737  047536                     JSR     PC,T17CLEXP             ;CLEAR EXPD BUFFER
6661 042076  012701  046114                     MOV     #T17WFDATA,R1           ;EXPD WRITE FIFO DATA BUFFER
6662 042102  013702  002312                     MOV     COUNT,R2                ;TEST PATTERN SIZE
6663 042106  022737  000001  002316             CMP     #1,TSTFLAG              ;INCREMENT PATTERN THIS TIME THRU?
6664 042114  001005                             BNE     115$                    ;BR IF NO
6665 042116  005000                             CLR     R0                      ;INCREMENT TEST PATTERN
6666 042120  110021                     110$:   MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
6667 042122  005200                             INC     R0                      ;SET NEXT PATTERN
6668 042124  005302                             DEC     R2                      ;DONE?
6669 042126  003374                             BGT     110$                    ;BR IF NO
6670 042130  022737  000002  002316     115$:   CMP     #2,TSTFLAG              ;DECREMENT PATTERN THIS TIME THRU?
6671 042136  001006                             BNE     125$                    ;BR IF NO
6672 042140  012700  000377                     MOV     #377,R0                 ;DECREMENT TEST PATTERN
6673 042144  110021                     120$:   MOVB    R0,(R1)+                ;STORE DECREMENT TEST BYTE
6674 042146  005300                             DEC     R0                      ;SET NEXT PATTERN
6675 042150  005302                             DEC     R2                      ;DONE?
6676 042152  003374                             BGT     120$                    ;BR IF NO
6677 042154  022737  000003  002316     125$:   CMP     #3,TSTFLAG              ;TSTBLK PATTERNS THIS TIME THRU?
6678 042162  001005                             BNE     135$                    ;BR IF NO
6679 042164  012700  002754                     MOV     #TSTBLK,R0              ;FLOAT 1'S/0'S ETC. TEST TABLE
6680 042170  112021                     130$:   MOVB    (R0)+,(R1)+             ;STORE A TSTBLK BYTE
6681 042172  005302                             DEC     R2                      ;DONE?
6682 042174  003375                             BGT     130$                    ;BR IF NO
6683 042176                             135$:
6684 042176  013700  002312                     MOV     COUNT,R0                ;FIFO BYTE COUNT
6685 042202  012701  046114                     MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
6686 042206  004737  047462                     JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
6687 042212  012704  047760                     MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6688 042216  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```
6689 042222  004737  016376                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6690 042226                                    FORCERROR       142$                    ;@@FORCE ERROR IF FORCER=1
6691 042242  103407                            BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
6692 042244  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6693 042246                                    NEXT.ERRNO
6694 042246                          142$.     ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042246  104455                                                                    TRAP    C$ERDF
     042250  001153                                                                    .WORD   619
     042252  046446                                                                    .WORD   T175SSR
     042254  012106                                                                    .WORD   PKTSSR
6695 042256  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
6696 042262                          150$:     CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042262  104406                                                                    TRAP    C$CLP1
6697
6698                                 ;         Do a Write Subsystem READ STATUS
6699 042264  004737  047374                    JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
6700 042270  012704  047760                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6701 042274  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6702 042300  004737  016376                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6703 042304                                    FORCERROR       157$                    ;@@FORCE ERROR IF FORCER=1
6704 042320  103407                            BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
6705 042322  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6706 042324                                    NEXT.ERRNO
6707 042324                          157$:     ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042324  104455                                                                    TRAP    C$ERDF
     042326  001154                                                                    .WORD   620
     042330  046336                                                                    .WORD   T173SSR
     042332  012106                                                                    .WORD   PKTSSR
6708 042334  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
6709 042340                          160$:     CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042340  104406                                                                    TRAP    C$CLP1
6710
6711                                 ;         Set WORDS 0-7 of expd message buffer = to recv since not testing
6712 042342  004737  047556                    JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
6713 042346  012701  046012                    MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
6714 042352  012702  047652                    MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
6715 042356  012221                            MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
6716 042360  011211                            MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
6717 042362  052711  000020                    BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
6718 042366  052711  000040                    BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
6719 042372  042711  000200                    BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
6720 042376  042711  000100                    BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
6721                                 ;         If Input Ready NOT=1 then Print Error
6722                                 ;         If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
6723 042402  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
6724 042404  012701  047632                    MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
6725 042410  012702  045772                    MOV     #T17EXP,R2              ;EXPD ADDRESS
6726 042414  012703  000024                    MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
6727 042420  004737  011540                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
6728 042424                                    FORCERROR       162$,NOTSSR             ;@@D
6729 042434  103404                            BCS     170$                    ;BR IF YES
6730 042436                                    NEXT.ERRNO
6731 042436                          162$:     ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
     042436  104456                                                                    TRAP    C$ERHRD
     042440  001155                                                                    .WORD   621
     042442  046733                                                                    .WORD   T173CMP
     042444  012410                                                                    .WORD   MSGSTAT
```

F13

SEQ 0161

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```
6732 042446                          170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042446  104406                                                                  TRAP    C$CLP1
6733
6734                                 ;       Do Write Subsystem READ FIFO
6735 042450  013700  002312                 MOV     COUNT,R0                ;SET READ BYTE COUNT
6736 042454  004737  047516                 JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
6737 042460  012704  047760                 MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6738 042464  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6739 042470  004737  016376                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6740 042474                                 FORCERROR       172$                    ;##DFORCE ERROR IF FORCER=1
6741 042510  103407                         BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
6742 042512  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6743 042514                                 NEXT.ERRNO
6744 042514                          172$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042514  104455                                                                  TRAP    C$ERDF
     042516  001156                                                                  .WORD   622
     042520  046512                                                                  .WORD   T176SSR
     042522  012106                                                                  .WORD   PKTSSR
6745 042524  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
6746 042530                          180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042530  104406                                                                  TRAP    C$CLP1
6747
6748                                 ;       If Data read from FIFO NOT= to Data sent Then Print Error
6749 042532  005000                         CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
6750 042534  012702  046114                 MOV     #T17WFDATA,R2           ;GET EXPECTED ADDRESS FOR CKMSG2
6751 042540  012701  047652                 MOV     #T17BFSTA,R1            ;GET RECEIVED ADDRESS FOR CKMSG2
6752 042544  013703  002312                 MOV     COUNT,R3                ;NUMBER OF BYTES TO COMPARE
6753 042550  004737  011540                 JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
6754 042554                                 FORCERROR       192$,NOTSSR             ;##D
6755 042564  103406                         BCS     200$                    ;BR IF YES
6756 042566                                 NEXT.ERRNO
6757 042566  013701  002312          192$:   MOV     COUNT,R1                ;GET BYTE COUNT
6758 042572                                 ERRHRD  ERRNO,T175CMP,FIFEXP    ;REPORT ERROR
     042572  104456                                                                  TRAP    C$ERHRD
     042574  001157                                                                  .WORD   623
     042576  047102                                                                  .WORD   T175CMP
     042600  012230                                                                  .WORD   FIFEXP
6759 042602                          200$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042602  104406                                                                  TRAP    C$CLP1
6760
6761                                 ;       Do a Write Subsystem READ STATUS
6762 042604  004737  047374                 JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
6763 042610  012704  047760                 MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6764 042614  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6765 042620  004737  016376                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6766 042624                                 FORCERROR       212$                    ;##DFORCE ERROR IF FORCER=1
6767 042640  103407                         BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
6768 042642  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6769 042644                                 NEXT.ERRNO
6770 042644                          212$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042644  104455                                                                  TRAP    C$ERDF
     042646  001160                                                                  .WORD   624
     042650  046336                                                                  .WORD   T173SSR
     042652  012106                                                                  .WORD   PKTSSR
6771 042654  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
6772 042660                          220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042660  104406                                                                  TRAP    C$CLP1
```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```
6773                                    ;       Set WORDS 0 7 of expd message buffer = to recv since not testing
6774 042662  004737  047556                      JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
6775 042666  012701  046012                      MOV     @T17EXSTA,R1            ;GET EXPECTED READ STATUS
6776 042672  012702  047652                      MOV     @T17BFSTA,R2            ;GET RECV READ STATUS
6777 042676  012221                              MOV     (R2)+,(R1)+             ;SET EXPD WORD @8 = RECV TEMP
6778 042700  011211                              MOV     (R2),(R1)               ;SET EXPD WORD @9 = RECV TEMP
6779 042702  052711  000020                      BIS     @S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
6780 042706  042711  000040                      BIC     @S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
6781 042712  042711  000200                      BIC     @S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
6782 042716  042711  000100                      BIC     @S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
6783                                    ;       If Input Ready NOT=1 then Print Error
6784                                    ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
6785 042722  005000                              CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
6786 042724  012701  047632                      MOV     @T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
6787 042730  012702  045772                      MOV     @T17EXP,R2              ;EXPD ADDRESS
6788 042734  012703  000024                      MOV     @20.,R3                 ;NUMBER OF BYTES TO COMPARE
6789 042740  004737  011540                      JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
6790 042744                                      FORCERROR       232$,NOTSSR     ;@@D
6791 042754  103404                              BCS     240$                    ;BR IF YES
6792 042756                                      NEXT.ERRNO
6793 042756                             232$:    ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
     042756  104456                                                                      TRAP    C$ERHRD
     042760  001161                                                                      .WORD   625
     042762  047017                                                                      .WORD   T174CMP
     042764  012410                                                                      .WORD   MSGSTAT
6794 042766                             240$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042766  104406                                                                      TRAP    C$CLP1
6795 042770                                      FORCEXIT        250$            ;@@D
6796 043000  005237  002312                      INC     COUNT                   ;GET NEXT BYTE COUNT
6797 043004  023727  002312  000077              CMP     COUNT,@77               ;DONE 0 TO 77
6798 043012  101002                              BHI     250$                    ;BR IF YES
6799 043014  000137  042010                      JMP     100$                    ;DO ANOTHER BYTE COUNT
6800 043020  005237  002316             250$:    INC     TSTFLAG                 ;GET NEXT TEST PATTERN CODE
6801 043024  023727  002316  000003              CMP     TSTFLAG,@3              ;DONE INC,DEC,TSTBLK PATTERNS?
6802 043032  101002                              BHI     255$                    ;BR IF YES
6803 043034  000137  042002                      JMP     95$                     ;DO ANOTHER TEST PATTERN
6804 043040                             255$:
6805 043040                                      ENDSUB                          ;////////////// END SUBTEST //////////
     043040                                                                              L10061:
     043040  104403                                                                              TRAP    C$ESUB
6806
6807 043042  005737  002222                      TST     FATFLG                  ;ANY FATAL ERRORS ?
6808 043046  001402                              BEQ     260$                    ;BRANCH IF NOT
6809 043050  004737  017242                      JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
6810 043054                             260$:
6811
6812
6813
6814                                             .SBTTL  TEST  6: SUBTEST 4: FIFO Verify ILW Status
6815
6816                                    ;+*
6817                                    ; TEST 6: SUBTEST 4:
6818                                    ;
6819                                    ; SUBTEST DESCRIPTION:
6820                                    ;
6821                                    ;       This subtest verifies that reading the FIFO when it is
6822                                    ;       empty causes the Last Word (ILW) status to assert.
```

H13

TEST 6: SUBTEST 4: FIFO Verify ILW Status

```
                                        ;
                                        ; 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
                                        ;
043054                                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
043054                                                                                  T6.4:
043054  104402                                                                          TRAP    C$BSUB
                                        ;       Write to TSSR register to soft initialize the controller
043056                                  5$:
043056  004737  016034                          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
043062  103405                                  BCS     10$                     ;BR IF SOFT INIT OKAY
043064  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043066                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
043066  104455                                                                          TRAP    C$ERDF
043070  001161                                                                          .WORD   625
043072  003654                                                                          .WORD   SFIERR
043074  012074                                                                          .WORD   SFIMSG
                                        ;       Do a WRITE CHARACTERISTICS to setup a message buffer
043076  005037  002222                  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
043102  012704  047610                          MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
043106  004737  010662                          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
043112                                          FORCERROR       42$                     ;aaDFORCE ERROR IF FORCER=1
043126  103407                                  BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
043130  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043132                                          NEXT.ERRNO
043132                                  42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
043132  104455                                                                          TRAP    C$ERDF
043134  001162                                                                          .WORD   626
043136  046235                                                                          .WORD   T17SSR
043140  012106                                                                          .WORD   PKTSSR
043142  005237  002222                          INC     FATFLG                  ;SET FATAL ERROR FLAG
043146                                  50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
043146  104406                                                                          TRAP    C$CLP1
                                        ;       Do Write Subsystem READ FIFO with byte count equal to 1
043150  012700  000001                          MOV     #1,R0                   ;SET READ BYTE COUNT
043154  004737  047516                          JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
043160  012704  047760                          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
043164  010465  000000                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
043170  004737  016376                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
043174                                          FORCERROR       142$                    ;aaDFORCE ERROR IF FORCER=1
043210  103407                                  BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
043212  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043214                                          NEXT.ERRNO
043214                                  142$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
043214  104455                                                                          TRAP    C$ERDF
043216  001163                                                                          .WORD   627
```

I13

SEQ 0164

TEST 6: SUBTEST 4: FIFO Verify ILW Status

```
        043220  046512                                                                  .WORD   T176SSR
        043222  012106                                                                  .WORD   PKTSSR
 6867   043224  005237  002222                 INC     FATFLG                          ;SET FATAL ERROR FLAG
 6868   043230                         150$:   CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
        043230  104406                                                                  TRAP    C$CLP1
 6869
 6870                                   ;       Do a Write Subsystem READ STATUS
 6871   043232  004737  047374                 JSR     PC,T17SRD                       ;SETUP PACKET FOR READ STATUS
 6872   043236  012704  047760                 MOV     #T17PK2,R4                      ;GET WRITE SUBSYSTEM COMMAND PACKET
 6873   043242  010465  000000                 MOV     R4,TSDB(R5)                     ;SET THE PACKET ADDRESS TO EXECUTE
 6874   043246  004737  016376                 JSR     PC,CHKTSSR                      ;WAIT FOR SSR TO SET
 6875   043252                                 FORCERROR       162$                            ;@@DFORCE ERROR IF FORCER=1
 6876   043266  103407                         BCS     170$                            ;BR IF CARRY SET (GOOD RETURN)
 6877   043270  010001                         MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
 6878   043272                                 NEXT.ERRNO
 6879   043272                         162$:   ERRDF   ERRNO,T173SSR,PKTSSR            ;DEVICE FATAL SSR FAILED TO SET
        043272  104455                                                                  TRAP    C$ERDF
        043274  001164                                                                  .WORD   628
        043276  046336                                                                  .WORD   T173SSR
        043300  012106                                                                  .WORD   PKTSSR
 6880   043302  005237  002222                 INC     FATFLG                          ;SET FATAL ERROR FLAG
 6881   043306                         170$:   CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
        043306  104406                                                                  TRAP    C$CLP1
 6882                                   ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
 6883   043310  004737  047556                 JSR     PC,T17SETEXP                    ;SET WORDS 0 7 EXPD=RECV
 6884   043314  012701  046012                 MOV     #T17EXSTA,R1                    ;GET EXPECTED READ STATUS
 6885   043320  012702  047652                 MOV     #T17BFSTA,R2                    ;GET RECV READ STATUS
 6886   043324  012221                         MOV     (R2)+,(R1)+                     ;SET EXPD WORD #8 = RECV TEMP
 6887   043326  011211                         MOV     (R2),(R1)                       ;SET EXPD WORD #9 = RECV TEMP
 6888   043330  052711  000020                 BIS     #S2.INRDY,(R1)                  ;SET EXP INPUT READY= 1
 6889   043334  042711  000040                 BIC     #S2.OUTRDY,(R1)                 ;SET EXP OUTPUT READY= 0
 6890   043340  042711  000200                 BIC     #S2.DIM,(R1)                    ;SET EXP DATA IN MISS = 0
 6891   043344  052711  000100                 BIS     #S2.ILW,(R1)                    ;SET EXP LAST WORD (ILW)=1
 6892                                   ;       If Input Ready NOT=1 then Print Error
 6893                                   ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
 6894                                   ;       If Last Word (ILW) NOT=1 Then Print Error
 6895   043350  005000                         CLR     R0                              ;HIGH RECV ADDRESS FOR CKMSG2
 6896   043352  012701  047632                 MOV     #T17BFR,R1                      ;LOW RECV ADDRESS FOR CKMSG2
 6897   043356  012702  045772                 MOV     #T17EXP,R2                      ;EXPD ADDRESS
 6898   043362  012703  000024                 MOV     #20.,R3                         ;NUMBER OF BYTES TO COMPARE
 6899   043366  004737  011540                 JSR     PC,CKMSG2                       ;EXPD EQUAL RECV?
 6900   043372                                 FORCERROR       172$,NOTSSR             ;@@D
 6901   043402  103404                         BCS     180$                            ;BR IF YES
 6902   043404                                 NEXT.ERRNO
 6903   043404                         172$:   ERRHRD  ERRNO,T176CMP,MSGSTAT           ;REPORT ERROR
        043404  104456                                                                  TRAP    C$ERHRD
        043406  001165                                                                  .WORD   629
        043410  047156                                                                  .WORD   T176CMP
        043412  012410                                                                  .WORD   MSGSTAT
 6904   043414                         180$:   CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
        043414  104406                                                                  TRAP    C$CLP1
 6905
 6906   043416                                 ENDSUB                                  ;////////////// END SUBTEST //////////
        043416                                                                                  L10062:
        043416  104403                                                                  TRAP    C$ESUB
 6907
 6908   043420  005737  002222                 TST     FATFLG                          ;ANY FATAL ERRORS ?
```

J13


```
TEST  6: SUBTEST 4: FIFO Verify ILW Status

6909 043424  001402                        BEQ     260$                    ;BRANCH IF NOT
6910 043426  004737  017242                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
6911 043432                          260$:
6912
6913
6914                                        .SBTTL  TEST  6: SUBTEST 5: FIFO Verify Input Ready
6915
6916                                ;**
6917                                ; TEST 6: SUBTEST 5:
6918                                ;
6919                                ; SUBTEST DESCRIPTION:
6920                                ;
6921                                ;       This subtest verifies that writing 64. bytes into the FIFO
6922                                ;       without reading any out causes the Input Ready status to
6923                                ;       negate. The Subtest then verifies that writing a 65th byte
6924                                ;       into the FIFO causes the Data In Miss status to assert.
6925                                ;       Next it is verified that the original 64 bytes can be read
6926                                ;       out correctly and that the data has not been corrupted.
6927                                ;
6928                                ; TEST STEPS:
6929                                ;
6930                                ; BEGIN
6931                                ;       Write to TSSR to soft initialize
6932                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
6933                                ;       Do a Write Subsystem WRITE NPR to set tape direction out
6934                                ;       Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
6935                                ;       Do a Write Subsystem READ STATUS
6936                                ;       If Input Ready  NOT=0   Then Print Error
6937                                ;       If Output Ready NOT=1   Then Print Error
6938                                ;       If Data In Miss NOT=0   Then Print Error
6939                                ;       Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
6940                                ;       Do a Write Subsystem READ STATUS
6941                                ;       If Input Ready  NOT=0   Then Print Error
6942                                ;       If Output Ready NOT=1   Then Print Error
6943                                ;       If Data In Miss NOT=1   Then Print Error
6944                                ;       Do Write Subsystem READ FIFO
6945                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
6946                                ;       Do a Write Subsystem READ STATUS
6947                                ;       If Input Ready  NOT=1   Then Print Error
6948                                ;       If Output Ready NOT=0   Then Print Error
6949                                ;       If Data In Miss NOT=1   Then Print Error
6950                                ; END
6951                                ;
6952 043432                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////
      043432                                                                          T6.5:
      043432  104402                                                                    TRAP    C$BSUB
6953
6954                                ;       Write to TSSR register to soft initialize the controller
6955 043434                          5$:
6956 043434  004737  016034                JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
6957 043440  103405                        BCS     10$                     ;BR IF SOFT INIT OKAY
6958 043442  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6959 043444                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      043444  104455                                                                    TRAP    C$ERDF
      043446  001165                                                                    .WORD   629
      043450  003654                                                                    .WORD   SFIERR
      043452  012074                                                                    .WORD   SFIMSG
```

TEST 6: SUBTEST 5: FIFO Verify Input Ready

```
6960
6961 043454  005037  002222          ;       Do a WRITE CHARACTERISTICS to setup a message buffer
6962 043460  012704  047610  10$:     CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
6963 043464  004737  010662          MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
6964 043470                          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
6965 043504  103407                  FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
6966 043506  010001                  BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
6967 043510                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6968 043510                          NEXT.ERRNO
     043510  104455          42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     043512  001166                                                                  TRAP    C$ERDF
     043514  046235                                                                  .WORD   630
     043516  012106                                                                  .WORD   T17SSR
                                                                                     .WORD   PKTSSR
6969 043520  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
6970 043524                  50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043524  104406                                                                  TRAP    C$CLP1
6971
6972                                 ;       Do a Write Subsystem WRITE NPR to set tape direction out
6973 043526  012700  000100  100$:    MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
6974 043532  004737  047436          JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
6975 043536  012704  047760          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6976 043542  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6977 043546  004737  016376          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6978 043552                          FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
6979 043566  103407                  BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
6980 043570  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6981 043572                          NEXT.ERRNO
6982 043572                  102$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     043572  104455                                                                  TRAP    C$ERDF
     043574  001167                                                                  .WORD   631
     043576  046403                                                                  .WORD   T174SSR
     043600  012106                                                                  .WORD   PKTSSR
6983 043602  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
6984 043606                  105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043606  104406                                                                  TRAP    C$CLP1
6985
6986                                 ;       Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
6987 043610  012737  000100  002312          MOV     #64.,COUNT              ;WRITE 64 BYTES
6988 043616  012701  046114                  MOV     #T17WFDATA,R1           ;EXPD WRITE FIFO DATA BUFFER
6989 043622  012702  000100                  MOV     #64.,R2                 ;TEST PATTERN SIZE
6990 043626  005000                          CLR     R0                      ;INCREMENT TEST PATTERN
6991 043630  110021                  110$:   MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
6992 043632  005200                          INC     R0                      ;SET NEXT PATTERN
6993 043634  005302                          DEC     R2                      ;DONE?
6994 043636  003374                          BGT     110$                    ;BR IF NO
6995 043640  013700  002312                  MOV     COUNT,R0                ;FIFO BYTE COUNT
6996 043644  012701  046114                  MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
6997 043650  004737  047462                  JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
6998 043654  012704  047760                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
6999 043660  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7000 043664  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
7001 043670                                  FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
7002 043704  103407                          BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
7003 043706  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7004 043710                                  NEXT.ERRNO
7005 043710                          142$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     043710  104455                                                                  TRAP    C$ERDF
```

L13

SEQ 0167

TEST 6: SUBTEST 5: FIFO Verify Input Ready

```
        043712  001170                                                                  .WORD   632
        043714  046446                                                                  .WORD   T17SSR
        043716  012106                                                                  .WORD   PKTSSR
  7006  043720  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
  7007  043724                          150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        043724  104406                                                                  TRAP    C$CLP1
  7008
  7009                                  ;       Do a Write Subsystem READ STATUS
  7010                                  ;       If Input Ready  NOT=0   Then Print Error
  7011                                  ;       If Output Ready NOT=1   Then Print Error
  7012                                  ;       If Data In Miss NOT=0   Then Print Error
  7013  043726  004737  047374                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
  7014  043732  012704  047760                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
  7015  043736  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
  7016  043742  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
  7017  043746                                  FORCERROR       157$                    ;@@FORCE ERROR IF FORCER=1
  7018  043762  103407                          BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
  7019  043764  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  7020  043766                                  NEXT.ERRNO
  7021  043766                          157$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        043766  104455                                                                  TRAP    C$ERDF
        043770  001171                                                                  .WORD   633
        043772  046336                                                                  .WORD   T173SSR
        043774  012106                                                                  .WORD   PKTSSR
  7022  043776  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
  7023  044002                          160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        044002  104406                                                                  TRAP    C$CLP1
  7024                                  ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
  7025  044004  004737  047556                  JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
  7026  044010  012701  046012                  MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
  7027  044014  012702  047652                  MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
  7028  044020  012221                          MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
  7029  044022  011211                          MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
  7030  044024  042711  000020                  BIC     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 0
  7031  044030  052711  000040                  BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
  7032  044034  042711  000200                  BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
  7033  044040  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
  7034  044042  012701  047632                  MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
  7035  044046  012702  045772                  MOV     #T17EXP,R2              ;EXPD ADDRESS
  7036  044052  012703  000024                  MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
  7037  044056  004737  011540                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
  7038  044062                                  FORCERROR       162$,NOTSSR             ;@@D
  7039  044072  103404                          BCS     170$                    ;BR IF YES
  7040  044074                                  NEXT.ERRNO
  7041  044074                          162$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
        044074  104456                                                                  TRAP    C$ERHRD
        044076  001172                                                                  .WORD   634
        044100  046733                                                                  .WORD   T173CMP
        044102  012410                                                                  .WORD   MSGSTAT
  7042  044104                          170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        044104  104406                                                                  TRAP    C$CLP1
  7043
  7044
  7045                                  ;       Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
  7046  044106  012700  000001                  MOV     #1,R0                   ;FIFO BYTE COUNT
  7047  044112  012701  046114                  MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
  7048  044116  004737  047462                  JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
```

TEST 6: SUBTEST 5: FIFO Verify Input Ready

```
7049 044122  012704  047760                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
7050 044126  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7051 044132  004737  016376                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
7052 044136                                    FORCERROR       172$                    ;aaDFORCE ERROR IF FORCER=1
7053 044152  103407                            BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
7054 044154  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7055 044156                                    NEXT.ERRNO
7056 044156                            172$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044156  104455                                                                    TRAP    C$ERDF
     044160  001173                                                                    .WORD   635
     044162  046446                                                                    .WORD   T175SSR
     044164  012106                                                                    .WORD   PKTSSR
7057 044166  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
7058 044172                            180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044172  104406                                                                    TRAP    C$CLP1
7059
7060                                   ;       Do a Write Subsystem READ STATUS
7061                                   ;       If Input Ready  NOT=0   Then Print Error
7062                                   ;       If Output Ready NOT=1   Then Print Error
7063                                   ;       If Data In Miss NOT=1   Then Print Error
7064 044174  004737  047374                    JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
7065 044200  012704  047760                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
7066 044204  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7067 044210  004737  016376                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
7068 044214                                    FORCERROR       187$                    ;aaDFORCE ERROR IF FORCER=1
7069 044230  103407                            BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
7070 044232  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7071 044234                                    NEXT.ERRNO
7072 044234                            187$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044234  104455                                                                    TRAP    C$ERDF
     044236  001174                                                                    .WORD   636
     044240  046336                                                                    .WORD   T173SSR
     044242  012106                                                                    .WORD   PKTSSR
7073 044244  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
7074 044250                            190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044250  104406                                                                    TRAP    C$CLP1
7075                                   ;       Set WORDS 0 7 of expd message buffer = to recv since not testing
7076 044252  004737  047556                    JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
7077 044256  012701  046012                    MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
7078 044262  012702  047652                    MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
7079 044266  012221                            MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
7080 044270  011211                            MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
7081 044272  042711  000020                    BIC     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 0
7082 044276  052711  000040                    BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
7083 044302  052711  000200                    BIS     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 1
7084 044306  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
7085 044310  012701  047632                    MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
7086 044314  012702  045772                    MOV     #T17EXP,R2              ;EXPD ADDRESS
7087 044320  012703  000024                    MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
7088 044324  004737  011540                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
7089 044330                                    FORCERROR       192$,NOTSSR             ;aaD
7090 044340  103404                            BCS     200$                    ;BR IF YES
7091 044342                                    NEXT.ERRNO
7092 044342                            192$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
     044342  104456                                                                    TRAP    C$ERHRD
     044344  001175                                                                    .WORD   637
     044346  046733                                                                    .WORD   T173CMP
```

TEST 6: SUBTEST 5: FIFO Verify Input Ready

```
      044350  012410                                                          .WORD   MSGSTAT
 7093 044352                      200$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      044352  104406                                                          TRAP    C$CLP1
 7094                             ;       Do Write Subsystem READ FIFO
 7095 044354  013700  002312              MOV     COUNT,R0                ;SET READ BYTE COUNT
 7096 044360  004737  047516              JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
 7097 044364  012704  047760              MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 7098 044370  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 7099 044374  004737  016376              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 7100 044400                              FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
 7101 044414  103407                      BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
 7102 044416  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7103 044420                              NEXT.ERRNO
 7104 044420                      212$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      044420  104455                                                          TRAP    C$ERDF
      044422  001176                                                          .WORD   638
      044424  046512                                                          .WORD   T176SSR
      044426  012106                                                          .WORD   PKTSSR
 7105 044430  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
 7106 044434                      220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      044434  104406                                                          TRAP    C$CLP1
 7107
 7108                             ;       If Data read from FIFO NOT= to Data sent Then Print Error
 7109 044436  005000                      CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 7110 044440  012702  046114              MOV     #T17WFDATA,R2           ;GET EXPECTED ADDRESS FOR CKMSG2
 7111 044444  012701  047652              MOV     #T17BFSTA,R1            ;GET RECEIVED ADDRESS FOR CKMSG2
 7112 044450  013703  002312              MOV     COUNT,R3                ;NUMBER OF BYTES TO COMPARE
 7113 044454  004737  011540              JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 7114 044460                              FORCERROR       232$,NOTSSR     ;@@D
 7115 044470  103406                      BCS     240$                    ;BR IF YES
 7116 044472                              NEXT.ERRNO
 7117 044472  013701  002312      232$:   MOV     COUNT,R1                ;GET BYTE COUNT
 7118 044476                              ERRHRD  ERRNO,T175CMP,FIFEXP    ;REPORT ERROR
      044476  104456                                                          TRAP    C$ERHRD
      044500  001177                                                          .WORD   639
      044502  047102                                                          .WORD   T175CMP
      044504  012230                                                          .WORD   FIFEXP
 7119 044506                      240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      044506  104406                                                          TRAP    C$CLP1
 7120
 7121                             ;       Do a Write Subsystem READ STATUS
 7122                             ;       If Input Ready  NOT=1   Then Print Error
 7123                             ;       If Output Ready NOT=0   Then Print Error
 7124                             ;       If Data In Miss NOT=1   Then Print Error
 7125 044510  004737  047374              JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
 7126 044514  012704  047760              MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 7127 044520  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 7128 044524  004737  016376              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 7129 044530                              FORCERROR       252$                    ;@@DFORCE ERROR IF FORCER=1
 7130 044544  103407                      BCS     260$                    ;BR IF CARRY SET (GOOD RETURN)
 7131 044546  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7132 044550                              NEXT.ERRNO
 7133 044550                      252$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      044550  104455                                                          TRAP    C$ERDF
      044552  001200                                                          .WORD   640
      044554  046336                                                          .WORD   T173SSR
      044556  012106                                                          .WORD   PKTSSR
```

TEST 6: SUBTEST 5: FIFO Verify Input Ready

```
7134 044560  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
7135 044564                          260$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044564  104406                                                                  TRAP    C$CLP1
7136                                 ;        Set WORDS 0-7 of expd message buffer = to recv since not testing
7137 044566  004737  047556                  JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
7138 044572  012701  046012                  MOV     @T17EXSTA,R1            ;GET EXPECTED READ STATUS
7139 044576  012702  047652                  MOV     @T17BFSTA,R2            ;GET RECV READ STATUS
7140 044602  012221                          MOV     (R2)+,(R1)+             ;SET EXPD WORD @8 = RECV TEMP
7141 044604  011211                          MOV     (R2),(R1)               ;SET EXPD WORD @9 = RECV TEMP
7142 044606  052711  000020                  BIS     @S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
7143 044612  042711  000040                  BIC     @S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
7144 044616  052711  000200                  BIS     @S2.DIM,(R1)            ;SET EXP DATA IN MISS = 1
7145 044622  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
7146 044624  012701  047632                  MOV     @T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
7147 044630  012702  045772                  MOV     @T17EXP,R2              ;EXPD ADDRESS
7148 044634  012703  000024                  MOV     @20.,R3                 ;NUMBER OF BYTES TO COMPARE
7149 044640  004737  011540                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
7150 044644                                  FORCERROR       272$,NOTSSR     ;@@D
7151 044654  103404                          BCS     280$                    ;BR IF YES
7152 044656                                  NEXT.ERRNO
7153 044656                          272$:   ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
     044656  104456                                                                  TRAP    C$ERHRD
     044660  001201                                                                  .WORD   641
     044662  047017                                                                  .WORD   T174CMP
     044664  012410                                                                  .WORD   MSGSTAT
7154 044666                          280$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044666  104406                                                                  TRAP    C$CLP1
7155
7156 044670                                  ENDSUB                          ;////////////// END SUBTEST //////////
     044670                                                                  L10063:
     044670  104403                                                                  TRAP    C$ESUB
7157
7158 044672  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
7159 044676  001402                          BEQ     300$                    ;BRANCH IF NOT
7160 044700  004737  017242                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
7161 044704                          300$
7162
7163
7164
7165                                         .SBTTL  TEST  6: SUBTEST 6: FIFO Verify Reset FIFO Test
7166
7167                                 ;+
7168                                 ; TEST 6: SUBTEST 6:
7169                                 ;
7170                                 ; SUBTEST DESCRIPTION:
7171                                 ;
7172                                 ;       This subtest verifies that the Reset FIFO function within
7173                                 ;       the Write Miscellaneous Control 1 function initializes
7174                                 ;       the FIFO to correct intial status. The following steps
7175                                 ;       are performed:
7176                                 ;               1. Reset an already initialized FIFO and check for
7177                                 ;                  proper status.
7178                                 ;               2. Write a varying number of bytes (1-65.) into the
7179                                 ;                  FIFO and verify that after each block of bytes is
7180                                 ;                  written the FIFO can be be reset to it's initial
7181                                 ;                  state.
7182                                 ;
```

```
TEST  6: SUBTEST 6: FIFO Verify Reset FIFO Test

                                        ; TEST STEPS:
                                        ;
                                        ; BEGIN
                                        ;       Write to TSSR to soft initialize
                                        ;       Do a WRITE CHARACTERISTICS to setup a message buffer
                                        ;       Do a Write Subsystem Write Misc to Reset FIFO
                                        ;       Do a Write Subsystem READ STATUS
                                        ;       If all Tape Status 2 (ICER,IFMK,IHER) flip flop
                                        ;         signals NOT=0 Then Print Error
                                        ;       Do a Write Subsystem WRITE NPR to set tape direction out
                                        ;
                                        ;  REPEAT FOR BYTE COUNT 1 TO 65.
                                        ;    BEGIN
                                        ;       Do a Write Subsystem WRITE FIFO with the current byte count
                                        ;       Do a Write Subsystem Write Misc to Reset FIFO
                                        ;       Do a Write Subsystem READ STATUS
                                        ;       If all Tape Status 2 (ICER,IFMK,IHER) flip flop
                                        ;         signals NOT=0 Then Print Error
                                        ;    END
                                        ; -
044704                                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
044704                                                                                  T6.6:
044704  104402                                                                          TRAP    C$BSUB

                                        ;       Write to TSSR register to soft initialize the controller
                                        5$:
044706  004737  016034                          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
044712  103405                          10$     BCS     10$                     ;BR IF SOFT INIT OKAY
044714  010001                          50$     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
044716                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
044716  104455                                                                          TRAP    C$ERDF
044720  001201                                                                          .WORD   641
044722  003654                                                                          .WORD   SFIERR
044724  012074                                                                          .WORD   SFIMSG
                                        ;       Do a WRITE CHARACTERISTICS to setup a message buffer
044726  005037  002222                  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
044732  012704  047610                          MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
044736  004737  010662                          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
044742                                          FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
044756  103407                                  BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
044760  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
044762                                          NEXT.ERRNO
044762                                  42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
044762  104455                                                                          TRAP    C$ERDF
044764  001202                                                                          .WORD   642
044766  046235                                                                          .WORD   T17SSR
044770  012106                                                                          .WORD   PKTSSR
044772  005237  002222                          INC     FATFLG                  ;SET FATAL ERROR FLAG
044776                                  50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
044776  104406                                                                          TRAP    C$CLP1
                                        ;       Do a Write Subsystem Write Misc to Reset FIFO
045000  004737  047414                          JSR     PC,T17RSFIF             ;SETUP PKT FOR WRITE MISC RESET FIFO
045004  012704  047760                          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
045010  010465  000000                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
045014  004737  016376                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
045020                                          FORCERROR       62$                     ;@@FORCE ERROR IF FORCER=1
045034  103407                                  BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```
7229 045036  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7230 045040                                 NEXT.ERRNO
7231 045040                          62$:   ERRDF   ERRNO,T172SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045040  104455                                                          TRAP    C$ERDF
     045042  001203                                                          .WORD   643
     045044  046272                                                          .WORD   T172SSR
     045046  012106                                                          .WORD   PKTSSR
7232 045050  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
7233 045054                          70$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045054  104406                                                          TRAP    C$CLP1
7234
7235                                 ;      Do a Write Subsystem READ STATUS
7236                                 ;      If all Tape Status 2 (ICER,IFMK,IHER) flip flop
7237                                 ;        signals NOT=0 Then Print Error
7238 045056  004737  047374                 JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
7239 045062  012704  047760                 MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
7240 045066  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7241 045072  004737  016376                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
7242 045076                                 FORCERROR       77$             ;aaDFORCE ERROR IF FORCER=1
7243 045112  103407                        BCS     80$                     ;BR IF CARRY SET (GOOD RETURN)
7244 045114  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7245 045116                                 NEXT.ERRNO
7246 045116                          77$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045116  104455                                                          TRAP    C$ERDF
     045120  001204                                                          .WORD   644
     045122  046336                                                          .WORD   T173SSR
     045124  012106                                                          .WORD   PKTSSR
7247 045126  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
7248 045132                          80$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045132  104406                                                          TRAP    C$CLP1
7249 045134  004737  047556                 JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
7250 045140  012701  046012                 MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
7251 045144  012702  047652                 MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
7252 045150  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
7253 045152  042711  002000                 BIC     #S1.ICER,(R1)           ;SET EXPD ICER  =0
7254 045156  042711  001000                 BIC     #S1.IFMK,(R1)           ;SET EXPD IFMK  =0
7255 045162  042711  000400                 BIC     #S1.IHER,(R1)           ;SET EXPD IHER  =0
7256 045166  016261  000002  000002         MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
7257 045174  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
7258 045176  012701  047632                 MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
7259 045202  012702  045772                 MOV     #T17EXP,R2              ;EXPD ADDRESS
7260 045206  012703  000024                 MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
7261 045212  004737  011540                 JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
7262 045216                                 FORCERROR       92$,NOTSSR      ;aaD
7263 045226  103404                        BCS     100$                    ;BR IF YES
7264 045230                                 NEXT.ERRNO
7265 045230                          92$:   ERRHRD  ERRNO,T177CMP,MSGSTAT   ;REPORT ERROR
     045230  104456                                                          TRAP    C$ERHRD
     045232  001205                                                          .WORD   645
     045234  047264                                                          .WORD   T177CMP
     045236  012410                                                          .WORD   MSGSTAT
7266 045240                          100$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045240  104406                                                          TRAP    C$CLP1
7267
7268                                 ;      Do a Write Subsystem WRITE NPR to set tape direction out
7269 045242  012700  000100                 MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
7270 045246  004737  047436                 JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
```

E14

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```
7271 045252  012704  047760                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
7272 045256  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7273 045262  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
7274 045266                                FORCERROR       112$                    ;@@FORCE ERROR IF FORCER=1
7275 045302  103407                        BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
7276 045304  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7277 045306                                NEXT.ERRNO
7278 045306                        112$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045306  104455                                                                 TRAP    C$ERDF
     045310  001206                                                                 .WORD   646
     045312  046403                                                                 .WORD   T174SSR
     045314  012106                                                                 .WORD   PKTSSR
7279 045316  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
7280 045322                        120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045322  104406                                                                 TRAP    C$CLP1
7281
7282                               ;       Setup incrementing pattern in FIFO data buffer
7283 045324  012701  046012                MOV     #T17EXSTA,R1            ;EXPD WRITE FIFO DATA BUFFER
7284 045330  012702  000100                MOV     #64.,R2                 ;TEST PATTERN SIZE
7285 045334  005000                        CLR     R0                      ;INCREMENT TEST PATTERN
7286 045336  110021                130$:   MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
7287 045340  005200                        INC     R0                      ;SET NEXT PATTERN
7288 045342  005302                        DEC     R2                      ;DONE?
7289 045344  003374                        BGT     130$                    ;BR IF NO
7290
7291                               ;  REPEAT FOR BYTE COUNT 1 TO 65.
7292 045346  012737  000001  002312        MOV     #1,COUNT                ;GET FIRST BYTE COUNT
7293                               ;       Do a Write Subsystem WRITE FIFO with the current byte count
7294 045354                        150$:                                   ;REPEAT LOOP LABEL
7295 045354  013700  002312                MOV     COUNT,R0                ;FIFO BYTE COUNT
7296 045360  012701  046012                MOV     #T17EXSTA,R1            ;FIFO WRITE DATA ADDRESS
7297 045364  004737  047462                JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
7298 045370  012704  047760                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
7299 045374  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7300 045400  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
7301 045404                                FORCERROR       152$                    ;@@FORCE ERROR IF FORCER=1
7302 045420  103407                        BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
7303 045422  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7304 045424                                NEXT.ERRNO
7305 045424                        152$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045424  104455                                                                 TRAP    C$ERDF
     045426  001207                                                                 .WORD   647
     045430  046446                                                                 .WORD   T175SSR
     045432  012106                                                                 .WORD   PKTSSR
7306 045434  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
7307 045440                        160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045440  104406                                                                 TRAP    C$CLP1
7308
7309                               ;       Do a Write Subsystem Write Misc to Reset FIFO
7310 045442  004737  047414                JSR     PC,T17RSFIF             ;SETUP PKT FOR WRITE MISC RESET FIFO
7311 045446  012704  047760                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
7312 045452  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7313 045456  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
7314 045462                                FORCERROR       162$                    ;@@FORCE ERROR IF FORCER=1
7315 045476  103407                        BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
7316 045500  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7317 045502                                NEXT.ERRNO
```

F14

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```
7318 045502                          162$:   ERRDF   ERRNO,T172SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     045502  104455                                                                  TRAP    C$ERDF
     045504  001210                                                                  .WORD   648
     045506  046272                                                                  .WORD   T172SSR
     045510  012106                                                                  .WORD   PKTSSR
7319 045512  005237  002222                  INC     FATFLG                 ;SET FATAL ERROR FLAG
7320 045516                          170$:   CKLOOP                         ;LOOP ON ERROR, IF FLAG SET
     045516  104406                                                                  TRAP    C$CLP1
7321
7322                                 ;       Do a Write Subsystem READ STATUS
7323                                 ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
7324                                 ;         signals NOT=0 Then Print Error
7325 045520  004737  047374                  JSR     PC,T17SRD              ;SETUP PACKET FOR READ STATUS
7326 045524  012704  047760                  MOV     #T17PK2,R4             ;GET WRITE SUBSYSTEM COMMAND PACKET
7327 045530  010465  000000                  MOV     R4,TSDB(R5)            ;SET THE PACKET ADDRESS TO EXECUTE
7328 045534  004737  016376                  JSR     PC,CHKTSSR             ;WAIT FOR SSR TO SET
7329 045540                                  FORCERROR       177$                    ;@@DFORCE ERROR IF FORCER=1
7330 045554  103407                          BCS     180$                   ;BR IF CARRY SET (GOOD RETURN)
7331 045556  010001                          MOV     R0,R1                  ;SAVE CONTENTS OF TSSR
7332 045560                                  NEXT.ERRNO
7333 045560                          177$:   ERRDF   ERRNO,T173SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     045560  104455                                                                  TRAP    C$ERDF
     045562  001211                                                                  .WORD   649
     045564  046336                                                                  .WORD   T173SSR
     045566  012106                                                                  .WORD   PKTSSR
7334 045570  005237  002222                  INC     FATFLG                 ;SET FATAL ERROR FLAG
7335 045574                          180$:   CKLOOP                         ;LOOP ON ERROR, IF FLAG SET
     045574  104406                                                                  TRAP    C$CLP1
7336 045576  004737  047556                  JSR     PC,T17SETEXP           ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
7337 045602  012701  046012                  MOV     #T17EXSTA,R1           ;GET EXPECTED READ STATUS
7338 045606  012702  047652                  MOV     #T17BFSTA,R2           ;GET RECV READ STATUS
7339 045612  011211                          MOV     (R2),(R1)              ;SET EXPD WORD #8 = RECV TEMP
7340 045614  042711  002000                  BIC     #S1.ICER,(R1)          ;SET EXPD ICER  =0
7341 045620  042711  001000                  BIC     #S1.IFMK,(R1)          ;SET EXPD IFMK  =0
7342 045624  042711  000400                  BIC     #S1.IHER,(R1)          ;SET EXPD IHER  =0
7343 045630  016261  000002  000002          MOV     2(R2),2(R1)            ;SET EXPD WORD #9 = RECV (NOT TESTING)
7344 045636  005000                          CLR     R0                     ;HIGH RECV ADDRESS FOR CKMSG2
7345 045640  012701  047632                  MOV     #T17BFR,R1             ;LOW RECV ADDRESS FOR CKMSG2
7346 045644  012702  045772                  MOV     #T17EXP,R2             ;EXPD ADDRESS
7347 045650  012703  000024                  MOV     #20.,R3                ;NUMBER OF BYTES TO COMPARE
7348 045654  004737  011540                  JSR     PC,CKMSG2              ;EXPD EQUAL RECV?
7349 045660                                  FORCERROR       192$,NOTSSR     ;@@D
7350 045670  103404                          BCS     200$                   ;BR IF YES
7351 045672                                  NEXT.ERRNO
7352 045672                          192$:   ERRHRD  ERRNO,T177CMP,MSGSTAT  ;REPORT ERROR
     045672  104456                                                                  TRAP    C$ERHRD
     045674  001212                                                                  .WORD   650
     045676  047264                                                                  .WORD   T177CMP
     045700  012410                                                                  .WORD   MSGSTAT
7353 045702                          200$:   CKLOOP                         ;LOOP ON ERROR, IF FLAG SET
     045702  104406                                                                  TRAP    C$CLP1
7354
7355
7356 045704                          250$:
7357 045704                                  FORCEXIT        260$
7358 045714  005237  002312                  INC     COUNT                  ;GET NEXT BYTE COUNT
7359 045720  023727  002312  000101          CMP     COUNT,#65.             ;DONE ALL BYTES?
```

```
 7360 045726  101002                    BHI     260$                    ;BR IF YES
 7361 045730  000137  045354             JMP     150$                    ;DO ANOTHER BYTE COUNT
 7362 045734                        260$:
 7363
 7364 045734                             ENDSUB                          ;////////////// END SUBTEST //////////
      045734                                                                     L10064:
      045734  104403                                                                 TRAP    C$ESUB
 7365
 7366 045736  005737  002222             TST     FATFLG                  ;ANY FATAL ERRORS ?
 7367 045742  001402                     BEQ     300$                    ;BRANCH IF NOT
 7368 045744  004737  017242             JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 7369 045750  004737  016516        300$: JSR     PC,TSTLOOP              ;DO ITERATIONS?
 7370 045754  103002                     BCC     305$                    ;BR IF NO
 7371 045756  000137  040014             JMP     T17LOOP                 ;LOOP UNTIL ITERATIONS DONE
 7372 045762                        305$:
 7373
 7374 045762                             EXIT    TST                     ;////////// EXIT TEST //////////
      045762  104432                                                                 TRAP    C$EXIT
      045764  002112                                                                 .WORD   L10056 .
 7375
 7376
 7377                                   ;+
 7378                                   ;LOCAL STORAGE FOR THIS TEST
 7379                                   ;-
 7380
 7381
 7382 045766                            T17MSK:                         ;MASK OF UNTESTED BITS IN READ STATUS BYTES
 7383                                                                   ;UNTESTED BITS ARE SET TO 1
 7384 045766      377                           .BYTE   ↑C<000>         ;BYTE 0 MASK
 7385 045767      037                           .BYTE   ↑C<340>         ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
 7386 045770      360                           .BYTE   ↑C<017>         ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
 7387 045771      000                           .BYTE   0               ;MAKE IT EVEN
 7388
 7389 045772                            T17EXP:                         ;BEGIN EXPECTED DATA BUFFER
 7390 045772  000000                           .WORD   0               ;MESSAGE TYPE
 7391 045774  000000                           .WORD   0               ;DATA FIELD LENGTH
 7392 045776  000000                           .WORD   0               ;RBPCR
 7393 046000  000000                           .WORD   0               ;XST0
 7394 046002  000000                           .WORD   0               ;XST1
 7395 046004  000000                           .WORD   0               ;XST2
 7396 046006  000000                           .WORD   0               ;XST3
 7397 046010  000000                           .WORD   0               ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
 7398 046012                            T17EXSTA: .BLKB 66.             ;EXPECTED READ STATUS AND WRITE FIFO DATA
 7399 046114                            T17EXEND:                               ;END EXPECTED DATA BUFFER
 7400
 7401 046114                            T17WFDATA: .BLKB 66.            ;WRITE FIFO EXPECTED DATA BUFFER
 7402
 7403                                   ;+
 7404                                   ;LOCAL TEXT MESSAGES FOR TEST
 7405                                   ;-
 7406
 7407 046216      106  111  106  T17TID:        .ASCIZ  'FIFO Exerciser'
 7408 046235      127  122  111  T17SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
 7409 046272      127  122  111  T172SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
 7410 046336      127  122  111  T173SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
 7411 046403      127  122  111  T174SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
 7412 046446      127  122  111  T175SSR:.ASCIZ  'WRITE SUBSYSTEM (Write FIFO) Failed'
```

```
7413 046512     127     122     111  T176SSR:.ASCIZ  'WRITE SUBSYSTEM (Read FIFO) Failed'
7414 046555     106     111     106  T171CMP:.ASCIZ  'FIFO Status in WORD #9 Incorrect after Initialize'
7415 046637     122     145     141  T172CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
7416 046733     106     111     106  T173CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after WRITE FIFO'
7417 047017     106     111     106  T174CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after READ FIFO'
7418 047102     122     145     141  T175CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO Data'
7419 047156     106     111     106  T176CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after READ FIFO from an Empty FIFO'
7420 047264     106     111     106  T177CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after RESET FIFO'
7421                                              .EVEN
7422
7423                                 ;*
7424                                 ; CLEAR MESSAGE BUFFER
7425                                 ;
7426 047350                          T17CLRBUF:                                      ;
7427 047350                                  SAVREG                                  ;SAVE R1-R5 UNTIL NEXT RETURN
7428 047354  012701  047632                 MOV     #T17BFR,R1                      ;GET MESSAGE BUFFER ADDRESS
7429 047360  012702  000120                 MOV     #T17BEND-T17BFR,R2              ;SIZE OF MESSAGE BUFFER IN BYTES
7430 047364  105021                  10$:    CLRB    (R1)+                           ;CLEAR A BYTE
7431 047366  005302                          DEC     R2                              ;DONE?
7432 047370  003375                          BGT     10$                             ;BR IF NO
7433 047372  000207                          RTS     PC                              ;RETURN
7434
7435                                 ;*
7436                                 ; SETUP T17PK2 PACKET FOR READ STATUS
7437                                 ;-
7438 047374                          T17SRD:
7439 047374  004737  047350                 JSR     PC,T17CLRBUF                    ;CLEAR MESSAGE BUFFER
7440 047400  012700  047770                 MOV     #T17DT2,R0                      ;WRITE SUBSYSTEM DATA BUFFER
7441 047404  112720  000005                 MOVB    #PW.RDSTATUS,(R0)+              ;STORE READ STATUS COMMAND IN BSEL0
7442 047410  105010                          CLRB    (R0)                            ;CLEAR BSEL1
7443 047412  000207                          RTS     PC                              ;RETURN
7444
7445
7446                                 ;*
7447                                 ; SETUP T17PK2 PACKET FOR WRITE MISC RESET FIFO
7448 047414                                 ;-
7449 047414  004737  047350          T17RSFIF:
7450 047420  012700  047770                 JSR     PC,T17CLRBUF                    ;CLEAR MESSAGE BUFFER
7451 047424  112720  000010                 MOV     #T17DT2,R0                      ;WRITE SUBSYSTEM DATA BUFFER
7452 047430  112710  000030                 MOVB    #PW.WMISC,(R0)+                 ;STORE WRITE MISCELLANEOUS IN BSEL0
7453 047434  000207                         MOVB    #MS.RSFIF!MS.RSTAP,(R0)         ;STORE BSEL1 CLEAR FIFO CODES
7454                                         RTS     PC                              ;RETURN
7455
7456                                 ;*
7457                                 ; SETUP T17PK2 PACKET FOR WRITE NPR
7458                                 ;
7459                                 ; INPUT:
7460                                 ;       R0 CONTAINS BSEL1 NPR DATA
7461                                 ;
7462                                 ;       SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
7463 047436                                 ;-
7464 047436  004737  047350          T17SNPR:
7465 047442  012701  047770                 JSR     PC,T17CLRBUF                    ;CLEAR MESSAGE BUFFER
7466 047446  112721  000011                 MOV     #T17DT2,R1                      ;WRITE SUBSYSTEM DATA BUFFER
7467 047452  052700  000020                 MOVB    #PW.WNPR,(R1)+                  ;STORE WRITE NPR IN BSEL0
7468 047456  110011                         BIS     #NP.WRP,R0                      ;DON'T WRITE WRONG PARITY
7469 047460  000207                         MOVB    R0,(R1)                         ;STORE NPR DATA IN BSEL1
                                            RTS     PC                              ;RETURN
```

```
7470
7471                                    ;*
7472                                    ; SETUP T17PK2 PACKET FOR WRITE FIFO
7473                                    ;
7474                                    ; INPUT:
7475                                    ;       R0 CONTAINS BYTE COUNT
7476                                    ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
7477                                    ;
7478 047462                            T17WFIF:
7479 047462                                    SAVREG                          ;SAVE R1 R5 UNTIL NEXT RETURN
7480 047466  004737  047350                    JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
7481 047472  012702  047770                    MOV     #T17DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
7482 047476  112722  000004                    MOVB    #PW.WFIFO,(R2)+         ;STORE WRITE FIFO IN BSEL0
7483 047502  110022                            MOVB    R0,(R2)+                ;STORE BYTE COUNT IN BSEL1
7484 047504  005022                            CLR     (R2)+                   ;CLEAR SEL2 (UNUSED)
7485 047506  112122                    10$:    MOVB    (R1)+,(R2)+             ;STORE DATA PATTERN BYTE
7486 047510  005300                            DEC     R0                      ;DONE ALL BYTES?
7487 047512  003375                            BGT     10$                     ;BR IF NO
7488 047514  000207                            RTS     PC                      ;RETURN
7489
7490                                    ;*
7491                                    ; SETUP T17PK2 PACKET FOR READ FIFO
7492                                    ;
7493                                    ; INPUT:
7494                                    ;       R0 CONTAINS SEL2 BYTE COUNT
7495                                    ;-
7496 047516                            T17RFIF:
7497 047516  004737  047350                    JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
7498 047522  012701  047770                    MOV     #T17DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
7499 047526  112721  000003                    MOVB    #PW.RFIFO,(R1)+         ;STORE READ FIFO IN BSEL0
7500 047532  110021                            MOVB    R0,(R1)+                ;STORE BYTE COUNT IN BSEL1
7501 047534  000207                            RTS     PC                      ;RETURN
7502
7503                                    ;*
7504                                    ; CLEAR EXPECTED DATA MESSAGE BUFFER
7505 047536                            ;
7506 047536  012701  045772            T17CLEXP:
7507 047542  012700  000122                    MOV     #T17EXP,R1              ;GET EXPD ADDRESS
7508 047546  105021                            MOV     #T17EXEND-T17EXP,R0     ;GET EXPD SIZE
7509 047550  005300                    10$:    CLRB    (R1)+                   ;CLEAR A BYTE
7510 047552  003375                            DEC     R0                      ;DONE?
7511 047554  000207                            BGT     10$                     ;BR IF NO
7512                                           RTS     PC                      ;RETURN
7513                                    ;*
7514                                    ;Set WORDS 0 7 of expd message buffer = to recv since not testing
7515                                    ;
7516 047556                            T17SETEXP:
7517 047556  012702  045772                    MOV     #T17EXP,R2              ;GET EXPD
7518 047562  012703  047632                    MOV     #T17BFR,R3              ;GET READ STATUS RECV BUFFER
7519 047566  012700  000010                    MOV     #8.,R0                  ;SET WORDS 0 7 EXP=RECV
7520 047572  012322                    5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
7521 047574  005300                            DEC     R0                      ;DONE WORDS 0 7 WORDS?
7522 047576  003375                            BGT     5$                      ;BR IF NO
7523 047600  000207                            RTS     PC                      ;RETURN
7524
7526         047610                            .=<.+10>&177770
7528                                    ;
```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```
                                        ;WRITE CHARACTERISTICS COMMAND PACKET
                                        ;
047610                                  T17PACKET:                              ;COMMAND PACKET FOR TEST
047610  100004                                .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
047612  047620                                .WORD   T17DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
047614  000000                                .WORD   0
047616  000012                                .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE

047620                                  T17DATA:                                ;CHARACTERISTICS DATA BLOCK
047620  047632                                .WORD   T17BFR                  ;ADDRESS OF MESSAGE BUFFER
047622  000000                                .WORD   0
047624  000024                                .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
047626  000000                                .WORD   0                       ;ESS,ENB,EAI,ERI
047630  000000                                .WORD   0                       ;EXTENDED FEATURES UNIT NO. ETC.


                                        ;MESSAGE BUFFER FOR ALL TEST 6 COMMANDS

047632                                  T17BFR:                                 ;BEGIN MESSAGE BUFFER
047632  000000                                .WORD   0                       ;MESSAGE TYPE
047634  000000                                .WORD   0                       ;DATA FIELD LENGTH
047636  000000                                .WORD   0                       ;RBPCR
047640  000000                                .WORD   0                       ;XST0
047642  000000                                .WORD   0                       ;XST1
047644  000000                                .WORD   0                       ;XST2
047646  000000                                .WORD   0                       ;XST3
047650  000000                                .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
047652                                  T17BFSTA: .BLKB 64.                     ;READ STATUS AND WRITE FIFO BUFFER
047752                                  T17BEND:                                ;END OF MESSAGE BUFFER
                                        ;
                                        ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
                                        ;
                                                .=<.+10>&177770
047760                                  T17PK2:
047760  100006                                .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
047762  047770                                .WORD   T17DT2                  ;LOW ADDRESS OF DATA BLOCK
047764  000000                                .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
047766  000012                                .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE

047770                                  T17DT2:                                 ;DATA BLOCK
047770     000                                .BYTE   0                       ;BSEL0
047771     000                                .BYTE   0                       ;BSEL1
047772  000000                                .WORD   0                       ;SEL2
047774                                        .BLKB   66.                     ;WRITE FIFO DATA OUTPUT BUFFER

                                        ENDTST

050076                                                                  L10056:
050076                                                                          TRAP    C$ETST
050076  104401
                                        .SBTTL  TEST  7: STATIC TRANSPORT BUS INTERFACE TEST
                                        ;*
                                        ; TEST DESCRIPTION:
                                        ;
                                        ;
                                        ; TEST STEPS:
                                        ;
                                        ;    REPEAT FOR LOOPCNT
```

K14

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
050100                                                                                  T7::
7603 050100 012700 050606               MOV     #TST18ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
7604 050104 004737 016550               JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
7605 050110 012737 000012 002216        MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
7606 050116                             T18LOOP:
                                        ;
                                        ;       Write to TSSR register to soft initialize the controller
7608 050116                             5$:
7609 050116 004737 016034               JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
7610 050122 103405                      BCS     10$                     ;BR IF SOFT INIT OKAY
7611 050124 010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7612 050126                             ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
050126 104455                                                                           TRAP    C$ERDF
050130 001274                                                                           .WORD   700
050132 003654                                                                           .WORD   SFIERR
050134 012074                                                                           .WORD   SFIMSG
                                        ;
                                        ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
7615 050136 005037 002222               10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
7616 050142 012704 051270                       MOV     #T18PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
7617 050146 004737 010662                       JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
7618 050152                                     FORCERROR       12$                     ;@@FORCE ERROR IF FORCER=1
7619 050166 103407                              BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
7620 050170 010001                              MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7621 050172                                     NEXT.ERRNO
7622 050172                             12$:    ERRDF   ERRNO,T18SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
050172 104455                                                                           TRAP    C$ERDF
050174 001275                                                                           .WORD   701
050176 050645                                                                           .WORD   T18SSR
050200 012106                                                                           .WORD   PKTSSR
7623 050202 005237 002222                       INC     FATFLG                  ;SET FATAL ERROR FLAG
7624 050206                             15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
050206 104406                                                                           TRAP    C$CLP1
                                        ;
                                        ;       If Extended Features Hardware Switch Clear then:
                                        ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
7628 050210 012701 051312               MOV     #T18BFR,R1              ;MESSAGE BUFFER ADDRESS
7629 050214 032761 000200 000012        BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
7630 050222 001026                      BNE     30$                     ;BR IF YES
7631 050224 004737 051136               JSR     PC,T18SMISC             ;SETUP PACKET FOR WRITE MISCELLANEOUS
7632 050230 012704 051340               MOV     #T18PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
7633 050234 010465 000000               MOV     R4,TSJB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7634 050240 004737 016376               JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
7635 050244                                     FORCERROR       22$                     ;@@FORCE ERROR IF FORCER=1
7636 050260 103407                              BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```
 7637 050262  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7638 050264                                 NEXT.ERRNO
 7639 050264                          22$:   ERRDF   ERRNO,T182SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      050264  104455                                                            TRAP    C$ERDF
      050266  001276                                                            .WORD   702
      050270  050702                                                            .WORD   T182SSR
      050272  012106                                                            .WORD   PKTSSR
 7640 050274  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 7641 050300                          30$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      050300  104406                                                            TRAP    C$CLP1
 7642
 7643
 7644                                 ;      Do WRITE CHARACTERISTICS to select reserved unit 7
 7645 050302  005037  002222                 CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 7646 050306  012704  051270                 MOV     #T18PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 7647 050312  004737  010662                 JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
 7648 050316                                 FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
 7649 050332  103407                         BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
 7650 050334  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7651 050336                                 NEXT.ERRNO
 7652 050336                          42$:   ERRDF   ERRNO,T18SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      050336  104455                                                            TRAP    C$ERDF
      050340  001277                                                            .WORD   703
      050342  050645                                                            .WORD   T18SSR
      050344  012106                                                            .WORD   PKTSSR
 7653 050346  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 7654 050352                          50$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      050352  104406                                                            TRAP    C$CLP1
 7655
 7656                                 ;      Clear message buffer
 7657 050354  012701  051312                 MOV     #T18BFR,R1              ;GET MESSAGE BUFFER ADDRESS
 7658 050360  013700  051304                 MOV     T18DATA+4,R0            ;SIZE OF MESSAGE BUFFER IN BYTES
 7659 050364  105021                  60$:   CLRB    (R1)+                   ;CLEAR A BYTE
 7660 050366  005300                         DEC     R0                      ;DONE?
 7661 050370  003375                         BGT     60$                     ;BR IF NO
 7662                                 ;      Do a Write Subsystem READ STATUS
 7663 050372  004737  051116                 JSR     PC,T18SRD               ;SETUP PACKET FOR READ STATUS
 7664 050376  012704  051340                 MOV     #T18PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 7665 050402  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 7666 050406  004737  016376                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 7667 050412                                 FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
 7668 050426  103407                         BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
 7669 050430  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7670 050432                                 NEXT.ERRNO
 7671 050432                          62$:   ERRDF   ERRNO,T183SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      050432  104455                                                            TRAP    C$ERDF
      050434  001300                                                            .WORD   704
      050436  050746                                                            .WORD   T183SSR
      050440  012106                                                            .WORD   PKTSSR
 7672 050442  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 7673 050446                          70$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      050446  104406                                                            TRAP    C$CLP1
 7674
 7675
 7676                                 ;      Set first 8 words of expd message buffer = to recv since not testing
 7677                                 ;      Set unused bits in Read Status expd equal rcvd
 7678 050450  004737  051160                 JSR     PC,T18SETEXP            ;SET SOME EXPD TO RECV
```

M14

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```
7679                                    ;       If any transport interface signals are asserted then Print Error
7680 050454  005000                             CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
7681 050456  012701  051312                     MOV     #T18BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
7682 050462  012702  050556                     MOV     #T18EXP,R2              ;EXPD ADDRESS
7683 050466  012703  000012                     MOV     #10.,R3                 ;NUMBER OF WORDS TO COMPARE
7684 050472  004737  011540                     JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
7685 050476                                     FORCERROR       82$,NOTSSR      ;ASD
7686 050506  103404                            BCS     90$                     ;BR IF YES
7687 050510                                     NEXT.ERRNO
7688 050510                             82$:    ERRHRD  ERRNO,T18CMP,MSGSTAT    ;REPORT ERROR
     050510  104456                                                                    TRAP    C$ERHRD
     050512  001301                                                                    .WORD   705
     050514  051013                                                                    .WORD   T18CMP
     050516  012410                                                                    .WORD   MSGSTAT
7689 050520                             90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     050520  104406                                                                    TRAP    C$CLP1
7690
7691 050522  005737  002222                     TST     FATFLG                  ;ANY FATAL ERRORS ?
7692 050526  001402                            BEQ     160$                    ;BRANCH IF NOT
7693 050530  004737  017242                     JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
7694 050534  004737  016516             160$:   JSR     PC,TSTLOOP              ;DO ITERATIONS?
7695 050540  103002                            BCC     165$                    ;BR IF NO
7696 050542  000137  050116                     JMP     T18LOOP                 ;LOOP UNTIL ITERATIONS DONE
7697 050546                             165$:
7698 050546                                     EXIT    TST
     050546  104432                                                                    TRAP    C$EXIT
     050550  000606                                                                    .WORD   L10065-.
7699
7700
7701                                    ;*
7702                                    ;LOCAL STORAGE FOR THIS TEST
7703                                    ;
7704
7705 050552                             T18MSK:                                 ;MASK OF UNUSED BITS IN READ STATUS BYTES
7706 050552      377                            .BYTE   ^C<000>                 ;BYTE 0 MASK
7707 050553      037                            .BYTE   ^C<340>                 ;BYTE 1
7708 050554      100                            .BYTE   ^C<277>                 ;BYTE 2
7709 050555      000                            .BYTE   0                       ;MAKE IT EVEN
7710
7711 050556                             T18EXP:                                 ;EXPECTED DATA BUFFER
7712 050556  000000                            .WORD   0                       ;MESSAGE TYPE
7713 050560  000000                            .WORD   0                       ;DATA FIELD LENGTH
7714 050562  000000                            .WORD   0                       ;RBPCR
7715 050564  000000                            .WORD   0                       ;XST0
7716 050566  000000                            .WORD   0                       ;XST1
7717 050570  000000                            .WORD   0                       ;XST2
7718 050572  000000                            .WORD   0                       ;XST3
7719 050574  000000                            .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUB)
7720 050576  000000                            .WORD   0                       ;READ STATUS BYTE 1/0
7721 050600  000000                            .WORD   0                       ;READ STATUS BYTE 2
7722
7723 050602      377      020           T18XS:  .BYTE   377,020                 ;READ STATUS BYTE 0/1 EXPECTED BASE
7724 050604  000000                            .WORD   0                       ;READ STATUS BYTE 2 EXPECTED BASE
7725
7726                                    ;*
7727                                    ;LOCAL TEXT MESSAGES FOR TEST
7728                                    ;-
```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```
7729
7730 050606     123     164     141  TST18ID:        .ASCIZ  'Static Transport Bus Interface'
7731 050645     127     122     111  T18SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
7732 050702     127     122     111  T182SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
7733 050746     127     122     111  T183SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
7734 051013     124     162     141  T18CMP: .ASCIZ  'Transport Bus Interface Signals NOT Negated After Unit 7 Selected'
7735                                                 .EVEN
7736
7737                                 ;*
7738                                 ; SETUP T18PK2 PACKET FOR READ STATUS
7739                                 ;-
7740 051116                         T18SRD:
7741 051116                                 SAVREG                          ;SAVE R1 R5 UNTIL NEXT RETURN
7742 051122  012700  051350                MOV     #T18DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
7743 051126  112720  000005                MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
7744 051132  105010                        CLRB    (R0)                    ;CLEAR BSEL1
7745 051134  000207                        RTS     PC                      ;RETURN
7746
7747                                 ;*
7748                                 ; SETUP T18PK2 PACKET FOR WRITE MISC.
7749                                 ;-
7750 051136                         T18SMISC:
7751 051136                                 SAVREG                          ;SAVE R1 R5 UNTIL NEXT RETURN
7752 051142  012700  051350                MOV     #T18DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
7753 051146  112720  000010                MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
7754 051152  112710  000200                MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
7755 051156  000207                        RTS     PC                      ;RETURN
7756
7757                                 ;*
7758                                 ;Set first 8 words of expd message buffer = to recv since not testing
7759                                 ;       Set unused bits in Read Status expd equal rcvd
7760                                 ;
7761 051160                         T18SETEXP:
7762 051160  012702  050556                MOV     #T18EXP,R2              ;GET EXPD
7763 051164  012703  051312                MOV     #T18BFR,R3              ;GET READ STATUS RECV BUFFER
7764 051170  012700  000010                MOV     #8.,R0                  ;SET FIRST 8 WORDS EXP=RECV
7765 051174  012322                5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
7766 051176  005300                        DEC     R0                      ;DONE FIRST 8 WORDS?
7767 051200  003375                        BGT     5$                      ;BR IF NO
7768 051202  012701  050552                MOV     #T18MSK,R1              ;GET UNUSED BIT MASK
7769 051206  013712  050602                MOV     T18XS,(R2)              ;SETUP BASE EXPECTED BYTE 1/0
7770 051212  013762  050604  000002        MOV     T18XS+2,2(R2)           ;SETUP BASE EXPECTED BYTE 2
7771 051220  011300                        MOV     (R3),R0                 ;GET RECV BYTE 1 AND BYTE 0
7772 051222  041100                        BIC     (R1),R0                 ;CLEAR ALL BUT UNUSED
7773 051224  040012                        BIC     R0,(R2)                 ;CLEAR UNUSED IN EXP
7774 051226  050012                        BIS     R0,(R2)                 ;SET UNUSED EXPD=RECV FOR COMPARE
7775 051230  016300  000002                MOV     2(R3),R0                ;GET RECV BYTE 2
7776 051234  046100  000002                BIC     2(R1),R0                ;CLEAR ALL BUT UNUSED
7777 051240  040062  000002                BIC     R0,2(R2)                ;CLEAR UNUSED IN EXPD
7778 051244  050062  000002                BIS     R0,2(R2)                ;SET UNUSED EXPD=RECV FOR COMPARE
7779 051250  105062  000003                CLRB    3(R2)                   ;CLEAR EXPD BYTE 3 (UNUSED)
7780 051254  105063  000003                CLRB    3(R3)                   ;CLEAR RECV BYTE 3 (UNUSED)
7781 051260  000207                        RTS     PC                      ;RETURN
7782
7784         051270                        .=<.+10>&177770
7786
7787                                 ;WRITE CHARARTERISTICS COMMAND PACKET
```

```
TEST  7: STATIC TRANSPORT BUS INTERFACE TEST

                                        ;
                                T18PACKET:                              ;COMMAND PACKET FOR TEST
                                        .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
                                        .WORD   T18DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
                                        .WORD   0
                                        .WORD   10.                     ;MESSAGE PACKET MINIMUM SIZE

                                T18DATA:                                ;CHARACTERISTICS DATA BLOCK
                                        .WORD   T18BFR                  ;ADDRESS OF MESSAGE BUFFER
                                        .WORD   0
                                        .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
                                        .WORD   0                       ;ESS,ENB,EAI,ERI
                                        .WORD   7                       ;SELECT RESERVED UNIT 7


                                T18BFR:                                 ;MESSAGE BUFFER
                                        .WORD   0                       ;MESSAGE TYPE
                                        .WORD   0                       ;DATA FIELD LENGTH
                                        .WORD   0                       ;RBPCR
                                        .WORD   0                       ;XST0
                                        .WORD   0                       ;XST1
                                        .WORD   0                       ;XST2
                                        .WORD   0                       ;XST3
                                        .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
                                        .WORD   0                       ;READ STATUS BYTE 1/O RETURNED
                                        .WORD   0                       ;READ STATUS BYTE 2
                                ;
                                ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
                                ;
                                        .=<.+10>&177770
                                T18PK2:
                                        .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
                                        .WORD   T18DT2                  ;LOW ADDRESS OF DATA BLOCK
                                        .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
                                        .WORD   8.                      ;BUFFER EXTENT

                                T18DT2:                                 ;DATA BLOCK
                                        .BYTE   0                       ;BSEL0
                                        .BYTE   0                       ;BSEL1
                                        .WORD   0                       ;SEL2
                                        .WORD   0                       ;DATA


                                        ENDTST
                                                                                L10065:
                                                                                        TRAP    C$ETST
                                        .SBTTL  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 STEPS:
                                ;
```

```
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
                                        ; -

                                                BGNTST
                                                                                        T8::
                                                MOV     #TST19ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
                                                JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
                                                MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
                                        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 NPR 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 loopback signals group 1.
                                        ;       Do Write Subsystem Write Format to Drive loopback 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
                                        ;-
051376                                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
051376                                                                                  T8.1:
051376  104402                                                                            TRAP    C$BSUB

                                        ;       Write to TSSR register to soft initialize the controller
051400                                  5$:
051400  004737  016034                          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
051404  103405                                  BCS     10$                     ;BR IF SOFT INIT OKAY
051406  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
051410                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
051410  104455                                                                            TRAP    C$ERDF
051412  001440                                                                          .WORD   800
051414  003654                                                                          .WORD   SFIERR
051416  012074                                                                          .WORD   SFIMSG

                                        ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
051420  005037  002222                  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
051424  012704  061720                          MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
051430  004737  010662                          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
051434                                          FORCERROR       12$                     ;a#DFORCE ERROR IF FORCER=1
051450  103407                                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
051452  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
051454                                          NEXT.ERRNO
051454                                  12$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
051454  104455                                                                            TRAP    C$ERDF
051456  001441                                                                          .WORD   801
051460  057633                                                                          .WORD   T19SSR
051462  012106                                                                          .WORD   PKTSSR
051464  005237  002222                          INC     FATFLG                  ;SET FATAL ERROR FLAG
051470                                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
051470  104406                                                                          TRAP    C$CLP1

                                        ;       If Extended Features Hardware Switch Clear then:
                                        ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
051472  012701  061742                          MOV     #T19BFR,R1              ;MESSAGE BUFFER ADDRESS
051476  032761  000200  000012                BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
051504  001026                                  BNE     30$                     ;BR IF YES
051506  004737  061572                          JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
051512  012704  062070                          MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
051516  010465  000000                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
051522  004737  016376                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
051526                                          FORCERROR       22$                     ;a#DFORCE ERROR IF FORCER=1
051542  103407                                  BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
051544  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
051546                                          NEXT.ERRNO
051546                                  22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
051546  104455                                                                            TRAP    C$ERDF
051550  001442                                                                          .WORD   802
051552  057670                                                                          .WORD   T192SSR
051554  012106                                                                          .WORD   PKTSSR
051556  005237  002222                          INC     FATFLG                  ;SET FATAL ERROR FLAG
051562                                  30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
      051562  104406                                                          TRAP    C$CLP1
 7947
 7948 051564  005037  002222          ;       Do WRITE CHARACTERISTICS to select reserved unit 7
 7949 051570  012704  061720                  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 7950 051574  004737  010662                  MOV     @T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 7951 051600                                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
 7952 051614  103407                          FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
 7953 051616  010001                          BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
 7954 051620                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7955 051620                                  NEXT.ERRNO
      051620  104455                 42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      051622  001443                                                          TRAP    C$ERDF
      051624  057633                                                          .WORD   803
      051626  012106                                                          .WORD   T19SSR
 7956 051630  005237  002222                                                  .WORD   PKTSSR
 7957 051634                          INC     FATFLG                  ;SET FATAL ERROR FLAG
      051634  104406                 50$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
 7958                                                                         TRAP    C$CLP1
 7959 051636  012700  000100          ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
 7960 051642  052700  000040                  MOV     @NP.OUT,R0              ;SET TAPE DIRECTION OUT
 7961 051646  004737  061432                  BIS     @NP.LOOP,R0             ;SET LOOPBACK ENABLE
 7962 051652  012704  062070                  JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
 7963 051656  010465  000000                  MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 7964 051662  004737  016376                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 7965 051666                                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 7966 051702  103407                          FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
 7967 051704  010001                          BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
 7968 051706                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7969 051706                                  NEXT.ERRNO
      051706  104455                 62$:    ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      051710  001444                                                          TRAP    C$ERDF
      051712  060001                                                          .WORD   804
      051714  012106                                                          .WORD   T194SSR
 7970 051716  005237  002222                                                  .WORD   PKTSSR
 7971 051722                          INC     FATFLG                  ;SET FATAL ERROR FLAG
      051722  104406                 70$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
 7972                                                                         TRAP    C$CLP1
 7973                                 ;       Do Write Subsystem Write Control to CLEAR loopback signals group 1.
 7974                                 ;       Do Write Subsystem Write Format to CLEAR loopback signals group 2.
 7975                                 ;        (the loopback signals have to be cleared here due to the flip-flops
 7976 051724  005000                 ;          that are set on a 1 to 0 transition (IHER,IFMK,ICER))
 7977 051726  042700  000200                  CLR     R0                      ;WRITE 0'S
 7978 051732  042700  000100                  BIC     @WC.IFAD,R0             ;IFAD MUST  ALWAYS =0
 7979 051736  042700  000040                  BIC     @WC.IOTAD,R0            ;ITADO MUST ALWAYS =0
 7980 051742  004737  061532                  BIC     @WC.I1TAD,R0            ;ITAD1 MUST ALWAYS =0
 7981 051746  012704  062070                  JSR     PC,T19WCTL              ;SETUP PACKET FOR WRITE CONTROL
 7982 051752  010465  000000                  MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 7983 051756  004737  016376                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 7984 051762                                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 7985 051776  103407                          FORCERROR       82$                     ;@@DFORCE ERROR IF FORCER=1
 7986 052000  010001                          BCS     90$                     ;BR IF CARRY SET (GOOD RETURN)
 7987 052002                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7988 052002                                  NEXT.ERRNO
      052002  104455                 82$:    ERRDF   ERRNO,T197SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      052004  001445                                                          TRAP    C$ERDF
      052006  060153                                                          .WORD   805
      052010  012106                                                          .WORD   T197SSR
                                                                              .WORD   PKTSSR
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
 7989 052012  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 7990 052016                        90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      052016  104406                                                                TRAP    C$CLP1
 7991 052020  005000                        CLR     R0                      ;SET FORMAT DRIVE DATA=0
 7992 052022  004737  061552                JSR     PC,T19WFMT              ;SETUP PACKET FOR WRITE FORMAT
 7993 052026  012704  062070                MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 7994 052032  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 7995 052036  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 7996 052042                                FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
 7997 052056  103407                        BCS     110$                    ;BR IF CARRY SET (GOOD RETURN)
 7998 052060  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7999 052062                                NEXT.ERRNO
 8000 052062                        102$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      052062  104455                                                                TRAP    C$ERDF
      052064  001446                                                                .WORD   806
      052066  060222                                                                .WORD   T198SSR
      052070  012106                                                                .WORD   PKTSSR
 8001 052072  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8002 052076                        110$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      052076  104406                                                                TRAP    C$CLP1
 8003                               ;       Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
 8004 052100  004737  061410                JSR     PC,T19RSFIF             ;SETUP PKT FOR WRITE MISC Reset Tape Status F FLOPS
 8005 052104  012704  062070                MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8006 052110  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8007 052114  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8008 052120                                FORCERROR       122$                    ;@@DFORCE ERROR IF FORCER=1
 8009 052134  103407                        BCS     130$                    ;BR IF CARRY SET (GOOD RETURN)
 8010 052136  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8011 052140                                NEXT.ERRNO
 8012 052140                        122$:   ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      052140  104455                                                                TRAP    C$ERDF
      052142  001447                                                                .WORD   807
      052144  057670                                                                .WORD   T192SSR
      052146  012106                                                                .WORD   PKTSSR
 8013 052150  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8014 052154                        130$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      052154  104406                                                                TRAP    C$CLP1
 8015                               ;       Do a Write Subsystem READ STATUS
 8016                               ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
 8017                               ;         signals NOT=0 Then Print Error
 8018 052156  004737  061370                JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
 8019 052162  012704  062070                MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8020 052166  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8021 052172  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8022 052176                                FORCERROR       132$                    ;@@DFORCE ERROR IF FORCER=1
 8023 052212  103407                        BCS     140$                    ;BR IF CARRY SET (GOOD RETURN)
 8024 052214  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8025 052216                                NEXT.ERRNO
 8026 052216                        132$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      052216  104455                                                                TRAP    C$ERDF
      052220  001450                                                                .WORD   808
      052222  057734                                                                .WORD   T193SSR
      052224  012106                                                                .WORD   PKTSSR
 8027 052226  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8028 052232                        140$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      052232  104406                                                                TRAP    C$CLP1
 8029 052234  004737  061630                JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
8030 052240  012701  057472                    MOV     @T19EXSTA,R1            ;GET EXPECTED READ STATUS
8031 052244  012702  061762                    MOV     @T19BFSTA,R2            ;GET RECV READ STATUS
8032 052250  011211                            MOV     (R2),(R1)               ;SET EXPD WORD @8 = RECV TEMP
8033 052252  042711  002000                    BIC     @S1.ICER,(R1)           ;SET EXPD ICER  =0
8034 052256  042711  001000                    BIC     @S1.IFMK,(R1)           ;SET EXPD IFMK  =0
8035 052262  042711  000400                    BIC     @S1.IHER,(R1)           ;SET EXPD IHER  =0
8036 052266  016261  000002  000002            MOV     2(R2),2(R1)             ;SET EXPD WORD @9 = RECV (NOT TESTING)
8037 052274  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
8038 052276  012701  061742                    MOV     @T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
8039 052302  012702  057452                    MOV     @T19EXP,R2              ;EXPD ADDRESS
8040 052306  012703  000024                    MOV     @20.,R3                 ;NUMBER OF BYTES TO COMPARE
8041 052312  004737  011540                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
8042 052316                                    FORCERROR       152$,NOTSSR     ;a@D
8043 052326  103404                            BCS     160$                    ;BR IF YES
8044 052330                                    NEXT.ERRNO
8045 052330                            152$:   ERRHRD  ERRNO,T197CMP,MSGLOOP   ;REPORT ERROR
     052330  104456                                                                    TRAP    C$ERHRD
     052332  001451                                                                    .WORD   809
     052334  060673                                                                    .WORD   T197CMP
     052336  013124                                                                    .WORD   MSGLOOP
8046 052340                            160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     052340  104406                                                                    TRAP    C$CLP1
8047                                   ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
8048 052342  005037  057404                    CLR     T19PREV                 ;INIT 1-0 TRANSITION FLAG
8049 052346  012703  002754                    MOV     @TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
8050 052352  012300                    200$:   MOV     (R3)+,R0                ;GET A TEST PATTERN
8051 052354  010337  002320                    MOV     R3,TSTPTR               ;SAVE POINTER INTO TSTBLK
8052 052360  042700  000200                    BIC     @WC.IFAD,R0             ;IFAD MUST  ALWAYS =0
8053 052364  042700  000100                    BIC     @WC.IOTAD,R0            ;ITADO MUST ALWAYS =0
8054 052370  042700  000040                    BIC     @WC.I1TAD,R0            ;ITAD1 MUST ALWAYS =0
8055 052374  010037  002314                    MOV     R0,DATA                 ;SET DATA PATTERN
8056                                   ;       Do Write Subsystem Write Control to Drive loopback signals group 1.
8057                                   ;a@D    CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
8058 052400  013700  002314                    MOV     DATA,R0                 ;GET TEST PATTERN
8059 052404  004737  061654                    JSR     PC,T19CNVT              ;CONVERT PATTERN TO CONTROL DRIVE MASK
8060                                                                           ;R0 CONTAINS WRITE CONTROL DATA HERE
8061 052410  004737  061532                    JSR     PC,T19WCTL              ;SETUP PACKET FOR WRITE CONTROL
8062 052414  012704  062070                    MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8063 052420  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8064 052424  004737  016376                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8065 052430                                    FORCERROR       212$                    ;a@DFORCE ERROR IF FORCER=1
8066 052444  103407                            BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
8067 052446  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8068 052450                                    NEXT.ERRNO
8069 052450                            212$:   ERRDF   ERRNO,T197SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     052450  104455                                                                    TRAP    C$ERDF
     052452  001452                                                                    .WORD   810
     052454  060153                                                                    .WORD   T197SSR
     052456  012106                                                                    .WORD   PKTSSR
8070 052460  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
8071 052464                            220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     052464  104406                                                                    TRAP    C$CLP1
8072
8073                                   ;       Do Write Subsystem Write Format to Drive loopback signals group 2.
8074                                   ;a@D    CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
8075 052466  013700  002314                    MOV     DATA,R0                 ;GET TEST PATTERN
8076 052472  004737  061654                    JSR     PC,T19CNVT              ;CONVERT PATTERN TO FORMAT DRIVE MASK
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
 8077 052476  000300                        SWAB    R0                      ;WRITE FORMAT DATA RETURNED IN HIGH BYTE
 8078 052500  004737  061552                JSR     PC,T19WFMT              ;SETUP PACKET FOR WRITE FORMAT
 8079 052504  012704  062070                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8080 052510  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8081 052514  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8082 052520                                FORCERROR       232$                    ;@@FORCE ERROR IF FORCER=1
 8083 052534  103407                        BCS     240$                    ;BR IF CARRY SET (GOOD RETURN)
 8084 052536  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8085 052540                                NEXT.ERRNO
 8086 052540                        232$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      052540  104455                                                                TRAP    C$ERDF
      052542  001453                                                                .WORD   811
      052544  060222                                                                .WORD   T198SSR
      052546  012106                                                                .WORD   PKTSSR
 8087 052550  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8088 052554                        240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      052554  104406                                                                TRAP    C$CLP1
 8089                               ;       Do a Write Subsystem READ STATUS
 8090 052556  004737  061370                JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
 8091 052562  012704  062070                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8092 052566  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8093 052572  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8094 052576                                FORCERROR       252$                    ;@@FORCE ERROR IF FORCER=1
 8095 052612  103407                        BCS     260$                    ;BR IF CARRY SET (GOOD RETURN)
 8096 052614  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8097 052616                                NEXT.ERRNO
 8098 052616                        252$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      052616  104455                                                                TRAP    C$ERDF
      052620  001454                                                                .WORD   812
      052622  057734                                                                .WORD   T193SSR
      052624  012106                                                                .WORD   PKTSSR
 8099 052626  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8100 052632                        260$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      052632  104406                                                                TRAP    C$CLP1
 8101                               ;       If loopback data NOT= data sent Then Print Error
 8102 052634  004737  061630                JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
 8103 052640  012701  057472                MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
 8104 052644  012702  061762                MOV     #T198FSTA,R2            ;GET RECV READ STATUS
 8105 052650  013711  002314                MOV     DATA,(R1)               ;SET EXPD WORD #8 TO TEST DATA FIRST
 8106 052654  013700  057404                MOV     T19PREV,R0              ;GET PREVIOUS DATA PATTERN
 8107 052660  013703  002314                MOV     DATA,R3                 ;GET CURRENT PATTERN
 8108 052664  012704  000400                MOV     #S1.IHER,R4             ;SETUP IHER EXPECTED
 8109 052670  040411                        BIC     R4,(R1)                 ;SET EXPD IHER  =0
 8110 052672  030400                        BIT     R4,R0                   ;PREVIOUS =1?
 8111 052674  001403                        BEQ     275$                    ;BR IF NO
 8112 052676  030403                        BIT     R4,R3                   ;CURRENT  =0?
 8113 052700  001001                        BNE     275$                    ;BR IF NO
 8114 052702  050411                        BIS     R4,(R1)                 ;SET EXPD IHER  =1
 8115 052704  012704  001000        275$:   MOV     #S1.IFMK,R4             ;SETUP IFMK EXPECTED
 8116 052710  040411                        BIC     R4,(R1)                 ;SET EXPD IFMK  =0
 8117 052712  030400                        BIT     R4,R0                   ;PREVIOUS =1?
 8118 052714  001403                        BEQ     280$                    ;BR IF NO
 8119 052716  030403                        BIT     R4,R3                   ;CURRENT  =0?
 8120 052720  001001                        BNE     280$                    ;BR IF NO
 8121 052722  050411                        BIS     R4,(R1)                 ;SET EXPD IFMK  =1
 8122 052724  012704  002000        280$:   MOV     #S1.ICER,R4             ;SETUP ICER EXPECTED
 8123 052730  040411                        BIC     R4,(R1)                 ;SET EXPD ICER  =0
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
 8124 052732  030400                        BIT     R4,R0                   ;PREVIOUS =1?
 8125 052734  001403                        BEQ     285$                    ;BR IF NO
 8126 052736  030403                        BIT     R4,R3                   ;CURRENT  =0?
 8127 052740  001001                        BNE     285$                    ;BR IF NO
 8128 052742  050411                        BIS     R4,(R1)                 ;SET EXPD ICER  =1
 8129 052744  011100                285$:   MOV     (R1),R0                 ;GET EXPD WORD
 8130                               ;       If previous IIDENT=1 and current is IIDENT=1 then EXPD= 0 else 1
 8131 052746  012704  004000                MOV     @S1.IIDENT,R4           ;IIDENT
 8132 052752  050400                        BIS     R4,R0                   ;ASSUME EXPD=1
 8133 052754  030437  057404                BIT     R4,T19PREV              ;PREVIOUS IIDENT=1?
 8134 052760  001403                        BEQ     288$                    ;BR IF NO
 8135 052762  030403                        BIT     R4,R3                   ;IS CURRENT IIDENT=1?
 8136 052764  001401                        BEQ     288$                    ;BR IF NO
 8137 052766  040400                        BIC     R4,R0                   ;SET EXPD=0
 8138 052770  052700  040000          288$: BIS     @S1.I2RES,R0            ;IRESV2 EXPD ALWAYS=1
 8139 052774  052700  020000                BIS     @S1.I1RES,R0            ;IRESV1 EXPD ALWAYS=1
 8140 053000  042700  100000                BIC     @S1.PARERR,R0           ;IGNORE PARERR
 8141 053004  032712  100000                BIT     @S1.PARERR,(R2)         ;IS PARERR SET IN RECV?
 8142 053010  001402                        BEQ     290$                    ;BR IF NO
 8143 053012  052700  100000                BIS     @S1.PARERR,R0           ;SET IN EXPD
 8144 053016  010011                290$:   MOV     R0,(R1)                 ;SETUP FINAL EXPD IN WORD @8
 8145 053020  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD @9 = RECV (NOT TESTING)
 8146 053026  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 8147 053030  012701  061742                MOV     @T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 8148 053034  012702  057452                MOV     @T19EXP,R2              ;EXPD ADDRESS
 8149 053040  012703  000024                MOV     @20.,R3                 ;NUMBER OF BYTES TO COMPARE
 8150 053044  004737  011540                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 8151 053050                                FORCERROR       302$,NOTSSR     ;@@D
 8152 053060  103404                        BCS     310$                    ;BR IF YES
 8153 053062                                NEXT.ERRNO
 8154 053062                        302$:   ERRHRD  ERRNO,T198CMP,MSGLOOP   ;REPORT ERROR
      053062  104456                                                                TRAP    C$ERHRD
      053064  001455                                                                .WORD   813
      053066  060761                                                                .WORD   T198CMP
      053070  013124                                                                .WORD   MSGLOOP
 8155 053072                        310$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      053072  104406                                                                TRAP    C$CLP1
 8156                               ;       Do a Write Subsystem Write Misc to Reset Tape Status F FLOPS
 8157 053074  004737  061410                JSR     PC,T19RSFIF             ;SETUP PKT FOR WRITE MISC Reset STATUS
 8158 053100  012704  062070                MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8159 053104  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8160 053110  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8161 053114                                FORCERROR       322$                    ;@@DFORCE ERROR IF FORCER=1
 8162 053130  103407                        BCS     330$                    ;BR IF CARRY SET (GOOD RETURN)
 8163 053132  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8164 053134                                NEXT.ERRNO
 8165 053134                        322$:   ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      053134  104455                                                                TRAP    C$ERDF
      053136  001456                                                                .WORD   814
      053140  057670                                                                .WORD   T192SSR
      053142  012106                                                                .WORD   PKTSSR
 8166 053144  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8167 053150                        330$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      053150  104406                                                                TRAP    C$CLP1
 8168                               ;       Do a Write Subsystem READ STATUS
 8169 053152  004737  061370                JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
 8170 053156  012704  062070                MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
```

J15

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
8171 053162  010465  000000                    MOV     R4,T5DB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8172 053166  004737  016376                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8173 053172                                    FORCERROR       342$            ;@@OFORCE ERROR IF FORCER=1
8174 053206  103407                            BCS     350$                    ;BR IF CARRY SET (GOOD RETURN)
8175 053210  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8176 053212                                    NEXT.ERRNO
8177 053212                            342$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053212  104455                                                                    TRAP    C$ERDF
     053214  001457                                                                    .WORD   815
     053216  057734                                                                    .WORD   T193SSR
     053220  012106                                                                    .WORD   PKTSSR
8178 053222  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
8179 053226                            350$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053226  104406                                                                    TRAP    C$CLP1
8180 053230  004737  061630                    JSR     PC,T19SETEXP            ;SET WORDS 0 7 EXPD=RECV (NOT TESTING)
8181 053234  012701  057472                    MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
8182 053240  012702  061762                    MOV     #T198FSTA,R2            ;GET RECV READ STATUS
8183 053244  011211                            MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
8184 053246  042711  002000                    BIC     #S1.ICER,(R1)           ;SET EXPD ICER  =0
8185 053252  042711  001000                    BIC     #S1.IFMK,(R1)           ;SET EXPD IFMK  =0
8186 053256  042711  000400                    BIC     #S1.IHER,(R1)           ;SET EXPD IHER  =0
8187 053262  016261  000002  000002            MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
8188 053270  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
8189 053272  012701  061742                    MOV     #T198FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
8190 053276  012702  057452                    MOV     #T19EXP,R2              ;EXPD ADDRESS
6.91 053302  012703  000024                    MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
8192 053306  004737  011540                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
8193 053312                                    FORCERROR       362$,NOTSSR     ;@@D
8194 053322  103404                            BCS     370$                    ;BR IF YES
8195 053324                                    NEXT.ERRNO
8196 053324                            362$:   ERRHRD  ERRNO,T197CMP,MSGSTAT   ;REPORT ERROR
     053324  104456                                                                    TRAP    C$ERHRD
     053326  001460                                                                    .WORD   816
     053330  060673                                                                    .WORD   T197CMP
     053332  012410                                                                    .WORD   MSGSTAT
8197 053334                            370$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053334  104406                                                                    TRAP    C$CLP1
8198
8199 053336  013737  002314  057404            MOV     DATA,T19PREV            ;SETUP PREVIOUS DATA FOR EXPD CALC.
8200 053344  013703  002320                    MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
8201 053350  020327  003064                    CMP     R3,#TBLEND              ;END OF TSTBLK?
8202 053354  103002                            BHIS    400$                    ;BR IF YES
8203 053356  000137  052352                    JMP     200$                    ;DO NEXT TSTBLK PATTERN
8204 053362                            400$.
8205
8206 053362                                    ENDSUB                          ;////////////// END SUBTEST //////////
     053362                                                                        L10067:
     053362  104403                                                                    TRAP    C$ESUB
8207
8208 053364  005737  002222                    TST     FATFLG                  ;ANY FATAL ERRORS ?
8209 053370  001402                            BEQ     460$                    ;BRANCH IF NOT
8210 053372  004737  017242                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
8211 053376                            460$:
8212
8213
8214
8215
```

K15


```
                                            .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
                                    ;
 053376                                     BGNSUB                          ;///////////// BEGIN SUBTEST // ., . .
 053376                                                                                   T8.2:
 053376  104402                                                                                     TRAP    C$BSUB
                                    ;       Write to TSSR register to soft initialize the controller
 053400                             5$:
 053400  004737  016034                     JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
 053404  103405                             BCS     10$                     ;BR IF SOFT INIT OKAY
 053406  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 053410                                     ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
 053410  104455                                                                                     TRAP    C$ERDF
 053412  001460                                                                                     .WORD   816
 053414  003654                                                                                     .WORD   SFIERR
 053416  012074                                                                                     .WORD   SFIMSG
                                    ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
 053420  005037  002222             10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 053424  012704  061720                     MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 053430  004737  010662                     JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
 053434                                     FORCERROR       12$                 ;@@FORCE ERROR IF FORCER=1
 053450  103407                             BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 053452  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

L15

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

```
 8267 053454                                    NEXT.ERRNO
 8268 053454                            12$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      053454  104455                                                              TRAP    C$ERDF
      053456  001461                                                              .WORD   819
      053460  057633                                                              .WORD   T19SSR
      053462  012106                                                              .WORD   PKTSSR
 8269 053464  005237  002222                       INC     FATFLG                ;SET FATAL ERROR FLAG
 8270 053470                            15$:    CKLOOP                          ;LOOP ON ERROR. IF FLAG SET
      053470  104406                                                              TRAP    C$CLP1
 8271                                   ;       If Extended Features Hardware Switch Clear then
 8272                                   ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
 8273 053472  012701  061742                       MOV     @T19BFR,R1            ;MESSAGE BUFFER ADDRESS
 8274 053476  032761  000200  000012               BIT     @X2.EXTF,XST2(R1)     ;EXTENDED FEATURES SWITCH SET?
 8275 053504  001026                               BNE     30$                   ;BR IF YES
 8276 053506  004737  061572                       JSR     PC,T19SEXT            ;SETUP PACKET FOR WRITE MISC INVERT
 8277 053512  012704  062070                       MOV     @T19PK2,R4            ;GET WRITE SUBSYSTEM COMMAND PACKET
 8278 053516  010465  000000                       MOV     R4,TSDB(R5)           ;SET THE PACKET ADDRESS TO EXECUTE
 8279 053522  004737  016376                       JSR     PC,CHKTSSR            ;WAIT FOR SSR TO SET
 8280 053526                                       FORCERROR       22$           ;FORCE ERROR IF FORCER=1
 8281 053542  103407                               BCS     30$                   ;BR IF CARRY SET (GOOD RETURN)
 8282 053544  010001                               MOV     R0,R1                 ;SAVE CONTENTS OF TSSR
 8283 053546                                       NEXT.ERRNO
 8284 053546                            22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      053546  104455                                                              TRAP    C$ERDF
      053550  001462                                                              .WORD   818
      053552  057670                                                              .WORD   T192SSR
      053554  012106                                                              .WORD   PKTSSR
 8285 053556  005237  002222                       INC     FATFLG                ;SET FATAL ERROR FLAG
 8286 053562                            30$:    CKLOOP                          ;LOOP ON ERROR. IF FLAG SET
      053562  104406                                                              TRAP    C$CLP1
 8287                                   ;       Do WRITE CHARACTERISTICS to select reserved unit 7
 8288 053564  012704  061720                       MOV     @T19PACKET,R4         ;GET THE ADDRESS OF COMMAND PACKET
 8289 053570  004737  010662                       JSR     PC,WRTCHR             ;DO WRITE CHARACTERISTICS COMMAND
 8290 053574                                       FORCERROR       42$           ;FORCE ERROR IF FORCER=1
 8291 053610  103407                               BCS     50$                   ;BR IF CARRY SET (GOOD RETURN)
 8292 053612  010001                               MOV     R0,R1                 ;SAVE CONTENTS OF TSSR
 8293 053614                                       NEXT.ERRNO
 8294 053614                            42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      053614  104455                                                              TRAP    C$ERDF
      053616  001463                                                              .WORD   819
      053620  057633                                                              .WORD   T19SSR
      053622  012106                                                              .WORD   PKTSSR
 8295 053624  005237  002222                       INC     FATFLG                ;SET FATAL ERROR FLAG
 8296 053630                            50$:    CKLOOP                          ;LOOP ON ERROR. IF FLAG SET
      053630  104406                                                              TRAP    C$CLP1
 8297
 8298
 8299                                   ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
 8300 053632  012703  002754                       MOV     @TSTBLK,R3            ;SET FIRST PATTERN ADDRESS
 8301 053636  012337  002314            100$:    MOV     (R3)+,DATA            ;GET A WORD PATTERN
 8302 053642  042737  177400  002314               BIC     @^C<377>,DATA         ;DATA IS BYTE
 8303 053650  010337  002320                       MOV     R3,TSTPTR             ;SETUP CURRENT TSTBLK POINTER
 8304                                   ;       Do a WRITE NPR to set loopback and turn direction OUT
 8305 053654  012700  000100                       MOV     @NP.OUT,R0            ;SET TURN DIRECTION OUT
 8306 053660  052700  000040                       BIS     @NP.LOOP,R0           ;SET LOOPBACK
 8307 053664  004737  061432                       JSR     PC,T19SNPR            ;SETUP T19PK2 FOR WRITE NPR
 8308 053670  012704  062070                       MOV     @T19PK2,R4            ;GET WRITE SUBSYSTEM COMMAND PACKET
```

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

```
 8309 053674  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8310 053700  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8311 053704                                FORCERROR       102$                    ;aaDFORCE ERROR IF FORCER=1
 8312 053720  103407                        BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
 8313 053722  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8314 053724                                NEXT.ERRNO
 8315 053724                        102$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      05*724  104455                                                                    TRAP    C$ERDF
      053726  001464                                                                   .WORD   820
      053730  060001                                                                   .WORD   T194SSR
      053732  012106                                                                   .WORD   PKTSSR
 8316 053734  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8317 053740                        105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      053740  104406                                                                   TRAP    C$CLP1
 8318                               ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
 8319 053742  012700  000001                MOV     #1,R0                   ;WRITE 1 BYTE
 8320 053746  012701  002314                MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
 8321 053752  004737  061476                JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
 8322 053756  012704  062070                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8323 053762  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8324 053766  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8325 053772                                FORCERROR       107$                    ;aaDFORCE ERROR IF FORCER=1
 8326 054006  103407                        BCS     110$                    ;BR IF CARRY SET (GOOD RETURN)
 8327 054010  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8328 054012                                NEXT.ERRNO
 8329 054012                        107$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      054012  104455                                                                   TRAP    C$ERDF
      054014  001465                                                                   .WORD   821
      054016  060044                                                                   .WORD   T195SSR
      054020  012106                                                                   .WORD   PKTSSR
 8330 054022  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8331 054026                        110$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      054026  104406                                                                   TRAP    C$CLP1
 8332                               ;       Do a READ FIFO with tape direction OUT to load tape out write latch
 8333 054030  012700  000001                MOV     #1,R0                   ;SET READ BYTE COUNT
 8334 054034  004737  061456                JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
 8335 054040  012704  062070                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8336 054044  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8337 054050  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8338 054054                                FORCERROR       122$                    ;aaDFORCE ERROR IF FORCER=1
 8339 054070  103407                        BCS     130$                    ;BR IF CARRY SET (GOOD RETURN)
 8340 054072  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8341 054074                                NEXT.ERRNO
 8342 054074                        122$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      054074  104455                                                                   TRAP    C$ERDF
      054076  001466                                                                   .WORD   822
      054100  060110                                                                   .WORD   T196SSR
      054102  012106                                                                   .WORD   PKTSSR
 8343 054104  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8344 054110                        130$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      054110  104406                                                                   TRAP    C$CLP1
 8345                               ;       Do a WRITE NPR to set loopback and tape directi  IN
 8346 054112  005000                        CLR     R0                      ;CLR NP.OUT TO SET TAPE DIRECTION IN
 8347 054114  052700  000040                BIS     #NP.LOOP,R0             ;SET LOOPBACK
 8348 054120  004737  061432                JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
 8349 054124  012704  062070                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8350 054130  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
```

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

```
8351 054134  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8352 054140                                  FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
8353 054154  103407                          BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
8354 054156  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8355 054160                                  NEXT.ERRNO
8356 054 60                          142$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054160  104455                                                              TRAP    C$ERDF
     054162  001467                                                              .WORD   823
     054164  060001                                                              .WORD   T194SSR
     054166  012106                                                              .WORD   PKTSSR
8357 054170  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
8358 054174                          150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054174  104406                                                              TRAP    C$CLP1
8359                                 ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
8360 054176  012700  000001                  MOV     #1,R0                   ;WRITE 1 BYTE
8361 054202  012701  002314                  MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
8362 054206  004737  061476                  JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
8363 054212  012704  062070                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8364 054216  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8365 054222  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8366 054226                                  FORCERROR       162$                    ;@@DFORCE ERROR IF FORCER=1
8367 054242  103407                          BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
8368 054244  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8369 054246                                  NEXT.ERRNO
8370 054246                          162$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054246  104455                                                              TRAP    C$ERDF
     054250  001470                                                              .WORD   824
     054252  060044                                                              .WORD   T195SSR
     054254  012106                                                              .WORD   PKTSSR
8371 054256  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
8372 054262                          170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054262  104406                                                              TRAP    C$CLP1
8373                                 ;       Do a READ FIFO with tape direction IN to read data
8374                                 ;       If Data read from FIFO NOT= to Data sent Then Print Error
8375 054264  012700  000001                  MOV     #1,R0                   ;SET READ BYTE COUNT
8376 054270  004737  061456                  JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
8377 054274  012704  062070                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8378 054300  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8379 054304  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8380 054310                                  FORCERROR       182$                    ;@@DFORCE ERROR IF FORCER=1
8381 054324  103407                          BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
8382 054326  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8383 054330                                  NEXT.ERRNO
8384 054330                          182$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054330  104455                                                              TRAP    C$ERDF
     054332  001471                                                              .WORD   825
     054334  060110                                                              .WORD   T196SSR
     054336  012106                                                              .WORD   PKTSSR
8385 054340  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
8386 054344                          190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054344  104406                                                              TRAP    C$CLP1
8387 054346  004737  061630                  JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
8388 054352  012701  057472                  MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
8389 054356  012702  061762                  MOV     #T19RFSTA,R2            ;GET RECV READ STATUS
8390 054362  013711  002314                  MOV     DATA,(R1)               ;SET EXPD WORD #8 = DATA
8391 054366  016261  000002  000002          MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
8392 054374  005000                          CLR     R0                      ;HIGH RECV ADDRESS  JR CKMSG2
```

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

```
 8393 054376  012701  061742                MOV     @T198FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 8394 054402  012702  057452                MOV     @T19EXP,R2              ;EXPD ADDRESS
 8395 054406  012703  000022                MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
 8396 054412  004737  011540                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 8397 054416                                FORCERROR       202$,NOTSSR     ;@@D
 8398 054426  103404                        BCS     210$                    ;BR IF YES
 8399 054430                                NEXT.ERRNO
 8400 054430                        202$:   ERRHRD  ERRNO,T199CMP,MSGSUB    ;REPORT ERROR
      054430  104456                                                                TRAP    C$ERHRD
      054432  001472                                                                .WORD   826
      054434  061050                                                                .WORD   T199CMP
      054436  014002                                                                .WORD   MSGSUB
 8401 054440                        210$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      054440  104406                                                                TRAP    C$CLP1
 8402                               ;       Do a Write Subsystem READ STATUS
 8403                               ;       If Input Ready  NOT=1   Then Print Error
 8404                               ;       If Output Ready NOT=0   Then Print Error
 8405                               ;       If Data In Miss NOT=0   Then Print Error
 8406 054442  004737  061370                JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
 8407 054446  012704  062070                MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8408 054452  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8409 054456  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8410 054462                                FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
 8411 054476  103407                        BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
 8412 054500  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8413 054502                                NEXT.ERRNO
 8414 054502                        212$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      054502  104455                                                                TRAP    C$ERDF
      054504  001473                                                                .WORD   827
      054506  057734                                                                .WORD   T193SSR
      054510  012106                                                                .WORD   PKTSSR
 8415 054512  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8416 054516                        220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      054516  104406                                                                TRAP    C$CLP1
 8417 054520  004737  061630                JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
 8418 054524  012701  057472                MOV     @T19EXSTA,R1            ;GET EXPECTED READ STATUS
 8419 054530  012702  061762                MOV     @T198FSTA,R2            ;GET RECV READ STATUS
 8420 054534  012221                        MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
 8421 054536  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
 8422 054540  052711  000020                BIS     @S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
 8423 054544  042711  000040                BIC     @S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
 8424 054550  042711  000200                BIC     @S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
 8425 054554  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 8426 054556  012701  061742                MOV     @T198FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 8427 054562  012702  057452                MOV     @T19EXP,R2              ;EXPD ADDRESS
 8428 054566  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 8429 054572  004737  011540                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 8430 054576                                FORCERROR       232$,NOTSSR     ;@@D
 8431 054606  103404                        BCS     240$                    ;BR IF YES
 8432 054610                                NEXT.ERRNO
 8433 054610                        232$:   ERRHRD  ERRNO,T196CMP,MSGSTAT   ;REPORT ERROR
      054610  104456                                                                TRAP    C$ERHRD
      054612  001474                                                                .WORD   828
      054614  060610                                                                .WORD   T196CMP
      054616  012410                                                                .WORD   MSGSTAT
 8434 054620                        240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      054620  104406                                                                TRAP    C$CLP1
```

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

```
8435
8436
8437
8438 054622                                    FORCEXIT        255$            ;a@D
8439 054632  013703  002320                    MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
8440 054636  020327  003064                    CMP     R3,@TBLEND              ;END OF TSTBLK?
8441 054642  103002                           BHIS    255$                    ;BR IF YES
8442 054644  000137  053636                    JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
8443 054650                            255$:
8444
8445 054650                                    ENDSUB                          ;////////////// END SUBTEST //////////
     054650                                                                            L10070:
     054650  104403                                                                        TRAP    C$ESUB
8446
8447 054652  005737  002222                    TST     FATFLG                  ;ANY FATAL ERRORS ?
8448 054656  001402                           BEQ     260$                    ;BRANCH IF NOT
8449 054660  004737  017242                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
8450 054664                            260$:
8451
8452
8453                                           .SBTTL  TEST  8: SUBTEST 3: LOOPBACK WRITE STROBE TEST
8454
8455                                   ;+*
8456                                   ; TEST 8: SUBTEST 3:
8457                                   ;
8458                                   ; SUBTEST DESCRIPTION.
8459                                   ;
8460                                   ;       This subtest verifies the Write Strobe loopback path
8461                                   ;       can strobe data from the FIFO to the Data lines.
8462                                   ;       The signal IRESV3 drives IWSTR (write strobe) to write
8463                                   ;       data from the FIFO to the tape data out latch.
8464                                   ;
8465                                   ; TEST STEPS:
8466                                   ;
8467                                   ;
8468                                   ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
8469                                   ;    BEGIN
8470                                   ;       Write to TSSR register to soft initialize the controller
8471                                   ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
8472                                   ;       If Extended Features Hardware Switch Clear then:
8473                                   ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
8474                                   ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
8475                                   ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
8476                                   ;       Do a WRITE NPR to set loopback and tape direction OUT
8477                                   ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
8478                                   ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
8479                                   ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 0 to load write data latch
8480                                   ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
8481                                   ;       Do a WRITE NPR to set loopback and tape direction IN
8482                                   ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
8483                                   ;         to strobe loopback data into FIFO.
8484                                   ;       Do a READ FIFO with tape direction IN to read data
8485                                   ;       If Data read from FIFO NOT= to Data sent Then Print Error
8486                                   ;    END
8487                                   ;--
8488 054664                                    BGNSUB                          ;////////////// BEGIN SUBTEST //////////
     054664                                                                            T8.3:
```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```
      054664  104402                                                          TRAP    C$BSUB
 8489                                ;       Write to TSSR register to soft initialize the controller
 8490 054666                         5$:
 8491 054666  004737  016034                 JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
 8492 054672  103405                         BCS     10$                     ;BR IF SOFT INIT OKAY
 8493 054674  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8494 054676                                 ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      054676  104455                                                          TRAP    C$ERDF
      054700  001474                                                          .WORD   828
      054702  003654                                                          .WORD   SFIERR
      054704  012074                                                          .WORD   SFIMSG
 8495                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
 8496 054706  005037  002222         10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 8497 054712  012704  061720                 MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 8498 054716  004737  010662                 JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
 8499 054722                                 FORCERROR       12$                     ;@@FORCE ERROR IF FORCER=1
 8500 054736  103407                         BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 8501 054740  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8502 054742                                 NEXT.ERRNO
 8503 054742                         12$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      054742  104455                                                          TRAP    C$ERDF
      054744  001475                                                          .WORD   829
      054746  057633                                                          .WORD   T19SSR
      054750  012106                                                          .WORD   PKTSSR
 8504 054752  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 8505 054756                         15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      054756  104406                                                          TRAP    C$CLP1
 8506                                ;       If Extended Features Hardware Switch Clear then:
 8507                                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
 8508 054760  012701  061742                 MOV     #T198FR,R1              ;MESSAGE BUFFER ADDRESS
 8509 054764  032761  000200  000012         BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
 8510 054772  001026                         BNE     30$                     ;BR IF YES
 8511 054774  004737  061572                 JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
 8512 055000  012704  062070                 MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8513 055004  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8514 055010  004737  016376                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8515 055014                                 FORCERROR       22$                     ;@@FORCE ERROR IF FORCER=1
 8516 055030  103407                         BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
 8517 055032  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8518 055034                                 NEXT.ERRNO
 8519 055034                         22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      055034  104455                                                          TRAP    C$ERDF
      055036  001476                                                          .WORD   830
      055040  057670                                                          .WORD   T192SSR
      055042  012106                                                          .WORD   PKTSSR
 8520 055044  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 8521 055050                         30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      055050  104406                                                          TRAP    C$CLP1
 8522                                ;       Do WRITE CHARACTERISTICS to select reserved unit 7
 8523 055052  012704  061720                 MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 8524 055056  004737  010662                 JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
 8525 055062                                 FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
 8526 055076  103407                         BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
 8527 055100  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8528 055102                                 NEXT.ERRNO
 8529 055102                         42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      055102  104455                                                          TRAP    C$ERDF
```

E16

SEQ 0199

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```
        055104  001477                                                          .WORD   831
        055106  057633                                                          .WORD   T19SSR
        055110  012106                                                          .WORD   PKTSSR
  8530  055112  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
  8531  055116                        50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        055116  104406                                                          TRAP    C$CLP1
  8532
  8533                                  ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
  8534  055120  012703  002754                MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
  8535  055124  012337  002314        100$:   MOV     (R3)+,DATA              ;GET A TEST PATTERN
  8536  055130  042737  177400  002314        BIC     #^C<377>,DATA           ;DATA IS BYTE
  8537  055136  010337  002320                MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
  8538                                  ;       Do a WRITE NPR to set loopback and tape direction OUT
  8539  055142  012700  000100                MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
  8540  055146  052700  000040                BIS     #NP.LOOP,R0             ;SET LOOPBACK
  8541  055152  004737  061432                JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
  8542  055156  012704  062070                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
  8543  055162  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
  8544  055166  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
  8545  055172                                FORCERROR       102$                    ;aaDFORCE ERROR IF FORCER=1
  8546  055206  103407                        BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
  8547  055210  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  8548  055212                                NEXT.ERRNO
  8549  055212                        102$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        055212  104455                                                          TRAP    C$ERDF
        055214  001500                                                          .WORD   832
        055216  060001                                                          .WORD   T194SSR
        055220  012106                                                          .WORD   PKTSSR
  8550  055222  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
  8551  055226                        105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        055226  104406                                                          TRAP    C$CLP1
  8552                                  ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
  8553  055230  012700  000002                MOV     #WF.I3RES,R0            ;IRESV3==>IWSTR=1
  8554  055234  004737  061552                JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
  8555  055240  012704  062070                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
  8556  055244  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
  8557  055250  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
  8558  055254                                FORCERROR       112$                    ;aaDFORCE ERROR IF FORCER=1
  8559  055270  103407                        BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
  8560  055272  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  8561  055274                                NEXT.ERRNO
  8562  055274                        112$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        055274  104455                                                          TRAP    C$ERDF
        055276  001501                                                          .WORD   833
        055300  060222                                                          .WORD   T198SSR
        055302  012106                                                          .WORD   PKTSSR
  8563  055304  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
  8564  055310                        120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        055310  104406                                                          TRAP    C$CLP1
  8565                                  ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
  8566  055312  012700  000001                MOV     #1,R0                   ;WRITE 1 BYTE
  8567  055316  012701  002314                MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
  8568  055322  004737  061476                JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
  8569  055326  012704  062070                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
  8570  055332  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
  8571  055336  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
  8572  055342                                FORCERROR       132$                    ;aaDFORCE ERROR IF FORCER=1
```

F16

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```
 8573 055356  103407                        BCS     140$                    ;BR IF CARRY SET (GOOD RETURN)
 8574 055360  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8575 055362                                NEXT.ERRNO
 8576 055362                        132$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      055362  104455                                                                TRAP    C$ERDF
      055364  001502                                                                .WORD   834
      055366  060044                                                                .WORD   T195SSR
      055370  012106                                                                .WORD   PKTSSR
 8577 055372  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8578 055376                        140$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      055376  104406                                                                TRAP    C$CLP1
 8579                               ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 0
 8580 055400  005000                        CLR     R0                      ;SET IRESV3==>IWSTR=0
 8581 055402  004737  061552                JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORM
 8582 055406  012704  062070                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAN   ACKET
 8583 055412  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO E   CUTE
 8584 055416  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8585 055422                                FORCERROR       152$                    ;@@FORCE ERROR IF FORCER=1
 8586 055436  103407                        BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
 8587 055440  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8588 055442                                NEXT.ERRNO
 8589 055442                        152$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      055442  104455                                                                TRAP    C$ERDF
      055444  001503                                                                .WORD   835
      055446  060222                                                                .WORD   T198SSR
      055450  012106                                                                .WORD   PKTSSR
 8590 055452  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8591 055456                        160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      055456  104406                                                                TRAP    C$CLP1
 8592                               ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
 8593 055460  012700  000002                MOV     #WF.I3RES,R0            ;IRESV3==>IWSTR=1
 8594 055464  004737  061552                JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
 8595 055470  012704  062070                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8596 055474  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8597 055500  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8598 055504                                FORCERROR       172$                    ;@@FORCE ERROR IF FORCER=1
 8599 055520  103407                        BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
 8600 055522  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8601 055524                                NEXT.ERRNO
 8602 055524                        172$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      055524  104455                                                                TRAP    C$ERDF
      055526  001504                                                                .WORD   836
      055530  060222                                                                .WORD   T198SSR
      055532  012106                                                                .WORD   PKTSSR
 8603 055534  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8604 055540                        180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      055540  104406                                                                TRAP    C$CLP1
 8605
 8606                               ;       Do a WRITE NPR to set loopback and tape direction IN
 8607 055542  005000                        CLR     R0                      ;CLR NP.OUT TO SET TAPE DIRECTION IN
 8608 055544  052700  000040                BIS     #NP.LOOP,R0             ;SET LOOPBACK
 8609 055550  004737  061432                JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
 8610 055554  012704  062070                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8611 055560  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8612 055564  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8613 055570                                FORCERROR       182$                    ;@@FORCE ERROR IF FORCER=1
 8614 055604  103407                        BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
```

```
TEST  8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

 8615 055606  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8616 055610                                NEXT.ERRNO
 8617 055610                        182$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      055610  104455                                                                TRAP    C$ERDF
      055612  001505                                                                .WORD   837
      055614  060001                                                                .WORD   T194SSR
      055616  012106                                                                .WORD   PKTSSR
 8618 055620  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8619 055624                        190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      055624  104406                                                                TRAP    C$CLP1
 8620                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
 8621 055626  012700  000001                MOV     #1,R0                   ;WRITE 1 BYTE
 8622 055632  012701  002314                MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
 8623 055636  004737  061476                JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
 8624 055642  012704  062070                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8625 055646  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8626 055652  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8627 055656                                FORCERROR       202$                    ;@@DFORCE ERROR IF FORCER=1
 8628 055672  103407                        BCS     210$                    ;BR IF CARRY SET (GOOD RETURN)
 8629 055674  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8630 055676                                NEXT.ERRNO
 8631 055676                        202$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      055676  104455                                                                TRAP    C$ERDF
      055700  001506                                                                .WORD   838
      055702  060044                                                                .WORD   T195SSR
      055704  012106                                                                .WORD   PKTSSR
 8632 055706  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8633 055712                        210$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      055712  104406                                                                TRAP    C$CLP1
 8634                                ;       Do a READ FIFO with tape direction IN to read data
 8635 055714  012700  000001                MOV     #1,R0                   ;SET READ BYTE COUNT
 8636 055720  004737  061456                JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
 8637 055724  012704  062070                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8638 055730  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8639 055734  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8640 055740                                FORCERROR       222$                    ;@@DFORCE ERROR IF FORCER=1
 8641 055754  103407                        BCS     230$                    ;BR IF CARRY SET (GOOD RETURN)
 8642 055756  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8643 055760                                NEXT.ERRNO
 8644 055760                        222$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      055760  104455                                                                TRAP    C$ERDF
      055762  001507                                                                .WORD   839
      055764  060110                                                                .WORD   T196SSR
      055766  012106                                                                .WORD   PKTSSR
 8645 055770  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8646 055774                        230$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      055774  104406                                                                TRAP    C$CLP1
 8647                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
 8648 055776  004737  061630                JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
 8649 056002  012701  057472                MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
 8650 056006  012702  061762                MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
 8651 056012  013711  002314                MOV     DATA,(R1)               ;SET EXPD WORD #8 = DATA
 8652 056016  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
 8653 056024  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 8654 056026  012701  061742                MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 8655 056032  012702  057452                MOV     #T19EXP,R2              ;EXPD ADDRESS
 8656 056036  012703  000022                MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
```

```
TEST  8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

 8657 056042  004737  011540                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 8658 056046                                  FORCERROR       242$,NOTSSR     ;@@D
 8659 056056  103404                          BCS     250$                    ;BR IF YES
 8660 056060                                  NEXT.ERRNO
 8661 056060                          242$:   ERRHRD  ERRNO,T19WSTR,MSGSUB    ;REPORT ERROR
      056060  104456                                                              TRAP    C$ERHRD
      056062  001510                                                              .WORD   840
      056064  061133                                                              .WORD   T19WSTR
      056066  014002                                                              .WORD   MSGSUB
 8662 056070                          250$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      056070  104406                                                              TRAP    C$CLP1
 8663
 8664
 8665 056072                                  FORCEXIT        255$            ;@@D
 8666 056102  013703  002320                  MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
 8667 056106  020327  003064                  CMP     R3,@TBLEND              ;END OF TSTBLK?
 8668 056112  103002                          BHIS    255$                    ;BR IF YES
 8669 056114  000137  055124                  JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
 8670 056120                          255$:
 8671
 8672 056120                                  ENDSUB                          ;////////////// END SUBTEST //////////
      056120                                                                  L10071:
      056120  104403                                                              TRAP    C$ESUB
 8673
 8674 056122  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
 8675 056126  001402                          BEQ     260$                    ;BRANCH IF NOT
 8676 056130  004737  017242                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 8677 056134                          260$:
 8678                                         .SBTTL  TEST  8: SUBTEST 4 LOOPBACK READ STROBE TEST
 8679
 8680                                 ;**
 8681                                 ; TEST 8: SUBTEST 4:
 8682                                 ;
 8683                                 ; SUBTEST DESCRIPTION:
 8684                                 ;
 8685                                 ;       This subtest verifies the Read Strobe loopback path
 8686                                 ;       can strobe the data from the Data lines to the FIFO.
 8687                                 ;       The signal IRESV4 drives IRSTR (read strobe) to write
 8688                                 ;       from the data lines to the FIFO.
 8689                                 ;
 8690                                 ; TEST STEPS:
 8691                                 ;
 8692                                 ;
 8693                                 ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
 8694                                 ;    BEGIN
 8695                                 ;       Write to TSSR register to soft initialize the controller
 8696                                 ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
 8697                                 ;       If Extended Features Hardware Switch Clear then:
 8698                                 ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
 8699                                 ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
 8700                                 ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
 8701                                 ;       Do a WRITE NPR to set loopback and tape direction OUT
 8702                                 ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
 8703                                 ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
 8704                                 ;       Do a READ FIFO with tape direction OUT to load tape out write latch
 8705                                 ;       Do a WRITE NPR to set loopback and tape direction IN
 8706                                 ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 to write loop data to FIFO
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
 8707                                   ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
 8708                                   ;         (to strobe loopback data into FIFO.)
 8709                                   ;       Do a READ FIFO with tape direction IN to read data
 8710                                   ;       If Data read from FIFO NOT= to Data sent Then Print Error
 8711                                   ;   END
 8712                                   ; -
 8713 056134                            T8.4:   BGNSUB                          ;////////////// BEGIN SUBTEST //////////
      056134                                                                                    T8.4:
      056134  104402                                                                           TRAP    C$BSUB
 8714                                   ;       Write to TSSR register to soft initialize the controller
 8715 056136                            5$:
 8716 056136  004737  016034                    JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
 8717 056142  103405                            BCS     10$                     ;BR IF SOFT INIT OKAY
 8718 056144  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8719 056146                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      056146  104455                                                                           TRAP    C$ERDF
      056150  001510                                                                           .WORD   840
      056152  003654                                                                           .WORD   SFIERR
      056154  012074                                                                           .WORD   SFIMSG
 8720                                   ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
 8721 056156  005037  002222            10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 8722 056162  012704  061720                    MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 8723 056166  004737  010662                    JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
 8724 056172                                    FORCERROR       12$                     ;@@FORCE ERROR IF FORCER=1
 8725 056206  103407                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 8726 056210  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8727 056212                                    NEXT.ERRNO
 8728 056212                            12$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      056212  104455                                                                           TRAP    C$ERDF
      056214  001511                                                                           .WORD   841
      056216  057633                                                                           .WORD   T19SSR
      056220  012106                                                                           .WORD   PKTSSR
 8729 056222  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
 8730 056226                            15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      056226  104406                                                                           TRAP    C$CLP1
 8731                                   ;       If Extended Features Hardware Switch Clear then:
 8732                                   ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
 8733 056230  012701  061742                    MOV     #T19BFR,R1              ;MESSAGE BUFFER ADDRESS
 8734 056234  032761  000200  000012            BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
 8735 056242  001026                            BNE     30$                     ;BR IF YES
 8736 056244  004737  061572                    JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
 8737 056250  012704  062070                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 8738 056254  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8739 056260  004737  016376                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8740 056264                                    FORCERROR       22$                     ;@@FORCE ERROR IF FORCER=1
 8741 056300  103407                            BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
 8742 056302  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8743 056304                                    NEXT.ERRNO
 8744 056304                            22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      056304  104455                                                                           TRAP    C$ERDF
      056306  001512                                                                           .WORD   842
      056310  057670                                                                           .WORD   T192SSR
      056312  012106                                                                           .WORD   PKTSSR
 8745 056314  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
 8746 056320                            30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      056320  104406                                                                           TRAP    C$CLP1
 8747                                   ;       Do WRITE CHARACTERISTICS to select reserved unit 7
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
 8748 056322  012704  061720                MOV     @T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 8749 056326  004737  010662                JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
 8750 056332                                FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
 8751 056346  103407                        BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
 8752 056350  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8753 056352                                NEXT.ERRNO
 8754 056352                        42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      056352  104455                                                                TRAP    C$ERDF
      056354  001513                                                                .WORD   843
      056356  057633                                                                .WORD   T19SSR
      056360  012106                                                                .WORD   PKTSSR
 8755 056362  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8756 056366                        50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      056366  104406                                                                TRAP    C$CLP1
 8757
 8758                               ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
 8759 056370  012703  002754                MOV     @TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
 8760 056374  012337  002314        100$:   MOV     (R3)+,DATA              ;GET A TEST PATTERN
 8761 056400  042737  177400  002314        BIC     @#C<377>,DATA           ;DATA IS BYTE
 8762 056406  010337  002320                MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
 8763                               ;       Do a WRITE NPR to set loopback and tape direction OUT
 8764 056412  012700  000100                MOV     @#NP.OUT,R0             ;SET TAPE DIRECTION OUT
 8765 056416  052700  000040                BIS     @#NP.LOOP,R0            ;SET LOOPBACK
 8766 056422  004737  061432                JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
 8767 056426  012704  062070                MOV     @#T19PK2,R4             ;GET WRITE SUBSYSTEM COMMAND PACKET
 8768 056432  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8769 056436  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8770 056442                                FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
 8771 056456  103407                        BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
 8772 056460  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8773 056462                                NEXT.ERRNO
 8774 056462                        102$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      056462  104455                                                                TRAP    C$ERDF
      056464  001514                                                                .WORD   844
      056466  060001                                                                .WORD   T194SSR
      056470  012106                                                                .WORD   PKTSSR
 8775 056472  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8776 056476                        105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      056476  104406                                                                TRAP    C$CLP1
 8777                               ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
 8778 056500  012700  000001                MOV     @#WF.I4RES,R0           ;IRESV4==>IRSTR=1
 8779 056504  004737  061552                JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
 8780 056510  012704  062070                MOV     @#T19PK2,R4             ;GET WRITE SUBSYSTEM COMMAND PACKET
 8781 056514  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 8782 056520  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 8783 056524                                FORCERROR       112$                    ;@@DFORCE ERROR IF FORCER=1
 8784 056540  103407                        BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
 8785 056542  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 8786 056544                                NEXT.ERRNO
 8787 056544                        112$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      056544  104455                                                                TRAP    C$ERDF
      056546  001515                                                                .WORD   845
      056550  060222                                                                .WORD   T198SSR
      056552  012106                                                                .WORD   PKTSSR
 8788 056554  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 8789 056560                        120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      056560  104406                                                                TRAP    C$CLP1
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
8790
8791 056562  012700  000001          ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
8792 056566  012701  002314                  MOV     #1,R0                   ;WRITE 1 BYTE
8793 056572  004737  061476                  MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
8794 056576  012704  062070                  JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
8795 056602  010465  000000                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8796 056606  004737  016376                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8797 056612                                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8798 056626  103407                          FORCERROR       132$                    ;@@FORCE ERROR IF FORCER=1
8799 056630  010001                          BCS     140$                    ;BR IF CARRY SET (GOOD RETURN)
8800 056632                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8801 056632                                  NEXT.ERRNO
      056632  104455          132$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      056634  001516                                                          TRAP    C$ERDF
      056636  060044                                                          .WORD   846
      056640  012106                                                          .WORD   T195SSR
                                                                              .WORD   PKTSSR
8802 056642  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
8803 056646                          140$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      056646  104406                                                          TRAP    C$CLP1
8804
8805 056650  012700  000001          ;       Do a READ FIFO with tape direction OUT to load tape out write latch
8806 056654  004737  061456                  MOV     #1,R0                   ;SET READ BYTE COUNT
8807 056660  012704  062070                  JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
8808 056664  010465  000000                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8809 056670  004737  016376                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8810 056674                                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8811 056710  103407                          FORCERROR       152$                    ;@@FORCE ERROR IF FORCER=1
8812 056712  010001                          BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
8813 056714                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8814 056714                                  NEXT.ERRNO
      056714  104455          152$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      056716  001517                                                          TRAP    C$ERDF
      056720  060110                                                          .WORD   847
      056722  012106                                                          .WORD   T196SSR
                                                                              .WORD   PKTSSR
8815 056724  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
8816 056730                          160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      056730  104406                                                          TRAP    C$CLP1
8817
8818 056732  005000          ;       Do a WRITE NPR to set loopback and tape direction IN
8819 056734  052700  000040                  CLR     R0                      ;CLR NP.OUT TO SET TAPE DIRECTION IN
8820 056740  004737  061432                  BIS     #NP.LOOP,R0             ;SET LOOPBACK
8821 056744  012704  062070                  JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
8822 056750  010465  000000                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8823 056754  004737  016376                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8824 056760                                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8825 056774  103407                          FORCERROR       182$                    ;@@FORCE ERROR IF FORCER=1
8826 056776  010001                          BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
8827 057000                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8828 057000                                  NEXT.ERRNO
      057000  104455          182$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      057002  001520                                                          TRAP    C$ERDF
      057004  060001                                                          .WORD   848
      057006  012106                                                          .WORD   T194SSR
                                                                              .WORD   PKTSSR
8829 057010  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
8830 057C.                           190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      057C.   104406                                                          TRAP    C$CLP1
8831                                 ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 0
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
8832 057016  005000                        CLR     R0                      ;SET IRESV4==>IRSTR=0
8833 057020  004737  061552                JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
8834 057024  012704  062070                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8835 057030  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8836 057034  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8837 057040                                FORCERROR       202$                    ;@@DFORCE ERROR IF FORCER=1
8838 057054  103407                        BCS     210$                    ;BR IF CARRY SET (GOOD RETURN)
8839 057056  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8840 057060                                NEXT.ERRNO
8841 057060                        202$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057060  104455                                                                  TRAP    C$ERDF
     057062  001521                                                                  .WORD   849
     057064  060222                                                                  .WORD   T198SSR
     057066  012106                                                                  .WORD   PKTSSR
8842 057070  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
8843 057074                        210$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057074  104406                                                                  TRAP    C$CLP1
8844                                ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
8845 057076  012700  000001                MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR=1
8846 057102  004737  061552                JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
8847 057106  012704  062070                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8848 057112  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8849 057116  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8850 057122                                FORCERROR       222$                    ;@@DFORCE ERROR IF FORCER=1
8851 057136  103407                        BCS     230$                    ;BR IF CARRY SET (GOOD RETURN)
8852 057140  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8853 057142                                NEXT.ERRNO
8854 057142                        222$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057142  104455                                                                  TRAP    C$ERDF
     057144  001522                                                                  .WORD   850
     057146  060222                                                                  .WORD   T198SSR
     057150  012106                                                                  .WORD   PKTSSR
8855 057152  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
8856 057156                        230$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057156  104406                                                                  TRAP    C$CLP1
8857                                ;       Do a READ FIFO with tape direction IN to read data
8858 057160  012700  000001                MOV     #1,R0                   ;SET READ BYTE COUNT
8859 057164  004737  061456                JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
8860 057170  012704  062070                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
8861 057174  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
8862 057200  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
8863 057204                                FORCERROR       282$                    ;@@DFORCE ERROR IF FORCER=1
8864 057220  103407                        BCS     290$                    ;BR IF CARRY SET (GOOD RETURN)
8865 057222  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
8866 057224                                NEXT.ERRNO
8867 057224                        282$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057224  104455                                                                  TRAP    C$ERDF
     057226  001523                                                                  .WORD   851
     057230  060110                                                                  .WORD   T196SSR
     057232  012106                                                                  .WORD   PKTSSR
8868 057234  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
8869 057240                        290$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057240  104406                                                                  TRAP    C$CLP1
8870                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
8871 057242  004737  061630                JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
8872 057246  012701  057472                MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
8873 057252  012702  061762                MOV     #T198FSTA,R2            ;GET RECV READ STATUS
```

M16

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
 8874 057256  013711  002314                    MOV     DATA,(R1)               ;SET EXPD WORD #8 = DATA
 8875 057262  016261  000002  000002            MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
 8876 057270  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 8877 057272  012701  061742                    MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 8878 057276  012702  057452                    MOV     #T19EXP,R2              ;EXPD ADDRESS
 8879 057302  012703  000022                    MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
 8880 057306  004737  011540                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 8881 057312                                    FORCERROR       302$,NOTSSR     ;a@D
 8882 057322  103404                            BCS     310$                    ;BR IF YES
 8883 057324                                    NEXT.ERRNO
 8884 057324                            302$:   ERRHRD  ERRNO,T19RSTR,MSGSUB    ;REPORT ERROR
      057324  104456                                                                    TRAP    C$ERHRD
      057326  001524                                                                    .WORD   852
      057330  061240                                                                    .WORD   T19RSTR
      057332  014002                                                                    .WORD   MSGSUB
 8885 057334                            310$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      057334  104406                                                                    TRAP    C$CLP1
 8886
 8887
 8888 057336                                    FORCEXIT        355$            ;a@D
 8889 057346  013703  002320                    MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
 8890 057352  020327  003064                    CMP     R3,#TBLEND              ;END OF TSTBLK?
 8891 057356  103002                            BHIS    355$                    ;BR IF YES
 8892 057360  000137  056374                    JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
 8893 057364                            355$:
 8894
 8895 057364                                    ENDSUB                          ;////////////// END SUBTEST //////////
      057364                                                                            L10072:
      057364  104403                                                                    TRAP    C$ESUB
 8896
 8897 057366  005737  002222                    TST     FATFLG                  ;ANY FATAL ERRORS ?
 8898 057372  001402                            BEQ     360$                    ;BRANCH IF NOT
 8899 057374  004737  017242                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 8900 057400                            360$:
 8901
 8902 057400                            EXIT    TST                             ;////////// EXIT TEST //////////
      057400  104432                                                                    TRAP    C$EXIT
      057402  002602                                                                    .WORD   L10066-.
 8903
 8904
 8905                                   ;+
 8906                                   ;LOCAL STORAGE FOR THIS TEST
 8907                                   ;-
 8908
 8909 057404  000000                    T19PREV:        .WORD   0               ;DRIVE SIGNAL 1-0 TRANSITION FLAG
 8910                                   ;+
 8911                                   ; LOOPBACK DRIVE SIGNAL TABLE
 8912                                   ; THIS TABLE IS USED BY T19CNVT TO SETUP
 8913                                   ; A DRIVE PATTERN FROM THE TEST DATA INPUT PATTERN.
 8914                                   ;
 8915                                   ; WRITE CONTROL SIGNALS ARE OF FORM WC.XXX
 8916                                   ; WRITE FORMAT SIGNALS ARE OF FORM WF.XXXX
 8917                                   ;
 8918 057406                            T19BFCTL:                               ;WRITE CONTROL DRIVE SIGNALS
 8919 057406  000001                            WC.IGO                          ;IGO==>IFPT     DATA<0>
 8920 057410  000002                            WC.IFEN                         ;IFEN==>IFBY    DATA<1>
 8921 057412  000004                            WC.IRWU                         ;IRWU==>IRWD    DATA<2>
```

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

 8922 057414  000010                              WC.IREW              ;IREW==>IDBY    DATA<3>
 8923 057416  002000                              WF.IERASE*256.       ;IFAD==>ILDP    DATA<4>
 8924 057420  000040                              WC.I1TAD             ;ITAD1==>IONL   DATA<5>
 8925 057422  000100                              WC.IOTAD             ;ITAD0==>IRDY   DATA<6>
 8926 057424  000200                              WC.IFAD              ;IERASE==>ISPEED DATA<7>
 8927 057426  004000                              WF.IEDIT*256.        ;IEDIT==>IHER   DATA<8>
 8928 057430  010000                              WF.IWFM*256.         ;IWFM==>IFMK    DATA<9>
 8929 057432  020000                              WF.IREV*256.         ;IREV==>ICER    DATA<10>
 8930 057434  040000                              WF.IWRT*256.         ;IWRT==>IIDENT  DATA<11>
 8931 057436  100000                              WF.IHISP*256.        ;IHISP==>IEOT   DATA<12>
 8932 057440  000000                              .WORD   0            ;IRESV2 (UNUSED)DATA<13>
 8933 057442  000000                              .WORD   0            ;IRESV1 (UNUSED)DATA<14>
 8934 057444  000000                              .WORD   0            ;PARERR (UNTESTED)DATA<15>
 8935
 8936 057446                          T19MSK:                          ;MASK OF UNTESTED BITS IN READ STATUS BYTES
 8937                                                                  ;UNTESTED BITS ARE SET TO 1
 8938 057446     377                          .BYTE   ^C<000>         ;BYTE 0 MASK
 8939 057447     037                          .BYTE   ^C<340>         ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
 8940 057450     360                          .BYTE   ^C<017>         ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
 8941 057451     000                          .BYTE   0               ;MAKE IT EVEN
 8942
 8943 057452                          T19EXP:                          ;BEGIN EXPECTED DATA BUFFER
 8944 057452  000000                          .WORD   0               ;MESSAGE TYPE
 8945 057454  000000                          .WORD   0               ;DATA FIELD LENGTH
 8946 057456  000000                          .WORD   0               ;RBPCR
 8947 057460  000000                          .WORD   0               ;XST0
 8948 057462  000000                          .WORD   0               ;XST1
 8949 057464  000000                          .WORD   0               ;XST2
 8950 057466  000000                          .WORD   0               ;XST3
 8951 057470  000000                          .WORD   0               ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
 8952 057472                          T19EXSTA: .BLKB 64.              ;EXPECTED READ STATUS AND WRITE FIFO DATA
 8953 057572                          T19EXEND:                                ;END EXPECTED DATA BUFFER
 8954                                 ;*
 8955                                 ;LOCAL TEXT MESSAGES FOR TEST
 8956                                 ;-
 8957
 8958 057572  124  162  141  TST19ID:        .ASCIZ  'Transport Bus Interface Loopback'
 8959 057633  127  122  111  T19SSR:  .ASCIZ  'WRITE CHARACTERISTICS Failed'
 8960 057670  127  122  111  T192SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
 8961 057734  127  122  111  T193SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
 8962 060001  127  122  111  T194SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
 8963 060044  127  122  111  T195SSR:.ASCIZ   WRITE SUBSYSTEM (Write FIFO) Failed'
 8964 060110  127  122  111  T196SSR:.ASCIZ  'WRITE SUBSYSTEM (Read FIFO) Failed'
 8965 060153  127  122  111  T197SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Control) Failed'
 8966 060222  127  122  111  T198SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Format) Failed'
 8967 060270  106  111  106  T191CMP:.ASCIZ  'FIFO Status in WORD #9 Incorrect after Initialize'
 8968 060352  122  145  141  T192CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
 8969 060446  124  141  160  T193CMP:.ASCIZ  'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
 8970 060534  122  145  141  T195CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO Data'
 8971 060610  106  111  106  T196CMP:.ASCIZ  'FIFO Status (in WORD #9) Incorrect after READ FIFO'
 8972 060673  124  141  160  T197CMP:.ASCIZ  'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
 8973 060761  103  157  156  T198CMP:.ASCIZ  'Control Signal Loopback Data Error, Data is in WORD #8'
 8974 061050  122  145  141  T199CMP:.ASCIZ  'Read/Write Loopback Data Error, Data is in WORD #8'
 8975 061133  114  157  157  T19WSTR:.ASCIZ  'Loopback Data Error when strobed by Write strobe, Data is in WORD #8'
 8976 061240  114  157  157  T19RSTR:.ASCIZ  'Loopback Data Error when strobed by Read Strobe, Data is in WORD #8'
 8977
 8978                                 .EVEN
```

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

8979
8980
8981                                    ;*
8982                                    ; CLEAR MESSAGE BUFFER
8983                                    ;-
8984 061344                             T19CLRBUF:
8985 061344                                     SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
8986 061350  012701  061742                     MOV     #T19BFR,R1              ;GET MESSAGE BUFFER ADDRESS
8987 061354  012702  000120                     MOV     #T19BEND-T19BFR,R2      ;SIZE OF MESSAGE BUFFER IN BYTES
8988 061360  105021                     10$:    CLRB    (R1)+                   ;CLEAR A BYTE
8989 061362  005302                             DEC     R2                      ;DONE?
8990 061364  003375                             BGT     10$                     ;BR IF NO
8991 061366  000207                             RTS     PC                      ;RETURN
8992
8993                                    ;*
8994                                    ; SETUP T19PK2 PACKET FOR READ STATUS
8995                                    ;-
8996 061370                             T19SRD:
8997 061370  004737  061344                     JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
8998 061374  012700  062100                     MOV     #T19DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
8999 061400  112720  000005                     MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
9000 061404  105010                             CLRB    (R0)                    ;CLEAR BSEL1
9001 061406  000207                             RTS     PC                      ;RETURN
9002
9003                                    ;*
9004                                    ; SETUP T19PK2 PACKET FOR WRITE MISC Reset Tape Status F-FLOPS
9005                                    ;-
9006 061410                             T19RSFIF:
9007 061410  004737  061344                     JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
9008 061414  012700  062100                     MOV     #T19DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
9009 061420  112720  000010                     MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
9010 061424  112710  000030                     MOVB    #MS.RSFIF!MS.RSTAP,(R0) ;STORE BSEL1 CLEAR FIFO CODES
9011 061430  000207                             RTS     PC                      ;RETURN
9012
9013                                    ;*
9014                                    ; SETUP T19PK2 PACKET FOR WRITE NPR
9015                                    ;
9016                                    ; INPUT:
9017                                    ;       R0 CONTAINS BSEL1 NPR DATA
9018                                    ;
9019                                    ;       SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
9020                                    ;-
9021 061432                             T19SNPR:
9022 061432  004737  061344                     JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
9023 061436  012701  062100                     MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
9024 061442  112721  000011                     MOVB    #PW.WNPR,(R1)+          ;STORE WRITE NPR IN BSEL0
9025 061446  052700  000020                     BIS     #NP.WRP,R0              ;DON'T WRITE WRONG PARITY
9026 061452  110011                             MOVB    R0,(R1)                 ;STORE NPR DATA IN BSEL1
9027 061454  000207                             RTS     PC                      ;RETURN
9028
9029                                    ;*
9030                                    ; SETUP T19PK2 PACKET FOR READ FIFO
9031                                    ;
9032                                    ; INPUT:
9033                                    ;       R0 CONTAINS SEL2 BYTE COUNT
9034                                    ;-
9035 061456                             T19RFIF:
9036 061456  004737  061344                     JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
```

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

 9036 061462  012701  062100                MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
 9037 061466  112721  000003                MOVB    #PW.RFIFO,(R1)+         ;STORE READ FIFO IN BSEL0
 9038 061472  110021                        MOVB    R0,(R1)+                ;STORE BYTE COUNT IN BSEL1
 9039 061474  000207                        RTS     PC                      ;RETURN
 9040                                ;+
 9041                                ; SETUP T19PK2 PACKET FOR WRITE FIFO
 9042                                ;
 9043                                ; INPUT:
 9044                                ;       R0 CONTAINS BYTE COUNT
 9045                                ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
 9046                                ;-
 9047 061476                        T19WFIF:
 9048 061476                                SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
 9049 061502  004737  061344                JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
 9050 061506  012702  062100                MOV     #T19DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
 9051 061512  112722  000004                MOVB    #PW.WFIFO,(R2)+         ;STORE WRITE FIFO IN BSEL0
 9052 061516  110022                        MOVB    R0,(R2)+                ;STORE BYTE COUNT IN BSEL1
 9053 061520  005022                        CLR     (R2)+                   ;CLEAR SEL2 (UNUSED)
 9054 061522  112122                10$:    MOVB    (R1)+,(R2)+             ;STORE DATA PATTERN BYTE
 9055 061524  005300                        DEC     R0                      ;DONE ALL BYTES?
 9056 061526  003375                        BGT     10$                     ;BR IF NO
 9057 061530  000207                        RTS     PC                      ;RETURN
 9058                                ;+
 9059                                ; SETUP T19PK2 FOR WRITE CONTROL
 9060                                ;
 9061                                ; INPUT:
 9062                                ;       R0 CONTAINS DRIVING DATA PATTERN
 9063                                ;-
 9064 061532                        T19WCTL:
 9065 061532  004737  061344                JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
 9066 061536  012701  062100                MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
 9067 061542  112721  000006                MOVB    #PW.WCTL,(R1)+          ;STORE WRITE CONTROL IN BSEL0
 9068 061546  110021                        MOVB    R0,(R1)+                ;STORE DATA WORD IN BSEL1
 9069 061550  000207                        RTS     PC                      ;RETURN
 9070                                ;+
 9071                                ; SETUP T19PK2 FOR WRITE FORMAT TRANSPORT REGISTER
 9072                                ;
 9073                                ; INPUT:
 9074                                ;       R0 CONTAINS DRIVING DATA PATTERN
 9075                                ;-
 9076 061552                        T19WFMT:
 9077 061552  004737  061344                JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
 9078 061556  012701  062100                MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
 9079 061562  112721  000007                MOVB    #PW.WFMT,(R1)+          ;STORE WRITE FORMAT IN BSEL0
 9080 061566  110021                        MOVB    R0,(R1)+                ;STORE DATA WORD IN BSEL1
 9081 061570  000207                        RTS     PC                      ;RETURN
 9082                                ;+
 9083                                ; SETUP T19PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
 9084                                ;-
 9085 061572                        T19SEXT:
 9086 061572  012700  062100                MOV     #T19DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
 9087 061576  112720  000010                MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
 9088 061602  112710  000200                MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
 9089 061606  000207                        RTS     PC                      ;RETURN
 9090                                ;+
 9091                                ; CLEAR EXPECTED DATA MESSAGE BUFFER
 9092                                ;-
```

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

 9093 061610                            T19CLEXP:
 9094 061610  012701  057452                       MOV     #T19EXP,R1              ;GET EXPD ADDRESS
 9095 061614  012700  000120                       MOV     #T19EXEND-T19EXP,R0     ;GET EXPD SIZE
 9096 061620  105021                    10$:       CLRB    (R1)+                   ;CLEAR A BYTE
 9097 061622  005300                               DEC     R0                      ;DONE?
 9098 061624  003375                               BGT     10$                     ;BR IF NO
 9099 061626  000207                               RTS     PC                      ;RETURN
 9100
 9101                                   ;+
 9102                                   ;Set WORDS 0-7 of expd message BUFFER = to recv since not testing
 9103                                   ;-
 9104 061630                            T19SETEXP:
 9105 061630  012702  057452                       MOV     #T19EXP,R2              ;GET EXPD
 9106 061634  012703  061742                       MOV     #T19BFR,R3              ;GET READ STATUS RECV BUFFER
 9107 061640  012700  000010                       MOV     #8.,R0                  ;SET WORDS 0-7 EXP=RECV
 9108 061644  012322                    5$:        MOV     (R3)+,(R2)+             ;SET EXPD=RECV
 9109 061646  005300                               DEC     R0                      ;DONE WORDS 0-7 WORDS?
 9110 061650  003375                               BGT     5$                      ;BR IF NO
 9111 061652  000207                               RTS     PC                      ;RETURN
 9112
 9113                                   ;+
 9114                                   ; CONVERT A TEST PATTERN DATA WORD TO LOOPBACK DRIVE SIGNALS
 9115                                   ;
 9116                                   ; INPUTS:
 9117                                   ;
 9118                                   ;       R0      TEST PATTERN
 9119                                   ;
 9120                                   ; IMPLICIT INPUTS:
 9121                                   ;
 9122                                   ;       T19BFCTL  - CONTAINS WRITE CONTROL / WRITE FORMAT CONVERSION BITS
 9123                                   ;
 9124                                   ; OUTPUTS:
 9125                                   ;
 9126                                   ;       R0      - LOW BYTE CONTAINS WRITE CONTROL DATA
 9127                                   ;               - HIGH BYTE CONTAINS WRITE FORMAT DATA
 9128                                   ;-
 9129 061654                            T19CNVT:
 9130 061654                                       SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
 9131 061660  012701  057406                       MOV     #T19BFCTL,R1            ;CONVERSION TABLE ADDRESS
 9132 061664  005002                               CLR     R2                      ;INIT RESULT OF CONVERSION
 9133 061666  012703  000020                       MOV     #16.,R3                 ;BIT COUNT
 9134 061672  006000                    10$:       ROR     R0                      ;IS THIS BIT EQUAL TO 1?
 9135 061674  103001                               BCC     20$                     ;BR IF NO
 9136 061676  051102                               BIS     (R1),R2                 ;SET CONVERTED BIT
 9137 061700  005721                    20$:       TST     (R1)+                   ;POINT TO NEXT BIT IN CONVERSION TABLE
 9138 061702  005303                               DEC     R3                      ;DONE?
 9139 061704  003372                               BGT     10$                     ;BR IF NO
 9140 061706  010200                               MOV     R2,R0                   ;COPY RESULT
 9141 061710  000207                               RTS     PC                      ;RETURN
 9142
 9143
 9144
 9146         061720                               .=<.+10>&177770
 9148                                   ;
 9149                                   ;WRITE CHARACTERISTICS COMMAND PACKET
 9150                                   ;
 9151 061720                            T19PACKET:                              ;COMMAND PACKET FOR TEST
```

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

 9152 061720  100004                    .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
 9153 061722  061730                    .WORD   T19DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
 9154 061724  000000                    .WORD   0
 9155 061726  000012                    .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
 9156
 9157 061730                    T19DATA:                                ;CHARACTERISTICS DATA BLOCK
 9158 061730  061742                    .WORD   T19BFR                  ;ADDRESS OF MESSAGE BUFFER
 9159 061732  000000                    .WORD   0
 9160 061734  000024                    .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
 9161 061736  000000                    .WORD   0                       ;ESS,ENB,EAI,ERI
 9162 061740  000007                    .WORD   7                       ;EXTENDED FEATURES UNIT NO.
 9163
 9164
 9165                           ;MESSAGE BUFFER FOR ALL TEST 8 COMMANDS
 9166
 9167 061742                    T19BFR:                                 ;BEGIN MESSAGE BUFFER
 9168 061742  000000                    .WORD   0                       ;MESSAGE TYPE
 9169 061744  000000                    .WORD   0                       ;DATA FIELD LENGTH
 9170 061746  000000                    .WORD   0                       ;RBPCR
 9171 061750  000000                    .WORD   0                       ;XST0
 9172 061752  000000                    .WORD   0                       ;XST1
 9173 061754  000000                    .WORD   0                       ;XST2
 9174 061756  000000                    .WORD   0                       ;XST3
 9175 061760  000000                    .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
 9176 061762                    T19BFSTA: .BLKB 64.                     ;READ STATUS AND WRITE FIFO BUFFER
 9177 062062                    T19BEND:                                ;END OF MESSAGE BUFFER
 9178                           ;
 9179                           ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
 9180                           ;
 9182         062070                    .=<.+10>&177770
 9184 062070                    T19PK2:
 9185 062070  100006                    .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
 9186 062072  062100                    .WORD   T19DT2                  ;LOW ADDRESS OF DATA BLOCK
 9187 062074  000000                    .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
 9188 062076  000012                    .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
 9189
 9190 062100                    T19DT2:                                 ;DATA BLOCK
 9191 062100     000            .BYTE   0                               ;BSEL0
 9192 062101     000            .BYTE   0                               ;BSEL1
 9193 062102  000000                    .WORD   0                       ;SEL2
 9194 062104                            .BLKB   64.                     ;WRITE FIFO DATA OUTPUT BUFFER
 9195
 9196
 9197 062204                    ENDTST
      062204                                                    L10066:
      062204  104401                                                   TRAP    C$ETST
 9198                           .SBTTL  TEST  9: READ/WRITE DATA PARITY TEST
 9199                           ;++
 9200                           ; TEST DESCRIPTION:
 9201                           ;
 9202                           ;       This test verifies that the Write Data Parity generator
 9203                           ;       and the Read Data Parity checker operate properly. The
 9204                           ;       Transport Bus signal loopback mode is enabled and a
 9205                           ;       Set Wrong parity function is executed. Then various
 9206                           ;       Write Subsystem Memory functions are performed to
 9207                           ;       write data to and from the FIFO in loopback mode.
 9208                           ;       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 W ITE 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
                                ;--


062206                                  BGNTST
062206                                                                          T9::
062206  012700  064572                  MOV     #TST20ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
062212  004737  016550                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
062216  012737  000012  002216          MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
062224                          T20LOOP:
```

TEST 9: READ/WRITE DATA PARITY TEST

```
 9269 062224                                    BGNSUB                          ;////////////// BEGIN SUBTEST //////'////
      062224                                                                                      T9.1:
      062224  104402                                                                              TRAP    C$BSUB
 9270                                   ;       Write to TSSR register to soft initialize the controller
 9271 062226                            5$:
 9272 062226  004737  016034                    JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
 9273 062232  103405                            BCS     10$                     ;BR IF SOFT INIT OKAY
 9274 062234  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9275 062236                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      062236  104455                                                                              TRAP    C$ERDF
      062240  001604                                                                              .WORD   900
      062242  003654                                                                              .WORD   SFIERR
      062244  012074                                                                              .WORD   SFIMSG
 9276                                   ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
 9277 062246  005037  002222            10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 9278 062252  012704  065770                    MOV     #T20PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 9279 062256  004737  010662                    JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
 9280 062262                                    FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
 9281 062276  103407                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 9282 062300  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9283 062302                                    NEXT.ERRNO
 9284 062302                            12$:    ERRDF   ERRNO,T20SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      062302  104455                                                                              TRAP    C$ERDF
      062304  001605                                                                              .WORD   901
      062306  064621                                                                              .WORD   T20SSR
      062310  012106                                                                              .WORD   PKTSSR
 9285 062312  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
 9286 062316                            15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      062316  104406                                                                              TRAP    C$CLP1
 9287                                   ;       If Extended Features Hardware Switch Clear then:
 9288                                   ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
 9289 062320  012701  066012                    MOV     #T20BFR,R1              ;MESSAGE BUFFER ADDRESS
 9290 062324  032761  000200  000012            BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
 9291 062332  001026                            BNE     30$                     ;BR IF YES
 9292 062334  004737  065706                    JSR     PC,T20SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
 9293 062340  012704  066140                    MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9294 062344  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9295 062350  004737  016376                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9296 062354                                    FORCERROR       22$                     ;@@DFORCE ERROR IF FORCER=1
 9297 062370  103407                            BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
 9298 062372  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9299 062374                                    NEXT.ERRNO
 9300 062374                            22$:    ERRDF   ERRNO,T202SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      062374  104455                                                                              TRAP    C$ERDF
      062376  001606                                                                              .WORD   902
      062400  064656                                                                              .WORD   T202SSR
      062402  012106                                                                              .WORD   PKTSSR
 9301 062404  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
 9302 062410                            30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      062410  104406                                                                              TRAP    C$CLP1
 9303                                   ;       Do WRITE CHARACTERISTICS to select reserved unit 7
 9304 062412  012704  065770                    MOV     #T20PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 9305 062416  004737  010662                    JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
 9306 062422                                    FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
 9307 062436  103407                            BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
 9308 062440  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9309 062442                                    NEXT.ERRNO
```

TEST 9: READ/WRITE DATA PARITY TEST

```
9310 062442                            42$:    ERRDF   ERRNO,T20SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     062442  104455                                                                    TRAP    C$ERDF
     062444  001607                                                                    .WORD   903
     062446  064621                                                                    .WORD   T20SSR
     062450  012106                                                                    .WORD   PKTSSR
9311 062452  005237  002222                        INC     FATFLG                  ;SET FATAL ERROR FLAG
9312 062456                            50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     062456  104406                                                                    TRAP    C$CLP1
9313
9314
9315                                   ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
9316 062460  012703  002754                    MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
9317 062464  012337  002314            100$:   MOV     (R3)+,DATA              ;GET A TEST PATTERN
9318 062470  042737  177400  002314            BIC     #^C<377>,DATA           ;DATA IS BYTE
9319 062476  010337  002320                    MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
9320                                   ;       Do a WRITE NPR to set loopback and tape direction OUT and
9321                                   ;                 and SET Write Wrong Parity.
9322 062502  012700  000100                    MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
9323 062506  052700  000040                    BIS     #NP.LOOP,R0             ;SET LOOPBACK
9324 062512  042700  000020                    BIC     #NP.WRP,R0              ;SET WRITE WRONG PARITY (INVERTED)
9325 062516  004737  065556                    JSR     PC,T20WNPR              ;SETUP T20PK2 FOR WRITE NPR
9326 062522  012704  066140                    MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
9327 062526  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
9328 062532  004737  016376                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
9329 062536                                    FORCERROR       102$                    ;@@FORCE ERROR IF FORCER=1
9330 062552  103407                            BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
9331 062554  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9332 062556                                    NEXT.ERRNO
9333 062556                            102$:   ERRDF   ERRNO,T204SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     062556  104455                                                                    TRAP    C$ERDF
     062560  001610                                                                    .WORD   904
     062562  064767                                                                    .WORD   T204SSR
     062564  012106                                                                    .WORD   PKTSSR
9334 062566  005237  002222                        INC     FATFLG                  ;SET FATAL ERROR FLAG
9335 062572                            105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     062572  104406                                                                    TRAP    C$CLP1
9336                                   ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
9337 062574  012700  000001                    MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR = 1
9338 062600  004737  065652                    JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
9339 062604  012704  066140                    MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
9340 062610  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
9341 062614  004737  016376                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
9342 062620                                    FORCERROR       112$                    ;@@FORCE ERROR IF FORCER=1
9343 062634  103407                            BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
9344 062636  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9345 062640                                    NEXT.ERRNO
9346 062640                            112$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     062640  104455                                                                    TRAP    C$ERDF
     062642  001611                                                                    .WORD   905
     062644  065141                                                                    .WORD   T208SSR
     062646  012106                                                                    .WORD   PKTSSR
9347 062650  005237  002222                        INC     FATFLG                  ;SET FATAL ERROR FLAG
9348 062654                            120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     062654  104406                                                                    TRAP    C$CLP1
9349                                   ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
9350 062656  012700  000001                    MOV     #1,R0                   ;WRITE 1 BYTE
9351 062662  012701  002314                    MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
```

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

 9352 062666  004737  065616                  JSR     PC,T20WFIF              ;SETUP T20PK2 FOR WRITE FIFO
 9353 062672  012704  066140                  MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9354 062676  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9355 062702  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9356 062706                                  FORCERROR       152$                    ;@@FORCE ERROR IF FORCER=1
 9357 062722  103407                          BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
 9358 062724  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9359 062726                                  NEXT.ERRNO
 9360 062726                          152$:   ERRDF   ERRNO,T205SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      062726  104455                                                                  TRAP    C$ERDF
      062730  001612                                                                  .WORD   906
      062732  065032                                                                  .WORD   T205SSR
      062734  012106                                                                  .WORD   PKTSSR
 9361 062736  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
 9362 062742                          160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      062742  104406                                                                  TRAP    C$CLP1
 9363                                 ;       Do a READ FIFO with tape direction OUT to load tape out write latch
 9364                                 ;         (this is when wrong parity (IWP) is set)
 9365 062744  012700  000001                  MOV     #1,R0                   ;SET READ BYTE COUNT
 9366 062750  004737  065576                  JSR     PC,T20RFIF              ;SETUP T20PK2 FOR READ FIFO
 9367 062754  012704  066140                  MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9368 062760  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9369 062764  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9370 062770                                  FORCERROR       172$                    ;@@FORCE ERROR IF FORCER=1
 9371 063004  103407                          BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
 9372 063006  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9373 063010                                  NEXT.ERRNO
 9374 063010                          172$:   ERRDF   ERRNO,T206SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      063010  104455                                                                  TRAP    C$ERDF
      063012  001613                                                                  .WORD   907
      063014  065076                                                                  .WORD   T206SSR
      063016  012106                                                                  .WORD   PKTSSR
 9375 063020  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
 9376 063024                          180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      063024  104406                                                                  TRAP    C$CLP1
 9377                                 ;       Do a WRITE FORMAT  to set IRESV4-->IRSTR = 0 (sets read strobe low)
 9378                                 ;               (Read Strobe sets PAR IN H [Parity Error])
 9379 063026  005000                          CLR     R0                      ;IRESV4-->IRSTR = 0
 9380 063030  004737  065652                  JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
 9381 063034  012704  066140                  MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9382 063040  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9383 063044  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9384 063050                                  FORCERROR       192$                    ;@@FORCE ERROR IF FORCER=1
 9385 063064  103407                          BCS     200$                    ;BR IF CARRY SET (GOOD RETURN)
 9386 063066  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9387 063070                                  NEXT.ERRNO
 9388 063070                          192$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      063070  104455                                                                  TRAP    C$ERDF
      063072  001614                                                                  .WORD   908
      063074  065141                                                                  .WORD   T208SSR
      063076  012106                                                                  .WORD   PKTSSR
 9389 063100  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
 9390 063104                          200$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      063104  104406                                                                  TRAP    C$CLP1
 9391                                 ;       Do a WRITE FORMAT  to set IRESV4-->IRSTR = 1 (sets read strobe high)
 9392 063106  012700  000001                  MOV     #WF.I4RES,R0            ;IRESV4-->IRSTR = 1
 9393 063112  004737  065652                  JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
```

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

 9394 063116  012704  066140                MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9395 063122  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9396 063126  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9397 063132                                FORCERROR       212$                    ;@@FORCE ERROR IF FORCER=1
 9398 063146  103407                        BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
 9399 063150  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9400 063152                                NEXT.ERRNO
 9401 063152                        212$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      063152  104455                                                                TRAP    C$ERDF
      063154  001615                                                                .WORD   909
      063156  065141                                                                .WORD   T208SSR
      063160  012106                                                                .WORD   PKTSSR
 9402 063162  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 9403 063166                        220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      063166  104406                                                                TRAP    C$CLP1
 9404                               ;       Do a Write Subsystem READ STATUS
 9405 063170  004737  065536                JSR     PC,T20SRD               ;SETUP PACKET FOR READ STATUS
 9406 063174  012704  066140                MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9407 063200  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9408 063204  004737  016376                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9409 063210                                FORCERROR       232$                    ;@@FORCE ERROR IF FORCER=1
 9410 063224  103407                        BCS     240$                    ;BR IF CARRY SET (GOOD RETURN)
 9411 063226  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9412 063230                                NEXT.ERRNO
 9413 063230                        232$:   ERRDF   ERRNO,T203SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      063230  104455                                                                TRAP    C$ERDF
      063232  001616                                                                .WORD   910
      063234  064722                                                                .WORD   T203SSR
      063236  012106                                                                .WORD   PKTSSR
 9414 063240  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 9415 063244                        240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      063244  104406                                                                TRAP    C$CLP1
 9416                               ;       If Read Data parity error NOT=1  Then Print Error
 9417 063246  004737  065744                JSR     PC,T20SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
 9418 063252  012701  064472                MOV     #T20EXSTA,R1            ;GET EXPECTED READ STATUS
 9419 063256  012702  066032                MOV     #T208FSTA,R2            ;GET RECV READ STATUS
 9420 063262  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
 9421 063264  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTED)
 9422 063272  052711  100000                BIS     #S1.PARERR,(R1)         ;SET EXP PAR ERR  =1
 9423 063276  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 9424 063300  012701  066012                MOV     #T208FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 9425 063304  012702  064452                MOV     #T20EXP,R2              ;EXPD ADDRESS
 9426 063310  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 9427 063314  004737  011540                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 9428 063320                                FORCERROR       252$,NOTSSR             ;@@D
 9429 063330  103404                        BCS     260$                    ;BR IF YES
 9430 063332                                NEXT.ERRNO
 9431 063332                        252$:   ERRHRD  ERRNO,T20SWP,MSGSTAT    ;REPORT ERROR
      063332  104456                                                                TRAP    C$ERHRD
      063334  001617                                                                .WORD   911
      063336  065207                                                                .WORD   T20SWP
      063340  012410                                                                .WORD   MSGSTAT
 9432 063342                        260$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      063342  104406                                                                TRAP    C$CLP1
 9433                               ;       Do a Write Misc to RESET FIFO
 9434 063344  012700  000020                MOV     #MS.RSFIF,R0            ;SET RESET FIFO COMMAND
 9435 063350  004737  065672                JSR     PC,T20WMISC             ;SETUP T20PK2 FOR WRITE MISC
```

TEST 9: READ/WRITE DATA PARITY TEST

```
9436 063354  012704  066140              MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
9437 063360  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
9438 063364  004737  016376              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
9439 063370                              FORCERROR       282$                    ;@@FORCE ERROR IF FORCER=1
9440 063404  103407                      BCS     290$                    ;BR IF CARRY SET (GOOD RETURN)
9441 063406  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9442 063410                              NEXT.ERRNO
9443 063410                      282$:   ERRDF   ERRNO,T202SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063410  104455                                                              TRAP    C$ERDF
     063412  001620                                                              .WORD   912
     063414  064656                                                              .WORD   T202SSR
     063416  012106                                                              .WORD   PKTSSR
9444 063420  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
9445 063424                      290$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063424  104406                                                              TRAP    C$CLP1
9446                                     ;       Do a Write Subsystem READ STATUS
9447                                     ;       If Read Data parity error NOT=0  Then Print Error
9448 063426  004737  065744              JSR     PC,T20SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
9449 063432  012701  064472              MOV     #T20EXSTA,R1            ;GET EXPECTED READ STATUS
9450 063436  012702  066032              MOV     #T20BFSTA,R2            ;GET RECV READ STATUS
9451 063442  011211                      MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
9452 063444  016261  000002  000002      MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTED)
9453 063452  042711  100000              BIC     #S1.PARERR,(R1)         ;SET EXP PAR ERR  =0
9454 063456  005000                      CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
9455 063460  012701  066012              MOV     #T20BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
9456 063464  012702  064452              MOV     #T20EXP,R2              ;EXPD ADDRESS
9457 063470  012703  000024              MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
9458 063474  004737  011540              JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
9459 063500                              FORCERROR       302$,NOTSSR             ;@@D
9460 063510  103404                      BCS     320$                    ;BR IF YES
9461 063512                              NEXT.ERRNO
9462 063512                      302$:   ERRHRD  ERRNO,T20RSF,MSGSTAT    ;REPORT ERROR
     063512  104456                                                              TRAP    C$ERHRD
     063514  001621                                                              .WORD   913
     063516  065316                                                              .WORD   T20RSF
     063520  012410                                                              .WORD   MSGSTAT
9463 063522                      320$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063522  104406                                                              TRAP    C$CLP1
9464                                     ;       (* Verify Data can be transferred without a Parity Error *)
9465                                     ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
9466 063524  012700  000001              MOV     #^F.I4RES,R0            ;IRESV4==>IRSTR = 1
9467 063530  004737  065652              JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
9468 063534  012704  066140              MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
9469 063540  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
9470 063544  004737  016376              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
9471 063550                              FORCERROR       332$                    ;@@FORCE ERROR IF FORCER=1
9472 063564  103407                      BCS     340$                    ;BR IF CARRY SET (GOOD RETURN)
9473 063566  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9474 063570                              NEXT.ERRNO
9475 063570                      332$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063570  104455                                                              TRAP    C$ERDF
     063572  001622                                                              .WORD   914
     063574  065141                                                              .WORD   T208SSR
     063576  012106                                                              .WORD   PKTSSR
9476 063600  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
9477 063604                      340$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063604  104406                                                              TRAP    C$CLP1
```

TEST 9: READ/WRITE DATA PARITY TEST

```
9478                                    ;       Do a WRITE NPR to set loopback and tape direction OUT and
9479                                    ;               and CLEAR Write Wrong Parity.
9480 063606  012700  000100                        MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
9481 063612  052700  000040                        BIS     #NP.LOOP,R0             ;SET LOOPBACK
9482 063616  052700  000020                        BIS     #NP.WRP,R0              ;CLEAR WRITE WRONG PARITY (INVERTED)
9483 063622  004737  065556                        JSR     PC,T20WNPR              ;SETUP T20PK2 FOR WRITE NPR
9484 063626  012704  066140                        MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
9485 063632  010465  000000                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
9486 063636  004737  016376                        JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
9487 063642                                        FORCERROR       352$                    ;aaDFORCE ERROR IF FORCER=1
9488 063656  103407                                BCS     360$                    ;BR IF CARRY SET (GOOD RETURN)
9489 063660  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9490 063662                                        NEXT.ERRNO
9491 063662                             352$:      ERRDF   ERRNO,T204SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063662  104455                                                                        TRAP    C$ERDF
     063664  001623                                                                      .WORD   915
     063666  064767                                                                      .WORD   T204SSR
     063670  012106                                                                      .WORD   PKTSSR
9492 063672  005237  002222                        INC     FATFLG                  ;SET FATAL ERROR FLAG
9493 063676                             360$:      CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063676  104406                                                                        TRAP    C$CLP1
9494                                    ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
9495 063700  012700  000001                        MOV     #1,R0                   ;WRITE 1 BYTE
9496 063704  012701  002314                        MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
9497 063710  004737  065616                        JSR     PC,T20WFIF              ;SETUP T20PK2 FOR WRITE FIFO
9498 063714  012704  066140                        MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
9499 063720  010465  000000                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
9500 063724  004737  016376                        JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
9501 063730                                        FORCERROR       372$                    ;aaDFORCE ERROR IF FORCER=1
9502 063744  103407                                BCS     380$                    ;BR IF CARRY SET (GOOD RETURN)
9503 063746  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9504 063750                                        NEXT.ERRNO
9505 063750                             372$:      ERRDF   ERRNO,T205SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063750  104455                                                                        TRAP    C$ERDF
     063752  001624                                                                      .WORD   916
     063754  065032                                                                      .WORD   T205SSR
     063756  012106                                                                      .WORD   PKTSSR
9506 063760  005237  002222                        INC     FATFLG                  ;SET FATAL ERROR FLAG
9507 063764                             380$:      CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063764  104406                                                                        TRAP    C$CLP1
9508                                    ;       Do a READ FIFO with tape direction OUT to load tape out write latch
9509 063766  012700  000001                        MOV     #1,R0                   ;SET READ BYTE COUNT
9510 063772  004737  065576                        JSR     PC,T20RFIF              ;SETUP T20PK2 FOR READ FIFO
9511 063776  012704  066140                        MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
9512 064002  010465  000000                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
9513 064006  004737  016376                        JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
9514 064012                                        FORCERROR       392$                    ;aaDFORCE ERROR IF FORCER=1
9515 064026  103407                                BCS     400$                    ;BR IF CARRY SET (GOOD RETURN)
9516 064030  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
9517 064032                                        NEXT.ERRNO
9518 064032                             392$:      ERRDF   ERRNO,T206SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064032  104455                                                                        TRAP    C$ERDF
     064034  001625                                                                      .WORD   917
     064036  065076                                                                      .WORD   T206SSR
     064040  012106                                                                      .WORD   PKTSSR
9519 064042  005237  002222                        INC     FATFLG                  ;SET FATAL ERROR FLAG
9520 064046                             400$:      CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

TEST 9: READ/WRITE DATA PARITY TEST

```
        064046  104406                                                                  TRAP    C$CLP1
 9521                           ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 0 (sets read strobe low)
 9522                           ;               (Read Strobe sets PAR IN H [Parity Error])
 9523 064050  005000            CLR     R0                      ;IRESV4==>IRSTR = 0
 9524 064052  004737  065652    JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
 9525 064056  012704  066140    MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9526 064062  010465  000000    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9527 064066  004737  016376    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9528 064072                    FORCERROR       412$                    ;@@FORCE ERROR IF FORCER=1
 9529 064106  103407            BCS     420$                    ;BR IF CARRY SET (GOOD RETURN)
 9530 064110  010001            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9531 064112                    NEXT.ERRNO
 9532 064112                    412$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        064112  104455                                                                  TRAP    C$ERDF
        064114  001626                                                                  .WORD   918
        064116  065141                                                                  .WORD   T208SSR
        064120  012106                                                                  .WORD   PKTSSR
 9533 064122  005237  002222    INC     FATFLG                  ;SET FATAL ERROR FLAG
 9534 064126                    420$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
        064126  104406                                                                  TRAP    C$CLP1
 9535                           ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
 9536 064130  012700  000001    MOV     @WF.I4RES,R0            ;IRESV4==>IRSTR = 1
 9537 064134  004737  065652    JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
 9538 064140  012704  066140    MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9539 064144  010465  000000    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9540 064150  004737  016376    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9541 064154                    FORCERROR       432$                    ;@@FORCE ERROR IF FORCER=1
 9542 064170  103407            B_S     440$                    ;BR IF CARRY SET (GOOD RETURN)
 9543 064172  010001            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9544 064174                    NEXT.ERRNO
 9545 064174                    432$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        064174  104455                                                                  TRAP    C$ERDF
        064176  001627                                                                  .WORD   919
        064200  065141                                                                  .WORD   T208SSR
        064202  012106                                                                  .WORD   PKTSSR
 9546 064204  005237  002222    INC     FATFLG                  ;SET FATAL ERROR FLAG
 9547 064210                    440$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
        064210  104406                                                                  TRAP    C$CLP1
 9548
 9549                           ;       Do a Write Subsystem READ STATUS
 9550 064212  004737  065536    JSR     PC,T20SRD               ;SETUP PACKET FOR READ STATUS
 9551 064216  012704  066140    MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 9552 064222  010465  000000    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 9553 064226  004737  016376    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9554 064232                    FOPCERROR       452$                    ;@@FORCE ERROR IF FORCER=1
 9555 064246  103407            BCS     460$                    ;BR IF CARRY SET (GOOD RETURN)
 9556 064250  010001            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9557 064252                    NEXT.ERRNO
 9558 064252                    452$:   ERRDF   ERRNO,T203SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        064252  104455                                                                  TRAP    C$ERDF
        064254  001630                                                                  .WORD   920
        064256  064722                                                                  .WORD   T203SSR
        064260  012106                                                                  .WORD   PKTSSR
 9559 064262  005237  002222    INC     FATFLG                  ;SET FATAL ERROR FLAG
 9560 064266                    460$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
        064266  104406                                                                  TRAP    C$CLP1
 9561                           ;       If Read Data parity error NOT=0  Then Print Error
```

TEST 9: READ/WRITE DATA PARITY TEST

```
 9562 064270  004737  065744                JSR     PC,T20SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
 9563 064274  012701  064472                MOV     #T20EXSTA,R1            ;GET EXPECTED READ STATUS
 9564 064300  012702  066032                MOV     #T20BFSTA,R2            ;GET RECV READ STATUS
 9565 064304  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
 9566 064306  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTED)
 9567 064314  042711  100000                BIC     #S1.PARERR,(R1)         ;SET EXP PAR ERR  =0
 9568 064320  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 9569 064322  012701  066012                MOV     #T20BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 9570 064326  012702  064452                MOV     #T20EXP,R2              ;EXPD ADDRESS
 9571 064332  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 9572 064336  004737  011540                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 9573 064342                                FORCERROR       472$,NOTSSR     ;a@D
 9574 064352  103404                        BCS     480$                    ;BR IF YES
 9575 064354                                NEXT.ERRNO
 9576 064354                        472$:   ERRHRD  ERRNO,T20CWP,MSGSTAT    ;REPORT ERROR
      064354  104456                                                                TRAP    C$ERHRD
      064356  001631                                                                .WORD   921
      064360  065417                                                                .WORD   T20CWP
      064362  012410                                                                .WORD   MSGSTAT
 9577 064364                        480$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      064364  104406                                                                TRAP    C$CLP1
 9578
 9579 064366                                FORCEXIT        555$            ;a@D
 9580 064376  013703  002320                MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
 9581 064402  020327  003064                CMP     R3,#TBLEND              ;END OF TSTBLK?
 9582 064406  103002                        BHIS    555$                    ;BR IF YES
 9583 064410  000137  062464                JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
 9584 064414                        555$:
 9585
 9586 064414                                ENDSUB                          ;////////////// END SUBTEST //////////
      064414                                                                        L10074:
      064414  104403                                                                TRAP    C$ESUB
 9587
 9588 064416  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
 9589 064422  001402                        BEQ     560$                    ;BRANCH IF NOT
 9590 064424  004737  017242                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 9591 064430                        560$:
 9592 064430  004737  016516                JSR     PC,TSTLOOP              ;DO ITERATIONS?
 9593 064434  103002                        BCC     565$                    ;BR IF NO
 9594 064436  000137  050116                JMP     T18LOOP                 ;LOOP UNTIL ITERATIONS DONE
 9595 064442                        565$:
 9596 064442                                EXIT    TST                     ;////////// EXIT TEST //////////
      064442  104432                                                                TRAP    C$EXIT
      064444  001610                                                                .WORD   L10073-.
 9597
 9598
 9599                                       ;+
 9600                                       ;LOCAL STORAGE FOR THIS TEST
 9601                                       ;-
 9602
 9603
 9604 064446                        T20MSK:                                 ;MASK OF UNTESTED BITS IN READ STATUS
 9605                                                                       ;UNTESTED BITS ARE SET TO 1
 9606 064446      377                               .BYTE   ^C<000>         ;BYTE 0 MASK
 9607 064447      037                               .BYTE   ^C<340>         ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
 9608 064450      360                               .BYTE   ^C<017>         ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
 9609 064451      000                               .BYTE   0               ;MAKE IT EVEN
```

C2

TEST 9: READ/WRITE DATA PARITY TEST

```
9610
9611 064452                            T20EXP:                                 ;BEGIN EXPECTED DATA BUFFER
9612 064452  000000                             .WORD   0                      ;MESSAGE TYPE
9613 064454  000000                             .WORD   0                      ;DATA FIELD LENGTH
9614 064456  000000                             .WORD   0                      ;RBPCR
9615 064460  000000                             .WORD   0                      ;XST0
9616 064462  000000                             .WORD   0                      ;XST1
9617 064464  000000                             .WORD   0                      ;XST2
9618 064466  000000                             .WORD   0                      ;XST3
9619 064470  000000                             .WORD   0                      ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
9620 064472                            T20EXSTA: .BLKB 64.                     ;EXPECTED READ STATUS AND WRITE FIFO DATA
9621 064572                            T20EXEND:                                       ;END EXPECTED DATA BUFFER
9622                                   ;*
9623                                   ;LOCAL TEXT MESSAGES FOR TEST
9624                                   ;-
9625
9626 064572     122     145     141   TST20ID:        .ASCIZ  'Read/Write Data Parity'
9627 064621     127     122     111   T20SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
9628 064656     127     122     111   T202SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
9629 064722     127     122     111   T203SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
9630 064767     127     122     111   T204SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
9631 065032     127     122     111   T205SSR:.ASCIZ  'WRITE SUBSYSTEM (Write FIFO) Failed'
9632 065076     127     122     111   T206SSR:.ASCIZ  'WRITE SUBSYSTEM (Read FIFO) Failed'
9633 065141     127     122     111   T208SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Format) Failed'
9634 065207     122     145     141   T20SWP: .ASCIZ  'Read Data Parity Error (PARERR) Failed to Set after Write Wrong Parity'
9635 065316     122     145     141   T20RSF: .ASCIZ  'Read Data Parity Error (PARERR) Failed to Clear after RESET FIFO'
9636 065417     122     145     141   T20CWP: .ASCIZ  'Read Data Parity Error (PARERR) occurred in Data Loopback'
9637                                           .EVEN
9638
9639                                   ;*
9640                                   ; CLEAR MESSAGE BUFFER
9641                                   ;-
9642 065512                            T20CLRBUF:                              ;
9643 065512                                    SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
9644 065516  012701  066012                     MOV     #T20BFR,R1             ;GET MESSAGE BUFFER ADDRESS
9645 065522  012702  000120                     MOV     #T20BEND-T20BFR,R2     ;SIZE OF MESSAGE BUFFER IN BYTES
9646 065526  105021                    10$:     CLRB    (R1)+                  ;CLEAR A BYTE
9647 065530  005302                             DEC     R2                     ;DONE?
9648 065532  003375                             BGT     10$                    ;BR IF NO
9649 065534  000207                             RTS     PC                     ;RETURN
9650
9651                                   ;*
9652                                   ; SETUP T20PK2 PACKET FOR READ STATUS
9653                                   ;-
9654 065536                            T20SRD:
9655 065536  004737  065512                     JSR     PC,T20CLRBUF           ;CLEAR MESSAGE BUFFER
9656 065542  012700  066150                     MOV     #T20DT2,R0             ;WRITE SUBSYSTEM DATA BUFFER
9657 065546  112720  000005                     MOVB    #PW.RDSTATUS,(R0)+     ;STORE READ STATUS COMMAND IN BSEL0
9658 065552  105010                             CLRB    (R0)                   ;CLEAR BSEL1
9659 065554  000207                             RTS     PC                     ;RETURN
9660
9661
9662                                   ;*
9663                                   ; SETUP T20PK2 PACKET FOR WRITE NPR
9664                                   ;
9665                                   ; INPUT:
9666                                   ;       R0 CONTAINS BSEL1 NPR DATA
```

TEST 9: READ/WRITE DATA PARITY TEST

```
 9667                                   ;
 9668                                   ;-
 9669 065556                            T20WNPR:
 9670 065556  004737  065512                       JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
 9671 065562  012701  066150                       MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
 9672 065566  112721  000011                       MOVB    #PW.WNPR,(R1)+          ;STORE WRITE NPR IN BSEL0
 9673 065572  110011                               MOVB    R0,(R1)                 ;STORE NPR DATA IN BSEL1
 9674 065574  000207                               RTS     PC                      ;RETURN
 9675
 9676                                   ;+
 9677                                   ; SETUP T20PK2 PACKET FOR READ FIFO
 9678                                   ;
 9679                                   ; INPUT:
 9680                                   ;       R0 CONTAINS SEL2 BYTE COUNT
 9681                                   ;-
 9682 065576                            T20RFIF:
 9683 065576  004737  065512                       JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
 9684 065602  012701  066150                       MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
 9685 065606  112721  000003                       MOVB    #PW.RFIFO,(R1)+         ;STORE READ FIFO IN BSEL0
 9686 065612  110021                               MOVB    R0,(R1)+                ;STORE BYTE COUNT IN BSEL1
 9687 065614  000207                               RTS     PC                      ;RETURN
 9688
 9689                                   ;+
 9690                                   ; SETUP T20PK2 PACKET FOR WRITE FIFO
 9691                                   ;
 9692                                   ; INPUT:
 9693                                   ;       R0 CONTAINS BYTE COUNT
 9694                                   ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
 9695 065616                            ;-
 9696 065616                            T20WFIF:
                                                SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
 9697 065622  004737  065512                       JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
 9698 065626  012702  066150                       MOV     #T20DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
 9699 065632  112722  000004                       MOVB    #PW.WFIFO,(R2)+         ;STORE WRITE FIFO IN BSEL0
 9700 065636  110022                               MOVB    R0,(R2)+                ;STORE BYTE COUNT IN BSEL1
 9701 065640  005022                               CLR     (R2)+                   ;CLEAR SEL2 (UNUSED)
 9702 065642  112122                    10$:       MOVB    (R1)+,(R2)+             ;STORE DATA PATTERN BYTE
 9703 065644  005300                               DEC     R0                      ;DONE ALL BYTES?
 9704 065646  003375                               BGT     10$                     ;BR IF NO
 9705 065650  000207                               RTS     PC                      ;RETURN
 9706
 9707                                   ;+
 9708                                   ; SETUP T20PK2 FOR WRITE FORMAT TRANSPORT REGISTER
 9709                                   ;
 9710                                   ; INPUT:
 9711                                   ;       R0 CONTAINS DRIVING DATA PATTERN
 9712                                   ;-
 9713 065652                            T20WFMT:
 9714 065652  004737  065512                       JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
 9715 065656  012701  066150                       MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
 9716 065662  112721  000007                       MOVB    #PW.WFMT,(R1)+          ;STORE WRITE FORMAT IN BSEL0
 9717 065666  110021                               MOVB    R0,(R1)+                ;STORE DATA WORD IN BSEL1
 9718 065670  000207                               RTS     PC                      ;RETURN
 9719
 9720                                   ;+
 9721                                   ; SETUP T20PK2 PACKET FOR WRITE MISC.
 9722                                   ;
 9723                                   ;       R0 CONTAINS WRITE MISC DATA
                                        ;-
```

E2

TEST 9: RFAD/WRITE DATA PARITY TEST

```
 9724 065672
 9725 065672  012701  066150          T20WMISC:
 9726 065676  112721  000010                MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
 9727 065702  110011                        MOVB    #PW.WMISC,(R1)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
 9728 065704  000207                        MOVB    R0,(R1)                 ;STORE INVERT EXTENDED FEATURES IN BSEL1
 9729                                       RTS     PC                      ;RETURN
 9730                                ;+
 9731                                ; SETUP T20PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
 9732 065706                        ;-
 9733 065706  012700  066150          T20SEXT:
 9734 065712  112720  000010                MOV     #T20DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
 9735 065716  112710  000200                MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
 9736 065722  000207                        MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
 9737                                       RTS     PC                      ;RETURN
 9738                                ;+
 9739                                ; CLEAR EXPECTED DATA MESSAGE BUFFER
 9740 065724                        ;-
 9741 065724  012701  064452          T20CLEXP:
 9742 065730  012700  000120                MOV     #T20EXP,R1              ;GET EXPD ADDRESS
 9743 065734  105021                        MOV     #T20EXEND-T20EXP,R0     ;GET EXPD SIZE
 9744 065736  005300          10$:          CLRB    (R1)+                   ;CLEAR A BYTE
 9745 065740  003375                        DEC     R0                      ;DONE?
 9746 065742  000207                        BGT     10$                     ;BR IF NO
 9747                                       RTS     PC                      ;RETURN
 9748                                ;+
 9749                                ;Set WORDS 0-7 of expd message BUFFER = to recv since not testing
 9750                                ;-
 9751 065744                          T20SETEXP:
 9752 065744  012702  064452                MOV     #T20EXP,R2              ;GET EXPD
 9753 065750  012703  066012                MOV     #T20BFR,R3              ;GET READ STATUS RECV BUFFER
 9754 065754  012700  000010                MOV     #8.,R0                  ;SET WORDS 0-7 EXP=RECV
 9755 065760  012322          5$:           MOV     (R3)+,(R2)+             ;SET EXPD=RECV
 9756 065762  005300                        DEC     R0                      ;DONE WORDS 0-7 WORDS?
 9757 065764  003375                        BGT     5$                      ;BR IF NO
 9758 065766  000207                        RTS     PC                      ;RETURN
 9759
 9760
 9761
 9765                                ;
 9766                                ;WRITE CHARACTERISTICS COMMAND PACKET
 9767                                ;
 9768 065770                          T20PACKET:                              ;COMMAND PACKET FOR TEST
 9769 065770  100004                        .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
 9770 065772  066000                        .WORD   T20DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
 9771 065774  000000                        .WORD   0
 9772 065776  000012                        .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
 9773
 9774 066000                          T20DATA:                                ;CHARACTERISTICS DATA BLOCK
 9775 066000  066012                        .WORD   T20BFR                  ;ADDRESS OF MESSAGE BUFFER
 9776 066002  000000                        .WORD   0
 9777 066004  000024                        .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
 9778 066006  000000                        .WORD   0                       ;ESS,ENB,EAI,ERI
 9779 066010  000007                        .WORD   7                       ;EXTENDED FEATURES UNIT NO.
 9780
 9781
 9782                                ;MESSAGE BUFFER FOR ALL TEST 17 COMMANDS
 9783
```

F2


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

 9784 066012                            T20BFR:                         ;BEGIN MESSAGE BUFFER
 9785 066012  000000                            .WORD   0               ;MESSAGE TYPE
 9786 066014  000000                            .WORD   0               ;DATA FIELD LENGTH
 9787 066016  000000                            .WORD   0               ;RBPCR
 9788 066020  000000                            .WORD   0               ;XST0
 9789 066022  000000                            .WORD   0               ;XST1
 9790 066024  000000                            .WORD   0               ;XST2
 9791 066026  000000                            .WORD   0               ;XST3
 9792 066030  000000                            .WORD   0               ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
 9793 066032                            T20BFSTA: .BLKB 64.             ;READ STATUS AND WRITE FIFO BUFFER
 9794 066132                            T20BEND:                        ;END OF MESSAGE BUFFER
 9795                                   ;
 9796                                   ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
 9797                                   ;
 9799          066140                           .=<.+10>&177770
 9801 066140                            T20PK2:
 9802 066140  100006                            .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
 9803 066142  066150                            .WORD   T20DT2                  ;LOW ADDRESS OF DATA BLOCK
 9804 066144  000000                            .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
 9805 066146  000012                            .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
 9806
 9807 066150                            T20DT2:                         ;DATA BLOCK
 9808 066150     000                            .BYTE   0               ;BSEL0
 9809 066151     000                            .BYTE   0               ;BSEL1
 9810 066152  000000                            .WORD   0               ;SEL2
 9811 066154                                    .BLKB   64.             ;WRITE FIFO DATA OUTPUT BUFFER
 9812
 9813
 9814 066254                                   ENDTST
      066254                                                                    L10073:
      066254  104401                                                                   TRAP    C$ETST
 9815                                           .SBTTL  TEST 10: MANUAL INTERVENTION
 9816
 9817                                   ;THE MANUAL INTERVENTION TEST IS A STANDALONE ROUTINE (NOT REALLY A "TEST")
 9818                                   ;THAT ALLOWS THE OPERATOR TO CHECK OUT VARIOUS ELEMENTS AND FUNCTIONS OF
 9819                                   ;THE SUBSYSTEM THAT CANNOT BE MANIPULATED BY THE PROGRAM ALONE. WHEN
 9820                                   ;THIS ROUTINE IS STARTED, IT FIRST PRINTS OUT A MENU OF SELECTABLE
 9821                                   ;SUBTESTS AND THEN WAITS FOR THE OPERATOR TO TYPE IN A SELECTION CODE.
 9822                                   ;THE ONLY WAYS TO EXIT THIS ROUTINE AND RETURN TO THE DIAGNOSTIC SUPERVISOR
 9823                                   ;ARE BY TYPING <CTRL-C> OR SELECTING CODE 7.
 9824                                   ;SELECTION CODES AND SUBROUTINES ARE:
 9825                                   ;
 9826                                   ;
 9827                                   ;       CODE    ROUTINE
 9828                                   ;
 9829                                   ;         0     HELP. PRINTS THIS MENU.
 9830                                   ;         1     TURN ON ALL M7196 LED INDICATORS
 9831                                   ;         2     TURN OFF ALL M7196 LED INDICATORS
 9832                                   ;         3     OFFLINE/ONLINE ATTENTION TEST
 9833                                   ;         4     WRITE-PROTECT TEST
 9834                                   ;         5     INITIATE TRANSPORT SERVO EXERCISER
 9835                                   ;         6     PRINT EXTENDED TRANSPORT STATUS
 9836                                   ;         7     EXIT (RETURN TO SUPERVISOR)
 9837                                   ;
 9838                                   ;
 9839                                   ;EACH MENU ITEM CORRESPONDS TO A SUBTEST, AS FOLLOWS:
 9840                                   ;
```

```
9841                                    ;
9842                                    ;
9843                                    ;PRINTS OUT THE MENU ON THE CONSOLE TERMINAL.
9844                                    ;
9845                                    ;
9846                                    ;CAUSES ALL THREE LED INDICATORS ON THE M7196 MODULE
9847                                    ;TO BE ILLUMINATED. AFTER INITIATING THIS ROUTINE, THE OPERATOR
9848                                    ;SHOULD OBSERVE THE LED'S AND VERIFY THAT THEY ARE INDEED ALL LIT.
9849                                    ;THIS ROUTINE FIRST USES THE WRITE SUBSYSTEM MEMORY COMMAND TO
9850                                    ;SET THE FORCE WRONG PARITY FLIP-FLOP, WHICH SERVES TO DRIVE THE
9851                                    ;"PROCESSOR NOT OK" LED. THEN IT ENTERS A LOOP THAT CONTINUALLY
9852                                    ;WRITES THE LOW BYTE OF TSDB AND READS THE TSSR. THESE LATTER TWO
9853                                    ;OPERATIONS WILL CAUSE THE "NOT SSR" AND "DRIVING BUS" LED'S TO
9854                                    ;GLOW -- THEY ARE NOT REALLY LIT AT ALL TIMES BUT SHOULD APPEAR
9855                                    ;REASONABLY VISIBLE.
9856                                    ;
9857                                    ;
9858                                    ;INITIALIZES THE CONTROLLER TO CAUSE ALL LED'S TO
9859                                    ;EXTINGUISH.
9860                                    ;
9861                                    ;
9862                                    ;
9863                                    ;
9864                                    ;THIS ROUTINE INITIALIZES THE CONTROLLER, ISSUES A
9865                                    ;WRITE CHARACTERISTICS COMMAND TO ENABLE ATTENTION INTERRUPTS,
9866                                    ;ISSUES A MESSAGE BUFFER RELEASE COMMAND, PRINTS A MESSAGE ON THE
9867                                    ;CONSOLE TERMINAL INSTRUCTING THE OPERATOR TO TOGGLE THE ON-LINE
9868                                    ;SWITCH ON THE TRANSPORT, THEN WAITS FOR AN ATTENTION INTERRUPT.
9869                                    ;EACH TIME THE TRANSPORT TRANSITIONS FROM ON-LINE TO OFF-LINE OR
9870                                    ;VICE-VERSA, AN ATTENTION INTERRUPT SHOULD BE GENERATED. THE PROGRAM
9871                                    ;WILL REPORT THE INTERRUPT AND THE CURRENT STATE OF THE TRANSPORT.
9872                                    ;THE OPERATOR SHOULD VERIFY THAT THE REPORTED STATE MATCHES THE
9873                                    ;STATE INDICATED BY THE LED ON THE FRONT PANEL OF THE TRANSPORT.
9874                                    ;IN ADDITION, WHEN THE TRANSPORT IS PLACED OFF-LINE, THE PROGRAM
9875                                    ;ISSUES A SEQUENCE OF TAPE-MOTION COMMANDS (READ, WRITE, POSITION, ETC.
9876                                    ;AND VERIFIES THAT, FOR EACH COMMAND, FUNCTION REJECT TERMINATION
9877                                    ;RESULTS, ALONG WITH THE NON-EXECUTABLE FUNCTION (NEF) ERROR BIT BEING
9878                                    ;SET.
9879                                    ;
9880                                    ;THIS ROUTINE INSTRUCTS THE OPERATOR TO MOUNT A SCRATCH
9881                                    ;TAPE REEL THAT DOES NOT HAVE A WRITE-ENABLE RING INSTALLED, THEN
9882                                    ;WAITS FOR THE OPERATOR TO RESPOND THAT THIS HAS BEEN ACCOMPLISHED.
9883                                    ;UPON THE RESPONSE, THE PROGRAM VERIFIES THAT THE TRANSPORT SHOWS
9884                                    ;A WRITE-PROTECTED STATUS, THEN ATTEMPTS TO WRITE DATA ON THE
9885                                    ;TAPE AND EXPECTS THE APPROPRIATE ERROR TERMINATION INDICATING THAT
9886                                    ;THE WRITE FUNCTION COULD NOT BE PERFORMED BECAUSE THE REEL IS
9887                                    ;WRITE-PROTECTED.  IF THE APPROPRIATE TERMINATION IS NOT RECEIVED,
9888                                    ;AN ERROR IS REPORTED.
9889                                    ;
9890                                    ;
9891                                    ;
9892                                    ;INSTRUCTS THE OPERATOR TO PLACE THE TAPE TRANSPORT(S)
9893                                    ;ON-LINE (IF ANY ARE OFF-LINE) THEN ATTEMPTS TO PERFORM AN EXTENDED
9894                                    ;STATUS READOUT.  FOR EACH TRANSPORT EQUIPPED WITH THIS FEATURE,
9895                                    ;THE PROGRAM FORMATS AND PRINTS OUT THE RESULTING STATUS. IF THE
9896                                    ;TRANSPORT IS NOT EQUIPPED WITH THIS FEATURE, A MESSAGE INDICATING
9897                                    ;SUCH IS ISSUED.
```

H2

TEST 10: MANUAL INTERVENTION

```
9898                                    ;
9899                                    ;
9900                                    ;
9901                                    ;
9902 066256                                     BGNTST
     066256                                                                  T10::
9907 066256                                     RFLAGS  R0              ;GET OPERATOR FLAGS
     066256 104421                                                              TRAP    C$RFLA
9908 066260 001403                              BEQ     21$             ;BR, IF OK TO RUN
9909 066262 012700 071630                       MOV     #T38NE,R0       ;"TEST NOT EXECUTED"
9910 066266 000402                              BR      3$              ;JUMP IF NOT FIRST TEST
9911 066270                             21$:
9912 066270 012700 072745                       MOV     #T38ID,R0       ;TEST ID MESSAGE
9913 066274 004737 016550               3$:     JSR     PC,TSTSETUP     ;DO THE COMMON SETUP
9914 066300 004737 020022                       JSR     PC,CHKMAN       ;IS MANUAL INTERVENTION ALLOWED?
9915 066304 103402                              BCS     22$             ;BR, IF MANUAL INTER ALLOWED
9916 066306 000137 071032                       JMP     64$             ;JUMP IF NOT ALLOWED
9917 066312                             22$:
9921 066312 005037 002222               2$:     CLR     FATFLG          ;CLEAR THE FATAL ERROR FLAG
9922 066316 012737 176750 071044                MOV     #65000.,T38DLY  ;SET UP DELAY COUNTER
9923 066324 004737 016034               5$:     JSR     PC,SOFINIT      ;DO A SOFT INIT
9924 066330 103427                              BCS     23$             ;BRANCH IF OK
9925 066332 010001                              MOV     R0,R1           ;CONTENTS OF TSSR REGISTER
9926 066334 032701 000200                       BIT     #SSR,R1         ;CHECK FOR TSSR SET
9927 066340 001023                              BNE     23$             ;KEEP GOING IF NOT SET
9928 066342                                     DELAY   250             ;CALL DELAY ROUTINE
     066342 012727 000250                                                       MOV     #250,(PC)+
     066346 000000                                                              .WORD   0
     066350 013727 002116                                                       MOV     L$DLY,(PC)+
     066354 000000                                                              .WORD   0
     066356 005367 177772                                                       DEC     -6(PC)
     066362 001375                                                              BNE     .-4
     066364 005367 177756                                                       DEC     -22(PC)
     066370 001367                                                              BNE     .-20
9929 066372 005337 071044                       DEC     T38DLY          ;BUMP COUNTER DOWN
9930 066376 001352                              BNE     5$              ;BR, IF MORE TIME LEFT
9931 066400                                     ERRDF   ERRNO,SFIERR,SFIMSG     ;REPORT FATAL ERROR
     066400 104455                                                              TRAP    C$ERDF
     066402 001751                                                              .WORD   1001
     066404 003654                                                              .WORD   SFIERR
     066406 012074                                                              .WORD   SFIMSG
9932 066410 012700 072772               23$:    MOV     #MIMENU,R0      ;MENU OF MANUAL INTERVENTIONS
9933 066414 012701 000005                       MOV     #5,R1           ;MAXIMUM ALLOWED SELECTION
9934 066420 004737 017600                       JSR     PC,GETSEL       ;GO GET THE OPERATORS SELECTION
9935 066424 010004                              MOV     R0,R4           ;GET NUMBER FROM ROUTINE
9936 066426 006304                              ASL     R4              ;CONVERT TO WORD OFFSET
9937 066430 000174 066434                       JMP     @6$(R4)         ;JUMP TO PROPER LOOP
9938 066434 066312                      6$:     .WORD   2$              ;RETYPE THE MENU
9939 066436 066450                              .WORD   10$             ; 1 TURN ON LED'S
9940 066440 066732                              .WORD   15$             ; 2 TURN OFF LED'S
9941 066442 067164                              .WORD   20$             ; 3 ONLINE ATTENTION
9942 066444 067620                              .WORD   25$             ; 4 WRITE PROTECT
9943 066446 070554                              .WORD   35$             ; 5 EXTENDED TRANSPORT STATUS
9944 066450                             10$:    PRINTF  #T38MS2         ;TELL OPERATOR TO CNTRL-C FOR EXIT
     066450 012746 072641                                                       MOV     #T38MS2,-(SP)
     066454 012746 000001                                                       MOV     #1,-(SP)
     066460 010600                                                              MOV     SP,R0
```

TEST 10: MANUAL INTERVENTION

```
                066462  104417                                                          TRAP    C$PNTF
                066464  062706  000004                                                  ADD     #4,SP
 9945           066470  004737  073402                  JSR     PC,T38REST              ;SET PACKET TO INITIAL VALUES
 9946           066474  004737  016034                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 9947           066500  103405                          BCS     100$                    ;BR IF SOFT INIT = OK
 9951           066502  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9952           066504                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
                066504  104455                                                          TRAP    C$ERDF
                066506  001752                                                          .WORD   1002
                066510  003654                                                          .WORD   SFIERR
                066512  012074                                                          .WORD   SFIMSG
 9953           066514  013737  002202  071570  100$:   MOV     UNITN,T38DSW            ;SET UNIT NUMBER
 9954
 9955           066522  012704  071550                  MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
 9956           066526  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 9957           066532  103405                          BCS     110$                    ;BR, IF COMMAND ISSUED OK
 9961           066534  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9962           066536                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
                066536  104456                                                          TRAP    C$ERHRD
                066540  001753                                                          .WORD   1003
                066542  005060                                                          .WORD   WRTMSG
                066544  012074                                                          .WORD   SFIMSG
 9963           066546                          110$:
 9964           066546  112737  000000  071061          MOVB    #0,T38BS1               ;CLEAR BIT #4
 9965           066554  112737  000011  071060          MOVB    #11,T38BSO              ;WRITE MISC COMMAND
 9966           066562  012704  071050                  MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
 9967                                    ;
 9968                                    ;NOTE: THIS COMMAND TURNS ON THE PROCESSOR FAIL LED
 9969                                    ;
 9970           066566  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 9971           066572  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 9972           066576  103405                          BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
 9973           066600  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 9977           066602                                  ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                066602  104455                                                          TRAP    C$ERDF
                066604  001754                                                          .WORD   1004
                066606  072246                                                          .WORD   T38SSR
                066610  012106                                                          .WORD   PKTSSR
 9978           066612                          150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                066612  104406                                                          TRAP    C$CLP1
 9979           066614                                  SETPRI  #PRI07                  ;RAISE THE PRIORITY
                066614  012700  000340                                                  MOV     #PRI07,R0
                066620  104441                                                          TRAP    C$SPRI
 9980           066622  005037  071036                  CLR     TTION2                  ;ASSUME INTERRUPTS ARE ENABLED
 9981           066626  032737  000100  177560          BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
 9982           066634  001005                          BNE     701$                    ;BRANCH IF YES
 9983           066636  005237  071036                  INC     TTION2                  ;FLAG SET IF INTERRUPTS OFF
 9984           066642  052737  000100  177560          BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
 9985           066650  012701  000060          701$:   MOV     #TTIVEC,R1              ;START OF TTI VECTORS
 9986           066654  011137  071040                  MOV     (R1),TVSAV2             ;SAVE THE CURRENT TTI VECTOR
 9987           066660  012721  070336                  MOV     #590$,(R1)+             ;SET NEW INTERRUPT ROUTINE
 9988           066664  011137  071042                  MOV     (R1),TPSAV2             ;SAVE THE VECTOR PRIORITY
 9989           066670  012711  000340                  MOV     #PRI07,(R1)             ;USE PRIORITY SEVEN
 9990           066674                                  SETPRI  #PRI00                  ;LOWER INTERRUPT BR LEVEL
                066674  012700  000000                                                  MOV     #PRI00,R0
                066700  104441                                                          TRAP    C$SPRI
 9991           066702  012701  177777                  MOV     #-1,R1                  ;DATA TO WRITE TO TSDB
```

J2

TEST 10: MANUAL INTERVENTION

```
 9992 066706 000240                 12$:    NOP                         ;ALLOW OPERATOR TO TYPE ↑C
 9993 066710 012702 001750                  MOV     #1000.,R2           ;SET UP INNER LOOP
 9994 066714 110165 000000          14$:    MOVB    R1,TSDB(R5)         ;WRITE DATA TO TSDB
 9995 066720 016500 000002                  MOV     TSSR(R5),R0         ;READ TSSR
 9996 066724 005302                         DEC     R2                  ;REDUCE INNER COUNT
 9997 066726 001372                         BNE     14$                 ;LOOP TILL EXPIRES
 9998 066730 000766                         BR      12$                 ;LOOP UNTIL HALTED
 9999
10000 066732                        15$:    PRINTF  #T38MS2             ;TYPE CNTL C TO EXIT
       066732 012746 072641                                                     MOV     #T38MS2,-(SP)
       066736 012746 000001                                                     MOV     #1,-(SP)
       066742 010600                                                            MOV     SP,R0
       066744 104417                                                            TRAP    C$PNTF
       066746 062706 000004                                                     ADD     #4,SP
10001 066752 004737 016034                  JSR     PC,SOFINIT          ;DO SOFT INIT OF CONTROLLER
10002 066756 103405                         BCS     200$                ;BR IF SOFT INIT = OK
10006 066760 010001                         MOV     R0,R1               ;SAVE CONTENTS OF TSSR
10007 066762                                ERRDF   ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
       066762 104455                                                            TRAP    C$ERDF
       066764 001755                                                            .WORD   1005
       066766 003654                                                            .WORD   SFIERR
       066770 012074                                                            .WORD   SFIMSG
10008 066772                        200$:
10009 066772 013737 002202 071570           MOV     UNITN,T38DSW        ;SET UNIT NUMBER
10010 067000 012704 071550                  MOV     #T38PK2,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
10011 067004 004737 010662                  JSR     PC,WRTCHR           ;ISSUE WRITE CHARACTERISTICS
10012 067010 103405                         BCS     210$                ;BR, IF COMMAND ISSUED OK
10016 067012 010001                         MOV     R0,R1               ;SAVE CONTENTS OF TSSR
10017 067014                                ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED
       067014 104456                                                            TRAP    C$ERHRD
       067016 001756                                                            .WORD   1006
       067020 005060                                                            .WORD   WRTMSG
       067022 012074                                                            .WORD   SFIMSG
10018                               ;*****************************************************************
10019                               ;
10020                               ;       THIS WRITE SUB-SYSTEM MEMORY COMMAND JUST HOLDS THE LEDS OFF
10021                               ;
10022                               ;*****************************************************************
10023 067024                        210$:
10024 067024 112737 000000 071061           MOVB    #0,T38BS1           ;CLEAR BIT #4
10025 067032 112737 000025 071060           MOVB    #25,T38BSO          ;STOP DRIVE TEST 22
10026 067040 012704 071050                  MOV     #T38PACKET,R4       ;SET UP NEW WRT. SUBSYS MEM. COMMAND
10027 067044 010465 000000                  MOV     R4,TSDB(R5)         ;SET THE PACKET ADDRESS
10028 067050 004737 016376                  JSR     PC,CHKTSSR          ;WAIT FOR SSR TO SET
10029 067054 103405                         BCS     250$                ;BR IF CARRY SET (GOOD RETURN)
10030 067056 010001                         MOV     R0,R1               ;SAVE CONTENTS OF TSSR
10034 067060                                ERRDF   ERRNO,T38SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
       067060 104455                                                            TRAP    C$ERDF
       067062 001757                                                            .WORD   1007
       067064 072246                                                            .WORD   T38SSR
       067066 012106                                                            .WORD   PKTSSR
10035 067070                        250$:   CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
       067070 104406                                                            TRAP    C$CLP1
10036 067072                                SETPRI  #PRI07              ;RAISE THE PRIORITY
       067072 012700 000340                                                     MOV     #PRI07,R0
       067076 104441                                                            TRAP    C$SPRI
10037 067100 005037 071036                  CLR     TTION2              ;ASSUME INTERRUPTS ARE ENABLED
```

K2

SEQ 0230

TEST 10: MANUAL INTERVENTION

```
 10038 067104  032737  000100  177560          BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
 10039 067112  001005                           BNE     710$                    ;BRANCH IF YES
 10040 067114  005237  071036                   INC     TTION2                  ;FLAG SET IF INTERRUPTS OFF
 10041 067120  052737  000100  177560          BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
 10042 067126  012701  000060          710$:   MOV     #TTIVEC,R1              ;START OF TTI VECTORS
 10043 067132  011137  071040                   MOV     (R1),TVSAV2             ;SAVE THE CURRENT TTI VECTOR
 10044 067136  012721  070336                   MOV     #590$,(R1)+             ;SET NEW INTERRUPT ROUTINE
 10045 067142  011137  071042                   MOV     (R1),TPSAV2             ;SAVE THE VECTOR PRIORITY
 10046 067146  012711  000340                   MOV     #PRI07,(R1)             ;USE PRIORITY SEVEN
 10047 067152                                   SETPRI  #PRI00                  ;LOWER INTERRUPT BR LEVEL
       067152  012700  000000                                                           MOV     #PRI00,R0
       067156  104441                                                                   TRAP    C$SPRI
 10048 067160  000240          260$:   NOP                                     ;ALLOW CNTL C
 10049 067162  000776                   BR      260$                            ;LOOP UNTIL STOPPED
 10050
 10051
 10052 067164          20$:    PRINTF  #T38MS2                                 ;TELL'EM WHAT TO TYPE
       067164  012746  072641                                                           MOV     #T38MS2,-(SP)
       067170  012746  000001                                                           MOV     #1,-(SP)
       067174  010600                                                                   MOV     SP,R0
       067176  104417                                                                   TRAP    C$PNTF
       067200  062706  000004                                                           ADD     #4,SP
 10053 067204                                   SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
       067204  012700  000000                                                           MOV     #PRI00,R0
       067210  104441                                                                   TRAP    C$SPRI
 10054 067212  005037  002224                   CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 10055 067216  004737  016034                   JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 10056 067222  103405                           BCS     300$                    ;BR IF SOFT INIT = OK
 10060 067224  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 10061 067226                                   ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
       067226  104455                                                                   TRAP    C$ERDF
       067230  001760                                                                   .WORD   1008
      .067232  003654                                                                   .WORD   SFIERR
       067234  012074                                                                   .WORD   SFIMSG
 10062 067236          300$:
 10063 067236  013737  002202  071570                   MOV     UNITN,T38DSW            ;SET UNIT NUMBER IN PACKET
 10064 067244  012737  000040  071566                   MOV     #BIT5,T38EAI            ;ENABLE ATTENTION INTERRUPTS
 10065 067252  012704  071550                           MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
 10066 067256  004737  010662                           JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 10067 067262  103405                                   BCS     310$                    ;BR, IF COMMAND ISSUED OK
 10071 067264  010001                                   MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 10072 067266                                           ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
       067266  104456                                                                           TRAP    C$ERHRD
       067270  001761                                                                           .WORD   1009
       067272  005060                                                                           .WORD   WRTMSG
       067274  012074                                                                           .WORD   SFIMSG
 10073 067276          310$:
 10074 067276  012704  071600                           MOV     #T38PK3,R4              ;SET UP NEW PACKET FOR MESS BUF REL
 10075 067302  010465  000000                           MOV     R4,TSDB(R5)             ;MESSAGE BUFFER RELEASE,ACK,CVC=1 CMD
 10076 067306  004737  016310                           JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 10077 067312  005002                                   CLR     R2                      ;MAKE SURE ALL IS CLEAR
 10078 067314  016501  000002                           MOV     TSSR(R5),R1             ;GET TSSR STATUS
 10079 067320  032701  000100                           BIT     #OFL,R1                 ;IS OFL SET
 10080 067324  001402                                   BEQ     320$                    ;BR, IF OFL IS NOT SET
 10081 067326  052702  000100                           BIS     #OFL,R2                 ;SET OFL IN EXPECTED
 10082 067332  052702  000200          320$:   BIS     #SSR,R2                 ;SET UP EXPECTED
 10083 067336  020201                           CMP     R2,R1                   ;IS EVERYTHING OK
```

L2

TEST 10: MANUAL INTERVENTION

```
10084 067340  001404                        BEQ     350$                ;BR, IF ALL IS WELL
10088 067342                                ERRHRD  ERRNO,T38SST,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      067342  104456                                                        TRAP    C$ERHRD
      067344  001762                                                        .WORD   1010
      067346  072456                                                        .WORD   T38SST
      067350  012106                                                        .WORD   PKTSSR
10089 067352                        350$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      067352  104406                                                        TRAP    C$CLP1
10090 067354                                PRINTF  #T38MS1                 ;TELL OPERATOR TO TOGGLE SWITCH
      067354  012746  072546                                                MOV     #T38MS1,-(SP)
      067360  012746  000001                                                MOV     #1,-(SP)
      067364  010600                                                        MOV     SP,R0
      067366  104417                                                        TRAP    C$PNTF
      067370  062706  000004                                                ADD     #4,SP
10091 067374                                PRINTF  #T38MS2                 ;TELL OPERATOR TO DO ^C TO EXIT
      067374  012746  072641                                                MOV     #T38MS2,-(SP)
      067400  012746  000001                                                MOV     #1,-(SP)
      067404  010600                                                        MOV     SP,R0
      067406  104417                                                        TRAP    C$PNTF
      067410  062706  000004                                                ADD     #4,SP
10092 067414                                SETPRI  #PRI07                  ;RAISE THE PRIORITY
      067414  012700  000340                                                MOV     #PRI07,R0
      067420  104441                                                        TRAP    C$SPRI
10093 067422  005037  071036                CLR     TTION2                  ;ASSUME INTERRUPTS ARE ENABLED
10094 067426  032737  000100  177560        BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
10095 067434  001005                        BNE     720$                    ;BRANCH IF YES
10096 067436  005237  071036                INC     TTION2                  ;FLAG SET IF INTERRUPTS OFF
10097 067442  052737  000100  177560        BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
10098 067450  012701  000060        720$:   MOV     #TTIVEC,R1              ;START OF TTI VECTORS
10099 067454  011137  071040                MOV     (R1),TVSAV2             ;SAVE THE CURRENT TTI VECTOR
10100 067460  012721  070336                MOV     #590$,(R1)+             ;SET NEW INTERRUPT ROUTINE
10101 067464  011137  071042                MOV     (R1),TPSAV2             ;SAVE THE VECTOR PRIORITY
10102 067470  012711  000340                MOV     #PRI07,(R1)             ;USE PRIORITY SEVEN
10103 067474                                SETPRI  #PRI00                  ;LOWER INTERRUPT BR LEVEL
      067474  012700  000000                                                MOV     #PRI00,R0
      067500  104441                                                        TRAP    C$SPRI
10104 067502  000240        360$:   NOP                                     ;ALLOW CONTROL C
10105 067504  005737  002224                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
10106 067510  001001                        BNE     370$                    ;BRANCH IF YES
10107 067512  000773                        BR      360$                    ;WAIT SOME MORE FOR INTERRUPT
10108 067514                        370$:   PRINTF  #T38INT                 ;"INTERRUPT RECEIVED"
      067514  012746  072336                                                MOV     #T38INT,-(SP)
      067520  012746  000001                                                MOV     #1,-(SP)
      067524  010600                                                        MOV     SP,R0
      067526  104417                                                        TRAP    C$PNTF
      067530  062706  000004                                                ADD     #4,SP
10109 067534  016501  000002                MOV     TSSR(R5),R1             ;READ TSSR STATUS
10110 067540  032701  000100                BIT     #OFL,R1                 ;CHECK THE OFF-LINE BIT
10111 067544  001011                        BNE     380$                    ;BR, IF DRIVE IS OFF-LINE
10112 067546                                PRINTF  #T38ONL                 ;"DRIVE IS NOW ON-LINE"
      067546  012746  072366                                                MOV     #T38ONL,-(SP)
      067552  012746  000001                                                MOV     #1,-(SP)
      067556  010600                                                        MOV     SP,R0
      067560  104417                                                        TRAP    C$PNTF
      067562  062706  000004                                                ADD     #4,SP
10113 067566  000410                        BR      390$                    ;ALMOST DONE
10114 067570                        380$:   PRINTF  #T38OFL                 ;"DRIVE IS NOW OFF-LINE"
```

M2

TEST 10: MANUAL INTERVENTION

```
                067570  012746  072422                                                          MOV     #T38OFL,-(SP)
                067574  012746  000001                                                          MOV     #1, (SP)
                067600  010600                                                                  MOV     SP,R0
                067602  104417                                                                  TRAP    C$PNTF
                067604  062706  000004                                                          ADD     #4,SP
10115           067610  005037  002224          390$:   CLR     INTRECV                 ;CLEAR INTERRUPT FLAG
10116           067614  000137  067236                  JMP     300$                    ;TRY AGAIN
10117           067620                          25$:    GMANIL  T38MSG,T38DAT,-1,NO     ;WAIT FOR OPERATOR TO MOUNT TAPE
                067620  104443                                                                  TRAP    C$GMAN
                067622  000404                                                                  BR      10000$
                067624  073400                                                                  .WORD   T38DAT
                067626  000120                                                                  .WORD   T$CODE
                067630  072705                                                                  .WORD   T38MSG
                067632  177777                                                                  .WORD   -1
                067634                                                                  10000$:
10118           067634                          BNCOMPLETE 25$                  ;RETRY IF ERROR
                067634  103371                                                                  BCC     25$
10119           067636  005737  073400                  TST     T38DAT                  ;DID OPERATOR SAY 'YES' ?
10120           067642  001002                          BNE     27$                     ;BRANCH IF YES
10121           067644  000137  066312                  JMP     2$                      ;RETURN TO MAIN MENU
10122           067650                          27$:
10123           067650  004737  016034                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
10124           067654  103405                          BCS     400$                    ;BR IF SOFT INIT = OK
10128           067656  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10129           067660                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
                067660  104455                                                                  TRAP    C$ERDF
                067662  001763                                                                  .WORD   1011
                067664  003654                                                                  .WORD   SFIERR
                067666  012074                                                                  .WORD   SFIMSG
10130           067670                          400$:   CKLOOP                          ;LOOP IF SELECTED
                067670  104406                                                                  TRAP    C$CLP1
10131           067672  013737  002202  071570          MOV     UNITN,T38DSW            ;SET UNIT NUMBER
10132           067700  012704  071550                  MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
10133           067704  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
10134           067710  103405                          BCS     410$                    ;BR, IF COMMAND ISSUED OK
10138           067712  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10139           067714                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
                067714  104456                                                                  TRAP    C$ERHRD
                067716  001764                                                                  .WORD   1012
                067720  005060                                                                  .WORD   WRTMSG
                067722  012074                                                                  .WORD   SFIMSG
10140           067724                          410$:   CKLOOP                          ;LOOP IF SELECTED
                067724  104406                                                                  TRAP    C$CLP1
10141           067726  013701  071074                  MOV     T38BFR+6,R1             ;PICK UP XSTO CONTENTS
10142           067732  010102                          MOV     R1,R2                   ;SET UP EXPECTED
10143           067734  052702  000004                  BIS     #BIT2,R2                ;SET UP THE WRITE LOCKED BIT
10144           067740  020102                          CMP     R1,R2                   ;ARE THEY CORRECT
10145           067742  001406                          BEQ     430$                    ;BR, IF ALL IS WELL (OK)
10149           067744                          ERRHRD  ERRNO,T38WRL,EXPREC     ;"WRITE LOCKED BIT IS NOT SET ETC."
                067744  104456                                                                  TRAP    C$ERHRD
                067746  001765                                                                  .WORD   1013
                067750  072064                                                                  .WORD   T38WRL
                067752  015534                                                                  .WORD   EXPREC
10150           067754  005237  002222                  INC     FATFLG                  ;SET FATAL FLAG
10151           067760                          430$:   CKLOOP                          ;LOOP IF SELECTED
                067760  104406                                                                  TRAP    C$CLP1
10152           067762  005737  002222                  TST     FATFLG                  ;WAS THE DRIVE NOT WRITE LOCKED
```

N2

SEQ 0233

TEST 10: MANUAL INTERVENTION

```
10153 067766  001402                        BEQ     435$                    ;BR, IF FLAG NOT SET
10154 067770  000137  066312                JMP     2$                      ;RE-WRITE MENU
10155 067774  017737  113126  071622  435$: MOV     @FREE,T38WR             ;SET UP WRITE BUFFER ADDRESS
10156 070002  012704  071620                MOV     #T38PK4,R4              ;GET PACKET ADDRESS
10157 070006  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
10158 070012  004737  016310                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
10159 070016  016501  000002                MOV     TSSR(R5),R1             ;GET TSSR
10160 070022  012702  100206                MOV     #SC!SSR!BIT1!BIT2,R2    ;SET UP EXPECTED
10161 070026  020102                        CMP     R1,R2                   ;ARE THEY EQUAL (CORRECT)
10162 070030  001404                        BEQ     440$                    ;BR, IF CORRECT STATUS
10166 070032                                ERRHRD  ERRNO,T38WRT,PKTSSR     ;"TSSR INCORRECT AFTER WRITE COMMAND
      070032  104456                                                                TRAP    C$ERHRD
      070034  001766                                                                .WORD   1014
      070036  072000                                                                .WORD   T38WRT
      070040  012106                                                                .WORD   PKTSSR
10167 070042                          440$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      070042  104406                                                                TRAP    C$CLP1
10168 070044  013701  071074                MOV     T38BFR+6,R1             ;READ XSTO CONTENTS
10169 070050  010102                        MOV     R1,R2                   ;SET UPR EXPECTED
10170 070052  052702  004000                BIS     #BIT11,R2               ;SET THE WRITE LOCK ERROR BIT (XSTO)
10171 070056  020102                        CMP     R1,R2                   ;WAS THE BIT SET
10172 070060  001404                        BEQ     450$                    ;BR, IF IT WAS (GOOD)
10176 070062                                ERRHRD  ERRNO,T38WLE,EXPREC     ;"WRITE LOCK ERROR BIT NOT SET"
      070062  104456                                                                TRAP    C$ERHRD
      070064  001767                                                                .WORD   1015
      070066  072125                                                                .WORD   T38WLE
      070070  015534                                                                .WORD   EXPREC
10177 070072                          450$: CKLOOP                          ;LOOP IF SELECTED
      070072  104406                                                                TRAP    C$CLP1
10178 070074  000137  066312                JMP     2$                      ;GO BACK TO MENU
10179
10180                                 ;*************************************************************************
10181                                 ;       SERVO EXERCISER NO LONGER USED
10182                                 ;*************************************************************************
10183 070100                          500$:
10184 070100  012746  071705                PRINTB  #T38MS3                 ;"EXE ANY OTHER MENU SELECTION TO STOP
      070104  012746  000001                                                         MOV     #T38MS3,-(SP)
      070110  010600                                                                 MOV     #1,-(SP)
      070112  104414                                                                 MOV     SP,R0
      070114  062706  000004                                                         TRAP    C$PNTB
10185 070120  004737  073402                JSR     PC,T38REST              ;SET PACKET TO INITIAL VALUES
10186 070124  004737  016034                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
10187 070130  103405                        BCS     500$                    ;BR IF SOFT INIT = OK
10191 070132  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10192 070134                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      070134  104455                                                                ADD     #4,SP
      070136  001770                                                                TRAP    C$ERDF
      070140  003654                                                                .WORD   1016
      070142  012074                                                                .WORD   SFIERR
10193 070144  013737  002202  071570  500$: MOV     UNITN,T38DSW            ;SET UNIT NUMBER
10194 070152  012704  071550                MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
10195 070156  004737  010662                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
10196 070162  103405                        BCS     510$                    ;BR, IF COMMAND ISSUED OK
10200 070164  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10201 070166                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      070166  104456                                                                TRAP    C$ERHRD
```

TEST 10: MANUAL INTERVENTION

```
                070170  001771                                                          .WORD   1017
                070172  005060                                                          .WORD   WRTMSG
                070174  012074                                                          .WORD   SFIMSG
 10202          070176                          510$:
 10203          070176  112737  000000  071061          MOVB    #0,T38BS1               ;CLEAR BIT #4
 10204          070204  112737  000020  071060          MOVB    #20,T38BSO              ;EXECUTE DRIVE TEST 22
 10205          070212  012704  071050                  MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
 10206          070216  010465  000000                  MOV     R4,TSD8(R5)             ;SET THE PACKET ADDRESS
 10207          070222  004737  016376                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 10208          070226  103405                          BCS     550$                    ;BR IF CARRY SET (GOOD RETURN)
 10209          070230  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 10213          070232                                  ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                070232  104455                                                          TRAP    C$ERDF
                070234  001772                                                          .WORD   1018
                070236  072246                                                          .WORD   T38SSR
                070240  012106                                                          .WORD   PKTSSR
 10214          070242                          550$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                070242  104406                                                          TRAP    C$CLP1
 10215          070244                                  SETPRI  #PRI07                  ;RAISE THE PRIORITY
                070244  012700  000340                                                  MOV     #PRI07,R0
                070250  104441                                                          TRAP    C$SPRI
 10216          070252  005037  071036                  CLR     TTION2                  ;ASSUME INTERRUPTS ARE ENABLED
 10217          070256  032737  000100  177560          BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
 10218          070264  001005                          BNE     555$                    ;BRANCH IF YES
 10219          070266  005237  071036                  INC     TTION2                  ;FLAG SET IF INTERRUPTS OFF
 10220          070272  052737  000100  177560          BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
 10221          070300  012701  000060          555$:   MOV     #TTIVEC,R1              ;START OF TTI VECTORS
 10222          070304  011137  071040                  MOV     (R1),TVSAV2             ;SAVE THE CURRENT TTI VECTOR
 10223          070310  012721  070336                  MOV     #590$,(R1)+             ;SET NEW INTERRUPT ROUTINE
 10224          070314  011137  071042                  MOV     (R1),TPSAV2             ;SAVE THE VECTOR PRIORITY
 10225          070320  012711  000340                  MOV     #PRI07,(R1)             ;USE PRIORITY SEVEN
 10226          070324                                  SETPRI  #PRI00                  ;LOWER INTERRUPT BR LEVEL
                070324  012700  000000                                                  MOV     #PRI00,R0
                070330  104441                                                          TRAP    C$SPRI
 10227          070332  000240          560$:   NOP                                     ;LOOP AWHILE
 10228          070334  000776                  BR      560$                            ;STAY IN "TIGHT" LOOP
 10229                                          ;+
 10230                                          ;PROCESS CONSOLE INTERRUPTS
 10231                                          ;-
 10232
 10233          070336  010046          590$:   MOV     R0,-(SP)                        ;SAVE WORK REGISTER
 10234          070340  113700  177562          MOVB    @#TTIBFR,R0             ;GET THE OPERATOR INPUT
 10235          070344  042700  000200          BIC     #200,R0                 ;STRIP OFF PARITY BIT
 10236          070350  122700  000015          CMPB    #15,R0                  ;IS IT A CARRIAGE RETURN ?
 10237          070354  001075                  BNE     591$                    ;JUST EXIT IF NOT
 10238          070356  012766  066312  000002  MOV     #2$,2(SP)               ;RETURN TO MASTER MENU
 10239          070364  005066  000004          CLR     4(SP)                   ;FORCE PRIORTY 0
 10240          070370  013737  071040  000060  MOV     TVSAV2,@#TTIVEC ;RESTORE VECTOR
 10241          070376  013737  071042  000062  MOV     TPSAV2,@#TTIVEC+2       ;RESTORE SUPER PRIORTY
 10242          070404  112737  000025  071060  MOVB    #25,T38BSO              ;STOP DRIVE TEST 22
 10243          070412  112737  000000  071061  MOVB    #0,T38BS1               ;CLEAR BS1
 10244          070420  012704  071050          MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
 10245          070424  010465  000000          MOV     R4,TSD8(R5)             ;SET THE PACKET ADDRESS
 10246          070430  012737  176750  071044  MOV     #65000.,T38DLY          ;SET UP DELAY COUNTER
 10247          070436  004737  016310  592$:   JSR     PC,WAITF                ;DO A WAIT FOR SSR
 10248          070442  016501  000002          MOV     TSSR(R5),R1             ;CONTENTS OF TSSR REGISTER
 10249          070446  032701  000200          BIT     #SSR,R1                 ;CHECK FOR TSSR SET
```

TEST 10: MANUAL INTERVENTION

```
10250 070452  001017                        BNE     595$                    ;KEEP GOING IF NOT SET
10251 070454                                DELAY   250                     ;CALL DELAY ROUTINE
      070454  012727  000250                                                          MOV     #250,(PC)+
      070460  000000                                                                  .WORD   0
      070462  013727  002116                                                          MOV     L$DLY,(PC)+
      070466  000000                                                                  .WORD   0
      070470  005367  177772                                                          DEC     -6(PC)
      070474  001375                                                                  BNE     .-4
      070476  005367  177756                                                          DEC     -22(PC)
      070502  001367                                                                  BNE     .-20
10252 070504  005337  071044                        DEC     T38DLY                  ;BUMP COUNTER DOWN
10253 070510  001352                        BNE     592$                    ;BR, IF MORE TIME LEFT
10254 070512  004737  016376          595$: JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
10255 070516  103405                        BCS     580$                    ;BR IF CARRY SET (GOOD RETURN)
10256 070520  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10260 070522                                ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      070522  104455                                                                  TRAP    C$ERDF
      070524  001773                                                                  .WORD   1019
      070526  072246                                                                  .WORD   T38SSR
      070530  012106                                                                  .WORD   PKTSSR
10261 070532                          580$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      070532  104406                                                                  TRAP    C$CLP1
10262 070534  005737  071036                        TST     TTION2                  ;ARE SUPER INTERRUPTS ENABLED
10263 070540  001403                        BEQ     591$                    ;BR, IF YES
10264 070542  042737  000100  177560        BIC     #100,@#TTICSR           ;RESTORE REGISTER
10265 070550  012600                  591$: MOV     (SP)+,R0                ;RESTORE REGISTER
10266 070552  000002                        RTI                             ;RETURN
10267 070554                          35$:
10268 070554  004737  073402                JSR     PC,T38REST              ;SET PACKET TO INITIAL VALUES
10269 070560  004737  016034                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
10270 070564  103405                        BCS     600$                    ;BR IF SOFT INIT = OK
10274 070566  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10275 070570                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      070570  104455                                                                  TRAP    C$ERDF
      070572  001774                                                                  .WORD   1020
      070574  003654                                                                  .WORD   SFIERR
      070576  012074                                                                  .WORD   SFIMSG
10276 070600                          600$: CKLOOP                          ;LOOP IF SELECTED
      070600  104406                                                                  TRAP    C$CLP1
10277 070602  012701  071066                        MOV     #T38BFR,R1              ;ADDRESS OF MESSAGE BUFFER
10278 070606  012702  125252                        MOV     #125252,R2              ;ALTERNATING 1'S AND 0'S
10279
10280 070612  010221                  601$: MOV     R2,(R1)+                ;CLEAR OUT THE MESSAGE BUFFER
10281 070614  022701  071542                        CMP     #T38EB,R1               ;END OF BUFFER YET
10282 070620  001401                        BEQ     605$                    ;BR, IF AT END OF BUFFER
10283 070622  000773                        BR      601$                    ;NOT AT END KEEP GOING
10284 070624  013737  002202  071570  605$: MOV     UNITN,T38DSW            ;SET UNIT NUMBER
10285 070632  012704  071550                        MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
10286 070636  004737  010662                        JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
10287 070642  103405                        BCS     610$                    ;BR, IF COMMAND ISSUED OK
10291 070644  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10292 070646                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      070646  104456                                                                  TRAP    C$ERHRD
      070650  001775                                                                  .WORD   1021
      070652  005060                                                                  .WORD   WRTMSG
      070654  012074                                                                  .WORD   SFIMSG
10293 070656                          610$: CKLOOP                          ;LOOP IF SELECTED
```

```
TEST 10: MANUAL INTERVENTION

        070656  104406                                                          TRAP    C$CLP1
10294   070660  112737  000000  071061          MOVB    #0,T38S1                ;CLEAR BIT #4
10295   070666  112737  000024  071060          MOVB    #24,T38S0               ;READ EXTENDED DRIVE STATUS
10296   070674  012704  071050                  MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
10297   070700  010465  000000                  MOV     R4,TSD8(R5)             ;SET THE PACKET ADDRESS
10298   070704  012737  000144  071044          MOV     #100.,T38DLY            ;SET UP DELAY ROUTINE
10299   070712  004737  016310          620$:   JSR     PC,WAITF                ;WAIT AWHILE FOR SSR TO SET
10300   070716  016501  000002                  MOV     TSSR(R5),R1             ;SEE IF IT REALLY DID
10301   070722  032701  000200                  BIT     #SSR,R1                 ;JUST CHECK THAT BIT
10302   070726  001017                          BNE     630$                    ;BR. IF SSR IS SET
10303   070730                                  DELAY   250                     ;DELAY ABOUT .25 SEC
        070730  012727  000250                                                  MOV     #250.,(PC)+
        070734  000000                                                          .WORD   0
        070736  013727  002116                                                  MOV     @#DLY,(PC)+
        070742  000000                                                          .WORD   0
        070744  005367  177772                                                  DEC     -6(PC)
        070750  001375                                                          BNE     .-4
        070752  005367  177756                                                  DEC     -22(PC)
        070756  001367                                                          BNE     .-20
10304   070760  005337  071044                  DEC     T38DLY                  ;START DELAY COUNT DOWN
10305   070764  001352                          BNE     620$                    ;BR. IF COUNTER IS NOT AT DONE
10306   070766  004737  016376          630$:   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
10307   070772  103405                          BCS     650$                    ;BR IF CARRY SET (GOOD RETURN)
10308   070774  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10312   070776                                  ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        070776  104455                                                          TRAP    C$ERDF
        071000  001776                                                          .WORD   1022
        071002  072246                                                          .WORD   T38SSR
        071004  012106                                                          .WORD   PKTSSR
10313   071006                  650$:   CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
        071006  104406                                                          TRAP    C$CLP1
10314   071010  012700  071106                  MOV     #T38BFR+20,R0           ;MESSAGE BUFFER ADDRESS
10315   071014  005001                          CLR     R1                      ;NO HIGH ORDER ADDRESS BITS
10316   071016  005037  003136                  CLR     KTENABLE                ;NO KT11 STUFF EITHER
10317   071022  004737  073440                  JSR     PC,T38MBP               ;GO PRINT MESSAGE BUFFER CONTENTS
10318   071026  000137  066312                  JMP     2$                      ;GO BACK TO MENU
10319
10320
10321                           ;63$:   JMP     200                     ;TYPE ^C TO RETURN TO THE SUPERVISOR
10322   071032                  64$:    EXIT    TST                     ;LEAVE TEST
        071032  104432                                                          TRAP    C$EXIT
        071034  003056                                                          .WORD   L10075-.
10323
10324                           ;+
10325                           ;LOCAL TEXT MESSAGES FOR TEST
10326                           ;-
10327
10328                           ;LOCAL STORAGE FOR THIS TEST
10329                           ;-
10330                           ;+
10331                           ;LOCAL STORAGE FOR THIS TEST
10332                           ;-
10333
10334   071036  000000          TTION2:         .WORD   0               ;WORD SET IF SUPERVISOR TTI INTER OFF
10335   071040  000000          TVSAV2: .WORD   0               ;SAVE TTI VECTOR
10336   071042  000000          TPSAV2: .WORD   0               ;SAVE TTI PRIORITY
10337
```

E3

TEST 10: MANUAL INTERVENTION

```
10338 071044  000000                T38DLY: .WORD   0                       ;DELAY COUNTER FOR TEST
10340         071050                        .=<.+10>&177770
10342 071050                        T38PACKET:                              ;COMMAND PACKET FOR TEST
10343 071050  140006                        .WORD   140006                  ;WRITE SUBSYSTEM MEM. CMD.  ACK,CVC=1
10344 071052  071060                        .WORD   T38TAD                  ;ADDRESS OF CHARACTERISTICS BLOCK
10345 071054  000000                        .WORD   0
10346 071056  000012                        .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE
10347 071060                        T38TAD:                         ;CHARACTERISTICS DATA BLOCK
10348 071060     000                T38BS0: .BYTE   0                       ;BSEL0 BYTE
10349 071061     000                T38BS1: .BYTE   0                       ;BSEL1 BYTE
10350 071062  000000                T38BS2: .WORD   0                       ;BSEL1 WORD
10351 071064  000000                        .WORD   0                       ;DATA
10352 071066                        T38BFR: .BLKW   150.                    ;MESSAGE BUFFER
10353 071542  000000                T38EB:  .WORD                           ;END OF BUFFER ADDRESS
10354
10355
10357         071550                        .=<.+10>&177770
10359 071550                        T38PK2:                                 ;COMMAND PACKET FOR TEST
10360 071550  140004                        .WORD   140004                  ;WRITE CHARA. MEM. CMD.,  ACK,CVC=1
10361 071552  071560                        .WORD   T38DTA                  ;ADDRESS OF SELECT DATA BLOCK
10362 071554  000000                        .WORD   0
10363 071556  000012                        .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE
10364
10365
10366 071560                        T38DTA:                                 ;SELECT DATA BLOCK
10367 071560  071066                        .WORD   T38BFR                  ;ADDRESS OF MESSAGE BUFFER
10368 071562  000000                        .WORD   0
10369 071564  000400                        .WORD   256.                    ;LENGTH OF MESSAGE BUFFER
10370 071566  000000                T38EAI: .WORD   0                       ;EAI BIT WORD
10371 071570  000000                T38DSW: .WORD   0                       ;DRIVE SELECT WORD ETC
10373         071600                        .=<.+10>&177770
10375 071600  140212                T38PK3: .WORD   140212                  ;MESSAGE BUFFER RELEASE COMMAND
10376 071602  000000                        .WORD   0                       ;NOT USED
10377 071604  000000                        .WORD   0                       ;NOT USED
10378 071606  000000                        .WORD   0                       ;NOT USED
10379 071610  000000                        .WORD   0                       ;NOT USED
10380                               ;
10381                               ;WRITE TAPE PACKET
10382                               ;
10384         071620                        .=<.+10>&177770
10386 071620  140005                T38PK4: .WORD   140005                  ;WRITE, ACK, CVC=1 COMMAND
10387 071622  000000                T38WR:  .WORD   0                       ;ADDRESS OF WRITE BUFFER
10388 071624  000000                        .WORD   0                       ;MORE ADDRESS OF WRITE BUFFER
10389 071626  000400                T38SIZ: .WORD   256.                    ;SIZE OF RECORD
10390
10391
10392
10393
10394                               ;.
10395                               ;LOCAL TEXT MESSAGES FOR TEST
10396                               ;-
10397
10398
10399
10400
10401
10402 071630     123     164     141  T38NE:  .ASCIZ  'Stand-alone Manual Intervention Not Executed'
```

F3


```
TEST 10: MANUAL INTERVENTION

10403 071705      045      116      045  T38MS3: .ASCIZ  '<NS>A Type <RETURN> To Stop Servo Exerciser, Return To Menu'
10404 072000      124      123      123  T38WRT: .ASCIZ  'TSSR Not Correct After WRITE, With WRITE PROTECT On'
10405 072064      127      122      111  T38WRL: .ASCIZ  'WRITE LOCKED Bit Not Set In XSTO'
10406 072125      127      122      111  T38WLE: .ASCIZ  'WRITE LOCK ERROR B t Not Set In XSTO'
10407 072172      127      122      111  T38NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
10408 072246      103      157      156  T38SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
10409 072336      045      116      045  T38INT: .ASCIZ  '<NS>A Interrupt Received'
10410 072366      045      116      045  T38ONL: .ASCIZ  '<NS>A Drive Is Now   ON LINE'
10411 072422      045      116      045  T38OFL: .ASCIZ  '<NS>A Drive Is Now  OFF LINE'
10412 072456      103      157      156  T38SST: .ASCIZ  'Contents Of TSSR Incorrect After MESSAGE BUFFER RELEASE'
10413 072546      045      116      045  T38MS1: .ASCIZ  '<NS>AToggle ON-LINE Switch to Generate ATTENTION Interrupts'
10414 072641      045      116      045  T38MS2: .ASCIZ  '<NS> 'ype RETURN To Return To Menu<NS>N'
10415 072705      111      163      040  T38MSG: .ASCIZ  'Is Write-Protected Tape Mounted'
10416 072745      115      141      156  T38ID:  .ASCIZ  'Manual Intervention'
10417                                                    .EVEN
10418 072772  073016   073070   073116  MIMENU: .WORD   1$,2$,3$,4$,5$,6$
10419 073006  073265   073330   073377          .WORD   8$,9$,10$,0
10420
10421 073016      012      123      105  1$:     .ASCIZ  <12>'SELECT OPERATION FROM FOLLOWING OPTIONS:'
10422 073070      012      011      060  2$:     .ASCIZ  <12>'    0       Display This Menu'
10423 073116      011      061      011  3$:     .ASCIZ  '        1       Turn On All M7196 LED''s'
10424 073150      011      062      011  4$:     .ASCIZ  '        2       Turn Off All M7196 LED''s'
10425 073203      011      063      011  5$:     .ASCIZ  '        3       Offline/Online Attention'
10426 073237      011      064      011  6$:     .ASCIZ  '        4       Write Protect Test'
10427 073265      011      065      011  8$:     .ASCIZ  '        5       Print Extended Transport Status'
10428 073330      011      136      103  9$:     .ASCIZ  '        ^C      To Return to Diagnostic Supervisor'
10429 073377      000                   10$:    .ASCIZ  ''
10430                                                    .EVEN
10431
10432                                    ;*
10433                                    ;LOCAL STORAGE FOR THIS TEST
10434                                    ;-
10435
10436 073400  000000                    T38DAT: .WORD   0               ;LOGICAL RESPONSE TO QUESTION
10437 073402                            T38REST:
10438 073402                                    SAVREG                  ;SAVE THE REGISTERS
10439 073406  012701   071050                   MOV     #T38PACKET,R1   ;START OF THE PACKET
10440 073412  012721   140206                   MOV     #140206,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
10441 073416  012721   071060                   MOV     #T38TAD,(R1)+   ;ADDRESS OF DATA BLOCK
10442 073422  005021                            CLR     (R1)+           ;EXTENDED ADDRESS
10443 073424  012721   000006                   MOV     #6.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
10444 073430  005021                            CLR     (R1)+           ;CLEAR BSEL0 AND BSEL1
10445 073432  005021                            CLR     (R1)+           ;CLEAR SEL2
10446 073434  005011                            CLR     (R1)            ;CLEAR DATA AREA
10447 073436  000207                            RTS     PC              ;RETURN
10448
10449
10450                                    ;*
10451                                    ;
10452                                    ;THIS ROUTINE PRINTS THE CONTENTS OF
10453                                    ;THE 256 BYTE MESSAGE BUFFER RETURNED BY THE
10454                                    ;TSV-05.
10455                                    ;
10456                                    ;INPUT:
10457                                    ;
10458                                    ;       R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
10459                                    ;       R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
```

TEST 10: MANUAL INTERVENTION

```
10460                                   ;       NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
10461                                   ;
10462                                   ;
10463                                   ;-
10464
10465 073440                            T38MBP:
10466 073440                                    SAVREG                          ;SAVE THE REGISTERS
10467 073444  010005                              MOV     R0,R5                 ;SAVE LOW ORDER ADDRESS
10468 073446  005737  003136                      TST     KTENABLE              ;ADDRESS ABOVE 28K?
10469 073452  001001                              BNE     910$                  ;BR IF YES
10470 073454  005001                              CLR     R1                    ;SET HIGH ORDER ADDRESS TO 0
10471 073456  010103                    910$:     MOV     R1,R3                 ;SAVE HIGH ORDER ADDRESS
10472 073460  006100                              ROL     R0                    ;SHIFT BIT15 TO C BIT
10473 073462  006101                              ROL     R1                    ;SHIFT TO HIGH ORDER FOR PRINTOUT
10474 073464                                      PRINTX  #T38AS0,R1,R5         ;PRINT MESSAGE BUFFER ADDRESS
      073464  010546                                                                  MOV     R5,-(SP)
      073466  010146                                                                  MOV     R1,-(SP)
      073470  012746  073742                                                          MOV     #T38AS0,-(SP)
      073474  012746  000003                                                          MOV     #3,-(SP)
      073500  010600                                                                  MOV     SP,R0
      073502  104415                                                                  TRAP    C$PNTX
      073504  062706  000010                                                          ADD     #10,SP
10475 073510                                      PRINTX  #T38AS1               ;PRINT HEADER FOR CONTENTS
      073510  012746  074007                                                          MOV     #T38AS1,-(SP)
      073514  012746  000001                                                          MOV     #1,-(SP)
      073520  010600                                                                  MOV     SP,R0
      073522  104415                                                                  TRAP    C$PNTX
      073524  062706  000004                                                          ADD     #4,SP
10476 073530  010501                              MOV     R5,R1                 ;COPY LOW ORDER ADDRESS
10477 073532  010300                              MOV     R3,R0                 ;COPY HIGH ORDER ADDRESS
10478 073534  001403                              BEQ     913$                  ;BR IF NOT ABOVE 28K
10479 073536  004737  017356                      JSR     PC,SETMAP             ;SETUP PAR ADDRESS IN R0
10480 073542  010005                              MOV     R0,R5                 ;GET PAR FORMAT ADDRESS ABOVE 28K
10481 073544  010537  074110              913$:   MOV     R5,T38CNT             ;HOLD ADDRESS
10482 073550  011504                      911$:   MOV     (R5),R4               ;GET BUFFER ENTRY
10483 073552  022704  125252                      CMP     #125252,R4            ;CHECK FOR NO LOAD CONDITION
10484 073556  001417                              BEQ     912$                  ;BR, IF BUFFER WASN'T LOADED
10485 073560  010403                              MOV     R4,R3                 ;MAKE COPY
10486 073562  042704  170377                      BIC     #170377,R4            ;ONLY BITS 11,10,9 AND 8 ARE SAVED
10487 073566  000241                              CLC                           ;CLEAR CARRY
10488 073570  006004                              ROR     R4                    ;11 TO 10 BIT POSITION
10489 073572  006004                              ROR     R4                    ;10 TO 9 BIT POSITION
10490 073574  006004                              ROR     R4                    ;9 TO 8 BIT POSITON
10491 073576  006004                              ROR     R4                    ;8 TO 7 BIT POSITION
10492 073600  042703  177760                      BIC     #177760,R3            ;ONLY BITS 3,2,1 AND 0 ARE SAVED
10493 073604  060403                              ADD     R4,R3                 ;"OR'EM TOGETHER
10494 073606  010325                              MOV     R3,(R5)+              ;PUT BACK IN BUFFER
10495 073610  020527  071542                      CMP     R5,#T38EB             ;END OF BUFFER YET
10496 073614  001355                              BNE     911$                  ;BR, IF NOT AT END YET
10497 073616  013705  074110              912$:   MOV     T38CNT,R5             ;PUT ADDRESS BACK
10498 073622  012704  000001                      MOV     #1,R4                 ;START BYTE NUMBER AT ONE
10499 073626                              915$:   PRINTX  #T38ASN,R4,(R5)+              ;PRT MEM BUFFER W/NEWLINE
      073626  012546                                                                  MOV     (R5)+,-(SP)
      073630  010446                                                                  MOV     R4,-(SP)
      073632  012746  074064                                                          MOV     #T38ASN,-(SP)
      073636  012746  000003                                                          MOV     #3,-(SP)
      073642  010600                                                                  MOV     SP,R0
```

H3

TEST 10: MANUAL INTERVENTION

```
        073644  104415                                                          TRAP    C$PNTX
        073646  062706  000010                                                  ADD     #10,SP
 10500  073652  005037  074110                  CLR     T38CNT                  ;CLEAR COUNTER
 10501  073656  000412                          BR      921$                    ;SKIP OTHER PRINT
 10502  073660                          920$:   PRINTX  #T38ASC,R4,(R5)+        ;PRINT THE CONTENTS OF MEMORY BUFFER
        073660  012546                                                          MOV     (R5)+,-(SP)
        073662  010446                                                          MOV     R4,-(SP)
        073664  012746  074045                                                  MOV     #T38ASC,-(SP)
        073670  012746  000003                                                  MOV     #3,-(SP)
        073674  010600                                                          MOV     SP,R0
        073676  104415                                                          TRAP    C$PNTX
        073700  062706  000010                                                  ADD     #10,SP
 10503  073704  005237  074110          921$:   INC     T38CNT                  ;BUMP COUNTER
 10504  073710  005204                          INC     R4              ;NUMBER OF THE NEXT
 10505  073712  020427  000200                  CMP     R4,#128.                ;DONE ALL YET ?
 10506  073716  003010                          BGT     50$             ;BRANCH IF ALL DONE
 10507  073720  023727  074110  000004        CMP     T38CNT,#4               ;DONE FOUR YET
 10508  073726  001401                          BEQ     925$                    ;BR. IF THREE DONE
 10509  073730  000753                          BR      920$            ;KEEP GOING
 10510  073732  005037  074110          925$:   CLR     T38CNT                  ;CLEAR COUNTER
 10511  073736  000733                          BR      915$                    ;PRINT WITH NEW LINE
 10512  073740  000207                  50$:    RTS     PC              ;RETURN
 10513
 10514  073742     045     116     045  T38AS0: .ASCIZ  '%N%A Message Buffer Address = %01%05'
 10515  074007     045     116     045  T38AS1: .ASCIZ  '%N%A Message Buffer Contents:'
 10516  074045     045     101     040  T38ASC: .ASCIZ  '%A  %04%A: %03'
 10517  074064     045     116     045  T38ASN: .ASCIZ  '%N%A Byte%04%A: %03'
 10518                                   .EVEN
 10519  074110  000000                  T38CNT: .WORD                   ;COUNTER FOR PRINT
 10520  074112                          ENDTST
        074112                                                                  L10075:
        074112  104401                                                                  TRAP    C$ETST
 10521                          .SBTTL  TEST 11: CONFIGURATION TYPEOUT
 10522
 10523                  ;THIS IS A STANDALONE ROUTINE THAT PRINTS OUT ON THE CONSOLE TERMINAL
 10524                  ;THE CONFIGURATION OF THE M7196 MODULE AND TSV05 SUBSYSTEM. SPECIFICALLY,
 10525                  ;THE FOLLOWING INFORMATION IS PRESENTED:
 10526                  ;
 10527                  ;
 10528                  ; 1.0 STATE OF THE EXTENDED FEATURES SWITCH ON THE M7196: ON (EXTENDED
 10529                  ;     FEATURES ENABLED) OR OFF (EXTENDED FEATURES DISABLED).
 10530                  ;
 10531                  ; 2.0 STATE OF THE BUFFERING ENABLE SWITCH ON THE M7196: ON
 10532                  ;     (BUFFERING ENABLED) OR OFF (BUFFERING DISABLED).
 10533                  ;
 10534                  ; 3.0 MICROCODE REVSION LEVEL OF THE M7196.
 10535                  ;
 10536                  ; 4.0 NUMBER OF TAPE TRANSPORTS CONNECTED TO THE CONTROLLER.
 10537                  ;
 10538                  ; 5.0 UNIT SELECT CODE AND STATE (ONLINE/OFFLINE, WRITE ENABLED/PROTECTED)
 10539                  ;     OF EACH CONNECTED TRANSPORT. IN ADDITION, THE PROGRAM WILL INDICATE,
 10540                  ;     FOR EACH ON-LINE TRANSPORT, WHETHER OR NOT IT IS EQUIPPED WITH THE
 10541                  ;     EXTENDED TAPE STATUS READOUT FEATURE.
 10542                  ;
 10543                  ;
 10544                  ;THE OPERATOR IS EXPECTED TO READ THE PRINTOUT AND VERIFY THAT IT MATCHES
 10545                  ;THE ACTUAL CONFIGURATION AT HAND.  IF, FOR EXAMPLE, THE PROGRAM INDICATES
```

TEST 11: CONFIGURATION TYPEOUT

```
10546                                   ;THAT IT "SEES" TWO TRANSPORTS CONNECTED WHEN IN FACT ONLY ONE IS PRESENT.
10547                                   ;THE OPERATOR MUST INTERPRET THIS AS AN ERROR AND ATTEMPT TO FIND THE
10548                                   ;CAUSE (BAD CABLE, FAULTY UNIT SELECT DECODING IN THE TRANSPORT, ETC.).
10549                                   ;[SINCE THE CONTROLLER CAN ONLY ACCESS UNIT 0 IF IT IS IN "STANDARD"
10550                                   ;MODE, THE PROGRAM WILL FORCE THE MODULE INTO EXTENDED MODE VIA THE
10551                                   ;WRITE SUBSYSTEM MEMORY COMMAND IN ORDER TO SCAN FOR CONNECTED TRANSPORTS.]
10552                                   ;
10553                                   ;
10554                                   ;THIS ROUTINE, WHEN ITS ACTIONS ARE COMPLETED, WILL EXIT BACK TO THE
10555                                   ;DIAGNOSTIC SUPERVISOR SO THAT IF ADDITIONAL UNITS (CONTROLLERS) ARE
10556                                   ;SELECTED (E.G., FROM THE INITIAL STARTUP DIALOG), THE ROUTINE WILL BE
10557                                   ;REENTERED SO THAT THEIR CONFIGURATIONS CAN BE PRINTED.
10558                                   ;
10559 074114                                   BGNTST
       074114                                                                   T11::
10564 074114                           632$:    RFLAGS  R0                      ;GET OPERATOR FLAGS
       074114  104421                                                                  TRAP    C$RFLA
10565 074116  001403                            BEQ     10$                     ;BR, IF OK TO RUN
10566 074120  012700  076373                     MOV     #T39NE,R0               ;"TEST NOT EXECUTED"
10567 074124  000402                            BR      11$                     ;JUMP OUT OF TEST IF NOT
10568 074126  012700  077574             10$:    MOV     #TST39ID,R0             ;TEST ID MESSAGE
10569 074132  004737  016550             11$:    JSR     PC,TSTSETUP             ;DO THE COMMON SETUP
10570 074136  004737  020022                     JSR     PC,CHKMAN               ;IS MANUAL INTERVENTION ALLOWED?
10571 074142  103402                            BCS     20$                     ;BR, IF MANUAL INTERVENTION ALLOWED
10572 074144  000137  075376                     JMP     64$                     ;JUMP TO OUT IF NOT
10573 074150                           20$:
10574 074150  004737  016034                     JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
10575 074154  103405                            BCS     25$                     ;BR IF SOFT INIT = OK
10579 074156  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10580 074160                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
       074160  104455                                                                  TRAP    C$ERDF
       074162  002115                                                                  .WORD   1101
       074164  003654                                                                  .WORD   SFIERR
       074166  012074                                                                  .WORD   SFIMSG
10581 074170                           25$:     CKLOOP                                  ;LOOP IF SELECTED
       074170  104406                                                                  TRAP    C$CLP1
10582 074172  013737  002202  076340             MOV     UNITN,T39DSW            ;SET UNIT NUMBER
10583 074200  012704  076320                     MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
10584 074204  004737  010662                     JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
10585 074210  103405                            BCS     50$                     ;BR, IF COMMAND ISSUED OK
10589 074212  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10590 074214                                    ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
       074214  104456                                                                  TRAP    C$ERHRD
       074216  002116                                                                  .WORD   1102
       074220  005060                                                                  .WORD   WRTMSG
       074222  012074                                                                  .WORD   SFIMSG
10591 074224                           50$:     CKLOOP                          ;LOOP IF SELECTED
       074224  104406                                                                  TRAP    C$CLP1
10592 074226  013701  075650                     MOV     T39BFR+12,R1            ;GET XST2 STATUS FROM MESSAGE BUFFER
10593 074232                                    PRINTX  #T39SFS                 ;"STATE OF EXTENDED FEATURES SW ="
       074232  012746  077312                                                          MOV     #T39SFS,-(SP)
       074236  012746  000001                                                          MOV     #1,-(SP)
       074242  010600                                                                  MOV     SP,R0
       074244  104415                                                                  TRAP    C$PNTX
       074246  062706  000004                                                          ADD     #4,SP
10594 074252  032701  000200                     BIT     #BIT7,R1                ;CHECK STATE OF E.F.S.
10595 074256  001011                            BNE     100$                    ;BR, IF EXT. FEA. SW. IS ON
```

J3

TEST 11: CONFIGURATION TYPEOUT

```
10596 074260                          PRINTX  #T390FF                 ;" OFF"
      074260  012746  077436                                                  MOV     #T390FF, (SP)
      074264  012746  000001                                                  MOV     #1, (SP)
      074270  010600                                                          MOV     SP,R0
      074272  104415                                                          TRAP    C$PNTX
      074274  062706  000004                                                  ADD     #4,SP
10597 074300  000410                  BR      110$                    ;SKIP OTHER PRINT STATEMENT
10598 074302                  100$:   PRINTX  #T390N                  ;" ON "
      074302  012746  077445                                                  MOV     #T390N,-(SP)
      074306  012746  000001                                                  MOV     #1, (SP)
      074312  010600                                                          MOV     SP,R0
      074314  104415                                                          TRAP    C$PNTX
      074316  062706  000004                                                  ADD     #4,SP
10599 074322                  110$:   PRINTX  #T39SBS                 ;"STATE OF BUFFERING SWITCH ="
      074322  012746  077364                                                  MOV     #T39SBS,-(SP)
      074326  012746  000001                                                  MOV     #1,-(SP)
      074332  010600                                                          MOV     SP,R0
      074334  104415                                                          TRAP    C$PNTX
      074336  062706  000004                                                  ADD     #4,SP
10600 074342  032701  000100          BIT     #BIT6,R1                ;CHECK STATE OF BUFFERING SW
10601 074346  001011                  BNE     120$                    ;BR, IF BUFFERING IS ON
10602 074350                          PRINTX  #T390FF                 ;" OFF"
      074350  012746  077436                                                  MOV     #T390FF,-(SP)
      074354  012746  000001                                                  MOV     #1,-(SP)
      074360  010600                                                          MOV     SP,R0
      074362  104415                                                          TRAP    C$PNTX
      074364  062706  000004                                                  ADD     #4,SP
10603 074370  000410                  BR      130$                    ;SKIP OTHER PRINT STATEMENT
10604 074372                  120$:   PRINTX  #T390N                  ;" ON "
      074372  012746  077445                                                  MOV     #T390N,-(SP)
      074376  012746  000001                                                  MOV     #1, (SP)
      074402  010600                                                          MOV     SP,R0
      074404  104415                                                          TRAP    C$PNTX
      074406  062706  000004                                                  ADD     #4,SP
10605 074 12  042701  177700          130$:   BIC     #177700,R1              ;ONLY LEAVE MICROCODE REV LEVEL
10606 074416  010137  077532                  MOV     R1,T39RL                ;LOAD UP REV LEVEL
10607 074422                          PRINTX  #T39MCL,T39RL           ;"MICROCODE REVISION LEVEL =000XXX"
      074422  013746  077532                                                  MOV     T39RL,-(SP)
      074426  012746  077454                                                  MOV     #T39MCL,-(SP)
      074432  012746  000002                                                  MOV     #2,-(SP)
      074436  010600                                                          MOV     SP,R0
      074440  104415                                                          TRAP    C$PNTX
      074442  062706  000006                                                  ADD     #6,SP
10608 074446  004737  016034                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
10609 074452  103405                  BCS     140$                    ;BR IF SOFT INIT = OK
10613 074454  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10614 074456                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      074456  104455                                                          TRAP    C$ERDF
      074460  002117                                                          .WORD   1103
      074462  003654                                                          .WORD   SFIERR
      074464  012074                                                          .WORD   SFIMSG
10615 074466                  140$:   CKLOOP                                  ;LOOP IF SELECTED
      074466  104406                                                          TRAP    C$CLP1
10616 074470  013737  002202  076340          MOV     UNITN,T39DSW            ;SET UNIT NUMBER
10617 074476  012704  076320                  MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
10618 074502  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
10619 074506  103405                  BCS     150$                    ;BR, IF COMMAND ISSUED OK
```

TEST 11: CONFIGURATION TYPEOUT

```
10623 074510  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10624 074512                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      074512  104456                                                          TRAP    C$ERHRD
      074514  002120                                                          .WORD   1104
      074516  005060                                                          .WORD   WRTMSG
      074520  012074                                                          .WORD   SFIMSG
10625 074522                          150$:   CKLOOP                          ;LOOP IF SELECTED
      074522  104406                                                          TRAP    C$CLP1
10626 074524  005737  002226                    TST     EXTFEA                  ;CHECK FOR EXTENDED FEATURES SW SWITCH
10627 074530  001036                            BNE     174$                    ;BR IF SWITCH IS ON
10628 074532  112737  000200  075631            MOVB    #200,T39BS1             ;WRITE MISCELLANEOUS CONT/READ STATUS
10629 074540  112737  000010  075630            MOVB    #10,T39BSO              ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
10630 074546  012704  075620                    MOV     #T39PACKET,R4           ;WRITE SUBSYS MEM PACKET
10631 074552  010465  000000                    MOV     R4,TSDB(R5)             ;ISSUE COMMAND
10632 074556  004737  016376                    JSR     PC,CHKTSSR              ;WAIT FOR SSR
10633 074562  103405                            BCS     160$                    ;BR, IF NO ERROR
10634 074564  010001                            MOV     R0,R1                   ;ERROR, SAVE TSSR
10638 074566                                  ERRHRD  ERRNO,T39NBA,PKTSSR     ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      074566  104456                                                          TRAP    C$ERHRD
      074570  002121                                                          .WORD   1105
      074572  077075                                                          .WORD   T39NBA
      074574  012106                                                          .WORD   PKTSSR
10639 074576                          160$:   CKLOOP                          ;LOOP IF SELECTED
      074576  104406                                                          TRAP    C$CLP1
10640 074600  012704  076320                    MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
10641                                 ;*****************************************************************
10642                                 ;
10643                                 ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
10644                                 ;
10645                                 ;*****************************************************************
10646
10647 074604  004737  010662                    JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
10648 074610  103405                            BCS     170$                    ;BR, IF COMMAND ISSUED OK
10652 074612  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10653 074614                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      074614  104456                                                          TRAP    C$ERHRD
      074616  002122                                                          .WORD   1106
      074620  005060                                                          .WORD   WRTMSG
      074622  012074                                                          .WORD   SFIMSG
10654 074624                          170$:   CKLOOP                          ;SCOPE LOOP
      074624  104406                                                          TRAP    C$CLP1
10655 074626  005037  002202            174$:   CLR     UNITN                   ;SET TO DRIVE 0
10656 074632  013737  002202  076340    175$:   MOV     UNITN,T39DSW            ;SET UNIT NUMBER
10657 074640  012704  076320                    MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
10658 074644  004737  010662                    JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
10659 074650  103405                            BCS     180$                    ;BR, IF COMMAND ISSUED OK
10663 074652  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
10664 074654                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      074654  104456                                                          TRAP    C$ERHRD
      074656  002123                                                          .WORD   1107
      074660  005060                                                          .WORD   WRTMSG
      074662  012074                                                          .WORD   SFIMSG
10665 074664                          180$:   CKLOOP                          ;LOOP IF SELECTED
      074664  104406                                                          TRAP    C$CLP1
10666
10667 074666  016501  000002            190$:   MOV     TSSR(R5),R1             ;GET TSSR STATUS
10668 074672  032701  000100                    BIT     #OFL,R1                 ;CHECK FOR OFF-LINE
```

TEST 11: CONFIGURATION TYPEOUT

```
10669 074676  001414                          BEQ     200$            ;BR, IF DRIVE IS ON-LINE
10670 074700                                  PRINTX  #T390F2,UNITN   ;"DRIVE NUMBER XX IS OFF-LINE"
      074700  013746  002202                                                  MOV     UNITN,-(SP)
      074704  012746  076634                                                  MOV     #T390F2,-(SP)
      074710  012746  000002                                                  MOV     #2,-(SP)
      074714  010600                                                         MOV     SP,R0
      074716  104415                                                         TRAP    C$PNTX
      074720  062706  000006                                                  ADD     #6,SP
10671 074724  000137  075260                  JMP     250$            ;DO NOT TRY TO GET ANYMORE INFO.
10672 074730                          200$:   PRINTX  #T390N2,UNITN   ;"DRIVE NUMBER XX IS ON-LINE"
      074730  013746  002202                                                  MOV     UNITN,-(SP)
      074734  012746  076700                                                  MOV     #T390N2,-(SP)
      074740  012746  000002                                                  MOV     #2,-(SP)
      074744  010600                                                         MOV     SP,R0
      074746  104415                                                         TRAP    C$PNTX
      074750  062706  000006                                                  ADD     #6,SP
10673 074754  013701  075644                  MOV     T39BFR+6,R1     ;READ EXTENDED STATUS (XSTO)
10674 074760  032701  000004                  BIT     #BIT2,R1        ;IS DRIVE WTITE PROTECTED
10675 074764  001013                          BNE     210$            ;BR, IF WRITE PROTECTED
10676 074766                                  PRINTX  #T39WPN,UNITN   ;"DRIVE NUMBER IS NOT WRT PRO"
      074766  013746  002202                                                  MOV     UNITN,-(SP)
      074772  012746  077016                                                  MOV     #T39WPN,-(SP)
      074776  012746  000002                                                  MOV     #2,-(SP)
      075002  010600                                                         MOV     SP,R0
      075004  104415                                                         TRAP    C$PNTX
      075006  062706  000006                                                  ADD     #6,SP
10677 075012  000412                          BR      220$            ;SKIP OVER
10678 075014                          210$:   PRINTX  #T39WRT,UNITN   ;"DRIVE NUMBER XX IS WRT PRO"
      075014  013746  002202                                                  MOV     UNITN,-(SP)
      075020  012746  076743                                                  MOV     #T39WRT,-(SP)
      075024  012746  000002                                                  MOV     #2,-(SP)
      075030  010600                                                         MOV     SP,R0
      075032  104415                                                         TRAP    C$PNTX
      075034  062706  000006                                                  ADD     #6,SP
10679 075040  012737  125252  075736  220$:   MOV     #125252,T39BFR+100      ;SET 1 LOC TO KNOWN VALUE
10680 075046  112737  000000  075631          MOVB    #0,T39BS1       ;EXTENDED TAPE STATUS
10681 075054  112737  000024  075630          MOVB    #24,T39BS0      ;EXTENDED TAPE STATUS
10682 075062  012704  075620                  MOV     #T39PACKET,R4   ;WRITE SUBSYS MEM PACKET
10683 075066  010465  000000                  MOV     R4,TSDB(R5)     ;ISSUE COMMAND
10684 075072  012737  000144  075612          MOV     #100.,T39DLY    ;SET UP DELAY ROUTINE
10685 075100  004737  016310          222$:   JSR     PC,WAITF        ;WAIT AWHILE FOR SSR TO SET
10686 075104  016501  000002                  MOV     TSSR(R5),R1     ;SEE IF IT REALLY DID
10687 075110  032701  000200                  BIT     #SSR,R1         ;JUST CHECK THAT BIT
10688 075114  001017                          BNE     225$            ;BR, IF SSR IS SET
10689 075116                                  DELAY   250             ;DELAY ABOUT .25 SEC
      075116  012727  000250                                                  MOV     #250.,(PC)+
      075122  000000                                                          .WORD   0
      075124  013727  002116                                                  MOV     L$DLY,(PC)+
      075130  000000                                                          .WORD   0
      075132  005367  177772                                                  DEC     -6(PC)
      075136  001375                                                          BNE     .-4
      075140  005367  177756                                                  DEC     -22(PC)
      075144  001367                                                          BNE     .-20
10690 075146  005337  075612                  DEC     T39DLY          ;START DELAY COUNT DOWN
10691 075152  001352                          BNE     222$            ;BR, IF COUNTER IS NOT AT DONE
10692 075154  004737  016376          225$:   JSR     PC,CHKTSSR      ;WAIT FOR SSR
10693 075160  103405                          BCS     230$            ;BR, IF NO ERROR
```

M3

TEST 11: CONFIGURATION TYPEOUT

```
10694 075162  010001                          MOV     R0,R1                   ;ERROR, SAVE TSSR
10698 075164                                  ERRHRD  ERRNO,T39NBA,PKTSSR     ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      075164  104456                                                                  TRAP    C$ERHRD
      075166  002124                                                                  .WORD   1108
      075170  077075                                                                  .WORD   T39NBA
      075172  012106                                                                  .WORD   PKTSSR
10699 075174                          230$:   CKLOOP                          ;LOOP IF SELECTED
      075174  104406                                                                  TRAP    C$CLP1
10700 075176  023727  075736  125252          CMP     T39BFR+100,#125252      ;DID LOC GET OVER WRITTEN
10701 075204  001013                          BNE     240$                    ;BR, IF IT DIDN'T GET ETC.
10702 075206                                  PRINTX  #T39ETN,UNITN           ;"DRIVE DOESN'T HAVE EXT TAPE STATUS
      075206  013746  002202                                                          MOV     UNITN,-(SP)
      075212  012746  076541                                                          MOV     #T39ETN,-(SP)
      075216  012746  000002                                                          MOV     #2,-(SP)
      075222  010600                                                                  MOV     SP,R0
      075224  104415                                                                  TRAP    C$PNTX
      075226  062706  000006                                                          ADD     #6,SP
10703 075232  000412                          BR      250$                    ;SKIP OVER
10704 075234                          240$:   PRINTX  #T39ETS,UNITN           ;"DRIVE HAS EXT TAPE STATUS"
      075234  013746  002202                                                          MOV     UNITN,-(SP)
      075240  012746  076452                                                          MOV     #T39ETS,-(SP)
      075244  012746  000002                                                          MOV     #2,-(SP)
      075250  010600                                                                  MOV     SP,R0
      075252  104415                                                                  TRAP    C$PNTX
      075254  062706  000006                                                          ADD     #6,SP
10705 075260  005237  002202          250$:   INC     UNITN                   ;BUMP DRIVE NUMBER
10706 075264  023727  002202  000003          CMP     UNITN,#3                ;AT END OF DRIVES YET
10707 075272  001402                          BEQ     631$                    ;BR, IF NO MORE DRIVES
10708 075274  000137  074632                  JMP     175$                    ;DO NEXT DRIVE
10709 075300  012700  075402          631$:   MOV     #NUMENU,R0              ;MENU OF MANUAL INTERVENTIONS
10710 075304  012701  000001                  MOV     #1,R1                   ;MAXIMUM ALLOWED SELECTION
10711 075310  004737  017600                  JSR     PC,GETSEL               ;GO GET THE OPERATORS SELECTION
10712 075314  010004                          MOV     R0,R4                   ;GET NUMBER FROM ROUTINE
10713 075316  006304                          ASL     R4                      ;CONVERT TO WORD OFFSET
10714 075320  000174  075324                  JMP     @6$(R4)                 ;JUMP TO PROPER LOOP
10715 075324  075300          6$:     .WORD   631$                    ;RETYPE THE MENU
10716 075326  074114                  .WORD   632$                    ;RERUN THE TEST
10717 075330                          63$:    SETPRI  #PRI00
      075330  012700  000000                                                          MOV     #PRI00,R0
      075334  104441                                                                  TRAP    C$SPRI
10718 075336                                  printf  #T38MS2
      075336  012746  072641                                                          MOV     #T38MS2,-(SP)
      075342  012746  000001                                                          MOV     #1,-(SP)
      075346  010600                                                                  MOV     SP,R0
      075350  104417                                                                  TRAP    C$PNTF
      075352  062706  000004                                                          ADD     #4,SP
10719 075356                                  PRINTX  #T39NFL                 ;NEW LINE
      075356  012746  076370                                                          MOV     #T39NFL,-(SP)
      075362  012746  000001                                                          MOV     #1,-(SP)
      075366  010600                                                                  MOV     SP,R0
      075370  104415                                                                  TRAP    C$PNTX
      075372  062706  000004                                                          ADD     #4,SP
10720 075376                          64$:    EXIT    TST                     ;EXIT THIS SECTION
      075376  104432                                                                  TRAP    C$EXIT
      075400  002222                                                                  .WORD   L10076-.
10721
10722                                 ;+
                                      ;LOCAL TEXT MESSAGES FOR TEST
```

TEST 11: CONFIGURATION TYPEOUT

```
10723                                   ;-
10724
10725                                   ;LOCAL STORAGE FOR THIS TEST
10726                                   ;-
10727 075402  075416  075470  075516    NUMENU: .WORD   1$,2$,3$,4$,5$,0
10728 075416     012     123     105    1$:     .ASCIZ  <12>'SELECT OPERATION FROM FOLLOWING OPTIONS:'
10729 075470     012     011     060    2$:     .ASCIZ  <12>'   0       Display This Menu'
10730 075516     011     061     011    3$:     .ASCIZ  '       1       Rerun this test'
10731 075541     011     136     103    4$:     .ASCIZ  '       ^C      To Return to Diagnostic Supervisor'
10732 075610     000                    5$:     .ASCIZ  ''
10733                                           .EVEN
10734
10735 075612  000000                    T39DLY: .WORD   0                       ;DELAY COUNTER FOR TEST
10737         075620                            .=<.+10>&177770
10739 075620                            T39PACKET:                      ;COMMAND PACKET FOR TEST
10740 075620  140006                            .WORD   140006                  ;WRITE SUBSYSTEM MEM. CMD.  ACK,CVC=1
10741 075622  075630                            .WORD   T39TAD                  ;ADDRESS OF CHARACTERISTICS BLOCK
10742 075624  000000                            .WORD   0
10743 075626  000012                            .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE
10744 075630                            T39TAD:                         ;CHARACTERISTICS DATA BLOCK
10745 075630     000                    T39BS0: .BYTE   0                       ;BSEL0 BYTE
10746 075631     000                    T39BS1: .BYTE   0                       ;BSEL1 BYTE
10747 075632  000000                    T39BS2: .WORD   0                       ;BSEL1 WORD
10748 075634  000000                            .WORD   0                       ;DATA
10749 075636                            T39BFR: .BLKW   150.                    ;MESSAGE BUFFER
10750
10751
10753         076320                            .=<.+10>&177770
10755 076320                            T39PK2:                         ;COMMAND PACKET FOR TEST
10756 076320  140004                            .WORD   140004                  ;WRITE CHARA. MEM. CMND.,  ACK,CVC=1
10757 076322  076330                            .WORD   T39DTA                  ;ADDRESS OF SELECT DATA BLOCK
10758 076324  000000                            .WORD   0
10759 076326  000012                            .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE
10760
10761
10762 076330                            T39DTA:                         ;SELECT DATA BLOCK
10763 076330  075636                            .WORD   T39BFR                  ;ADDRESS OF MESSAGE BUFFER
10764 076332  000000                            .WORD   0
10765 076334  000400                            .WORD   256.                    ;LENGTH OF MESSAGE BUFFER
10766 076336  000000                    T39EAI: .WORD   0                       ;EAI BIT WORD
10767 076340  000000                    T39DSW: .WORD   0                       ;DRIVE SELECT WORD ETC
10769         076350                            .=<.+10>&177770
10771 076350  140012                    T39PK3: .WORD   140012                  ;MESSAGE BUFFER RELEASE COMMAND
10772 076352  000000                            .WORD   0                       ;NOT USED
10773
10774                                   ;WRITE TAPE PACKET
10775                                   ;
10777         076360                            .=<.+10>&177770
10779 076360  140005                    T39PK4: .WORD   140005                  ;WRITE, ACK, CVC=1 COMMAND
10780 076362  000000                    T39WR:  .WORD   0                       ;ADDRESS OF WRITE BUFFER
10781 076364  000000                            .WORD   0                       ;MORE ADDRESS OF WRITE BUFFER
10782 076366  000400                    T39SIZ: .WORD   256.                    ;SIZE OF RECORD
10783
10784
10785
10786
10787                                   ;+
```

TEST 11: CONFIGURATION TYPEOUT

```
10788                                   ;LOCAL TEXT MESSAGES FOR TEST
10789                                   ;-
10790
10791
10792
10793
10794 076370     045     116     000    T39NFL: .ASCIZ  '%N'
10795 076373     123     164     141    T39NE:  .ASCIZ  'Stand-alone Configuration Typeout Not Executed'
10796 076452     045     116     045    T39ETS: .ASCIZ  '%N%A Extended Tape Status Availiable, Drive Number %D2'
10797 076541     045     116     045    T39ETN: .ASCIZ  '%N%A Extended Tape Status NOT Availiable, Drive Number %D2'
10798 076634     045     116     045    T39OF2: .ASCIZ  '%N%A Drive Number %D2%A Is Off-Line'
10799 076700     045     116     045    T39ON2: .ASCIZ  '%N%A Drive Number %D2%A Is On-Line'
10800 076743     045     116     045    T39WRT: .ASCIZ  '%N%A Drive Number %D2%A Is Write Protected'
10801 077016     045     116     045    T39WPN: .ASCIZ  '%N%A Drive Number %D2%A Is NOT Write Protected'
10802 077075     127     122     111    T39NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
10803 077151     103     157     156    T39SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
10804 077241     045     116     045    T39QUIT: .ASCIZ  '%N%A Type ^C to Return to the Supervisor'
10805 077312     045     116     045    T39SFS: .ASCIZ  '%N%A State Of Extended Features Switch  ='
10806 077364     045     116     045    T39SBS: .ASCIZ  '%N%A State Of Buffering Switch          ='
10807 077436     045     101     040    T39OFF: .ASCIZ  '%A OFF'
10808 077445     045     101     040    T39ON:  .ASCIZ  '%A ON '
10809 077454     045     116     045    T39MCL: .ASCIZ  '%N%A M7196 Microcode Revision Level     =%02'
10810                                           .EVEN
10811 077532  000000                    T39RL:  .WORD   0
10812                                           .EVEN
10813                                           .EVEN
10814
10815                                   ;*
10816                                   ;LOCAL STORAGE FOR THIS TEST
10817                                   ;-
10818
10819 077534  000000                    T39DAT: .WORD   0               ;LOGICAL RESPONSE TO QUESTION
10820 077536                            T39REST:
10821 077536                                    SAVREG                  ;SAVE THE REGISTERS
10822 077542  012701  075620                    MOV     #T39PACKET,R1   ;START OF THE PACKET
10823 077546  012721  140006                    MOV     #140006,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
10824 077552  012721  075630                    MOV     #T39TAD,(R1)+   ;ADDRESS OF DATA BLOCK
10825 077556  005021                            CLR     (R1)+           ;EXTENDED ADDRESS
10826 077560  012721  000006                    MOV     #6.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
10827 077564  005021                            CLR     (R1)+           ;CLEAR BSEL0 AND BSEL1
10828 077566  005021                            CLR     (R1)+           ;CLEAR SEL2
10829 077570  005011                            CLR     (R1)            ;CLEAR DATA AREA
10830 077572  000207                            RTS     PC              ;RETURN
10831
10832                                   ;*
10833                                   ;LOCAL TEXT MESSAGES FOR TEST
10834                                   ;-
10835
10836 077574     103     157     156    TST39ID:        .ASCIZ  'Configuration Typeout'
10837                                           .EVEN
10838 077622                                    ENDTST
      077622                                                            L10076:
      077622  104401                                                            TRAP    C$ETST
10839
10840                                           .SBTTL  TEST 12: SCOPE LOOPS
10841
10842
```

TEST 12: SCOPE LOOPS

```
10843                                   ;*
10844                                   ;
10845                                   ;
10846                                   ;
10847                                   ;       THIS IS A STANDALONE ROUTINE PROVIDING A NUMBER OF TIGHT "SCOPE
10848                                   ;       LOOPS" USEFUL FOR DEBUGGING BASIC REGISTER ACCESS PROBLEMS WITH
10849                                   ;       THE M7196 MODULE.  THESE SCOPE LOOPS CAN BE USED WHEN THE NORMAL
10850                                   ;       "LOOP ON ERROR" OR "LOOP ON TEST (SUBTEST)" FACILITIES DON'T
10851                                   ;       SEEM TO ALLOW THE OPERATOR TO ZERO IN A PROBLEM IN THE EARLY
10852                                   ;       TESTS (I.E. THE HARDWARE MAY NOT BE RESPONDING TO A REGISTER
10853                                   ;       ACCESS, CAUSING A BUS ERROR TRAP, EVEN THOUGH THE DEVICE ADDRESS
10854                                   ;       SELECTED BY THE PROGRAM MATCHES THE CONFIGURATION SET UP IN THE
10855                                   ;       HARDWARE DIP SWITCHES).  THE FOLLOWING MENU OF SCOPE LOOPS ARE
10856                                   ;       AVAILABLE:
10857                                   ;
10858                                   ;               CODE    SCOPE LOOP
10859                                   ;
10860                                   ;                 0     HELP. PRINT THIS MENU.
10861                                   ;                 1     TSBA READ ACCESS
10862                                   ;                 2     TSSR READ ACCESS
10863                                   ;                 3     INITIALIZE (TSSR WRITE ACCESS)
10864                                   ;                 4     TSDB HIGH BYTE WRITE ACCESS
10865                                   ;                 5     TSDB LOW BYTE WRITE ACCESS
10866                                   ;                 6     TSDB MAINTENANCE-MODE WORD WRITE ACCESS
10867                                   ;                 7     TSDBX (TSSR HIGH BYTE) WRITE ACCESS
10868                                   ;                         (EXTENDED FEATURES SWITCH MUST BE ON
10869                                   ;                          TO USE SELECTION CODE 7)
10870                                   ;                 8     EXIT (RETURN TO SUPERVISOR)
10871                                   ;
10872                                   ;       FOR SCOPE LOOPS THAT WRITE INTO REGISTERS, THE PROGRAM PROMPTS
10873                                   ;       THE OPERATOR FOR THE DATA TO BE WRITTEN.  TYPING <RETURN> CAUSES
10874                                   ;       AN EXIT FROM THE SCOPE LOOP BACK TO MENU LEVEL.
10875                                   ;
10876                                   ;-
10877
10878
10879 077624                                    BGNTST
       077624                                                                  T12::
10884 077624                                    RFLAGS  R0                      ;GET OPERATOR FLAGS
       077624  104421                                                                  TRAP    C$RFLA
10885 077626  001403                            BEQ     1$                      ;BR, IF OK TO RUN
10886 077630  012700  101215                    MOV     #T40NE,R0               ;"TEST NOT EXECUTED"
10887 077634  000402                            BR      100$                    ;JUST EXIT IF NOT
10888 077636  012700  101262              1$:   MOV     #TST40ID,R0             ;TEST ID MESSAGE
10889 077642  004737  016550              100$: JSR     PC,TSTSETUP             ;DO THE COMMON SETUP
10890 077646  004737  020022                    JSR     PC,CHKMAN               ;SEE IF MANUAL INTERVENTION ALLOWED
10891 077652  103402                            BCS     2$                      ;CARRY SET IF INTERVENTION ALLOWED
10892 077654  000137  100336                    JMP     64$                     ;EXIT IF NO MANUAL INTERVENTION
10893 077660  004737  016034              2$:   JSR     PC,SOFINIT              ;DO A SOFT INIT
10894 077664  103405                            BCS     5$                      ;BRANCH IF OK
10895 077666  010001                            MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
10899 077670                                    ERROF   ERRNO,SFIERR,SFIMSG     ;REPORT FATAL ERROR
       077670  104455                                                                  TRAP    C$EROF
       077672  002261                                                                  .WORD   1201
       077674  003654                                                                  .WORD   SFIERR
       077676  012074                                                                  .WORD   SFIMSG
10900 077700  012700  100350              5$:   MOV     #SCMENU,R0              ;MENU OF SCOPE LOOP SELECTIONS
```

TEST 12: SCOPE LOOPS

```
10901 077704  012701  000007                MOV     #7.,R1          ;MAXIMUM ALLOWED SELECTION
10902 077710  004737  017600                JSR     PC,GETSEL       ;GO GET THE OPERATORS SELECTION
10903 077714  005700                       TST     R0              ;WAS ZERO SPECIFIED ?
10904 077716  001760                       BEQ     2$              ;REPEA  MENU IF YES.
10905 077720  020027  000007                CMP     R0,#7           ;EXTENDED TSSR ?
10906 077724  001015                       BNE     3$              ;BRANCH IF NOT
10907 077726  005737  002226                TST     EXTFEA                  ;CHECK FOR EXTENDED FEATURES SET
10908 077732  001012                       BNE     3$              ;BR, IF IT IS ON
10909 077734                               PRINTF  #EXFMSG         ;WARN OPERATOR EXTENDED FEATURES CLEAR
      077734  012746  101137                                                  MOV     #EXFMSG,-(SP)
      077740  012746  000001                                                  MOV     #1,-(SP)
      077744  010600                                                          MOV     SP,R0
      077746  104417                                                          TRAP    C$PNTF
      077750  062706  000004                                                  ADD     #4,SP
10910 077754  000137  077660                JMP     2$              ;GO BACK TO BASIC MENU
10911 077760  010004                3$:     MOV     R0,R4           ;SAVE THE MENU SELECTION
10912 077762                               SETPRI  #PRI07          ;RAISE THE PRIORITY
      077762  012700  000340                                                  MOV     #PRI07,R0
      077766  104441                                                          TRAP    C$SPRI
10913 077770  005037  100342                CLR     TTION           ;ASSUME INTERRUPTS ARE ENABLED
10914 077774  032737  000100  177560        BIT     #100,@#TTICSR   ;ARE TTI INTERRUPTS ON ?
10915 100002  001005                       BNE     4$              ;BRANCH IF YES
10916 100004  005237  100342                INC     TTION           ;FLAG SET IF INTERRUPTS OFF
10917 100010  052737  000100  177560        BIS     #100,@#TTICSR   ;ENABLE INTERRUPTS
10918 100016  012701  000060          4$:   MOV     #TTIVEC,R1      ;START OF TTI VECTORS
10919 100022  011137  100344                MOV     (R1),TVECSAV    ;SAVE THE CURRENT TTI VECTOR
10920 100026  012721  100250                MOV     #60$,(R1)+      ;SET NEW INTERRUPT ROUTINE
10921 100032  011137  100346                MOV     (R1),TPRISAV    ;SAVE THE VECTOR PRIORITY
10922 100036  012711  000340                MOV     #PRI07,(R1)     ;USE PRIORITY SEVEN
10923 100042                               SETPRI  #PRI00          ;LOWER INTERRUPT BR LEVEL
      100042  012700  000000                                                  MOV     #PRI00,R0
      100046  104441                                                          TRAP    C$SPRI
10924 100050  006304                       ASL     R4              ;CONVERT TO WORD OFFSET
10925 100052  000174  100056                JMP     @6$(R4)         ;JUMP TO PROPER LOOP
10926 100056  077660                6$:     .WORD   2$              ;RETYPE THE MENU
10927 100060  100076                        .WORD   10$             ;TSBA READ ACCESS
10928 100062  100106                        .WORD   15$             ;TSSR READ ACCESS
10929 100064  100120                        .WORD   20$             ;TSSR WRITE ACCESS
10930 100066  100140                        .WORD   25$             ;TSDB HIGH BYTE WRITE ACCESS
10931 100070  100164                        .WORD   30$             ;TSDB LOW BYTE WRITE ACCESS
10932 100072  100210                        .WORD   35$             ;TSDB MAINTENANCE MODE
10933 100074  100230                        .WORD   40$             ;TSDBX WRITE ACCESS
10934                               ;                               ;LEAVE THE TEST
10935
10936
10937 100076  105065  000000        10$:    CLRB    TSDB(R5)        ;ENTER MAINTENANCE MODE
10938 100102  011500                12$:    MOV     (R5),R0         ;READ TSBA REGISTER
10939 100104  000776                        BR      12$             ;LOOP UNTIL HALTED
10940
10941
10942 100106  012703  000002        15$:    MOV     #TSSR,R3        ;ADDRESS OF TSSR REGISTER
109 3 100112  060503                        ADD     R5,R3           ;POINT TO TSV05'S REGISTERS
10944 100114  011300                18$:    MOV     (R3),R0         ;READ TSSR REGISTER
10945 100116  000776                        BR      18$             ;LOOP UNTIL STOPPED
10946
10947 100120  004737  017516        20$:    JSR     PC,GETPAT       ;READ THE DATA PATTERN
10948 100124  010001                        MOV     R0,R1           ;DATA PATTERN FOR LOOP
```

TEST 12: SCOPE LOOPS

```
10949 100126  012703  000002            MOV     #TSSR,R3        ;ADDRESS OF TSSR
10950 100132  060503                    ADD     R5,R3           ;POINT TO TSV05'S REGISTERS
10951 100134  010113            22$:    MOV     R1,(R3)         ;WRITE DATA TO TSSR
10952 100136  000776                    BR      22$             ;LOOP
10953
10954
10955 100140  105065  000000      25$:  CLRB    TSDB(R5)        ;ENTER MAINTENANCE MODE
10956 100144  004737  017516            JSR     PC,GETPAT       ;READ THE DATA PATTERN
10957 100150  010001                    MOV     R0,R1           ;DATA PATTERN FOR LOOP
10958 100152  012703  000001            MOV     #TSDBH,R3       ;ADDRESS OF HIGH BYTE OF TSDB
10959 100156  060503                    ADD     R5,R3           ;POINT TO TSV05'S REGISTERS
10960 100160  110113            27$:    MOVB    R1,(R3)         ;WRITE THE DATA TO TSDB, HIGH BYTE
10961 100162  000776                    BR      27$             ;LOOP UNTIL STOPPED
10962
10963
10964 100164  105065  000000      30$:  CLRB    TSDB(R5)        ;ENTER MAINTENANCE MODE
10965 100170  004737  017516            JSR     PC,GETPAT       ;READ THE DATA PATTERN
10966 100174  010001                    MOV     R0,R1           ;DATA PATTERN FOR LOOP
10967 100176  012703  000000            MOV     #TSDB,R3        ;ADDRESS OF TSSR
10968 100202  060503                    ADD     R5,R3           ;POINT TO TSV05'S REGISTERS
10969 100204  110113            32$:    MOVB    R1,(R3)         ;WRITE DATA TO TSSR, LOW BYTE
10970 100206  000776                    BR      32$             ;LOOP UNTIL HALTED BY OPERATOR
10971
10972 100210  004737  017516      35$:  JSR     PC,GETPAT       ;READ THE DATA PATTERN
10973 100214  010001                    MOV     R0,R1           ;DATA PATTERN FOR LOOP
10974 100216  012703  000000            MOV     #TSDB,R3        ;SELECT TSDB
10975 100222  060503                    ADD     R5,R3           ;POINT TO TSV05'S REGISTERS
10976 100224  010113            37$:    MOV     R1,(R3)         ;WRITE THE DATA PATTERN
10977
10978 100226  000776                    BR      37$             ;LOOP UNTIL HALTED
10979
10980 100230  004737  017516      40$:  JSR     PC,GETPAT       ;READ THE DATA PATTERN
10981 100234  010001                    MOV     R0,R1           ;SAVE THE DATA PATTERN
10982 100236  012703  000003            MOV     #TSSRH,R3       ;BYTE ADDRESS OF TSSR, HIGH BYTE
10983 100242  060503                    ADD     R5,R3           ;POINT TO TSV05'S REGISTERS
10984 100244  110113            42$:    MOVB    R1,(R3)         ;WRITE THE DATA TO REGISTER
10985 100246  000776                    BR      42$             ;LOOP UNTIL HALTED
10986
10987
10988                           ;+
10989                           ;PROCESS CONSOLE INTERRUPTS
10990                           ;-
10991
10992 100250  010046      60$:  MOV     R0,-(SP)        ;SAVE WORK REGISTER
10993 100252  113700  177562            MOVB    @#TTIBFR,R0     ;GET THE OPERATOR INPUT
10994 100256  042700  000200            BIC     #200,R0         ;STRIP OFF PARITY BIT
10995 100262  122700  000015            CMPB    #15,R0          ;IS IT A CARRIAGE RETURN ?
10996 100266  001021                    BNE     61$             ;JUST EXIT IF NOT
10997 100270  012766  077660  000002    MOV     #2$,2(SP)       ;RETURN TO MASTER MENU
10998 100276  005066  000004            CLR     4(SP)           ;FORCE PRIORITY ZERO
10999 100302  013737  100344  000060    MOV     TVECSAV,@#TTIVEC ;RESTORE SUPERVISOR VECTOR
11000 100310  013737  100346  000062    MOV     TPRISAV,@#TTIVEC+2 ;RESTORE SUPERVISOR PRIORITY
11001 100316  005737  100342            TST     TTION           ;ARE SUPERVISOR INTERRUPTS ENABLED ?
11002 100322  001403                    BEQ     61$             ;BRANCH IF YES
11003 100324  042737  000100  177560    BIC     #100,@#TTICSR   ;TURN OFF TTI INTERRUPTS
11004 100332  012600            61$:    MOV     (SP)+,R0        ;RESTORE REGISTER
11005 100334  000002                    RTI                     ;RETURN FROM INTERUPT
```

F4

TEST 12: SCOPE LOOPS

```
 11006
 11007 100336                           64$:
 11008 100336                           63$:    EXIT    TST                     ;EXIT THE TEST
       100336  104432                                                           TRAP    C$EXIT
       100340  000736                                                           .WORD   L10077-.
 11009                                  ;65$:   JMP     200                     ;RETURN TO SUPERVISOR
 11010                                  ;**************JUMP TO 200 IS NOT SUPPORTED ON XXDP+ VERSION 2.X.
 11011                                  ;+
 11012                                  ;LOCAL STORAGE FOR THIS TEST
 11013                                  ;
 11014
 11015 100342  000000                   TTION:          .WORD   0               ;WORD SET IF SUPERVISOR TTI INTER OFF
 11016 100344  000000                   TVECSAV:        .WORD   0               ;SAVE TTI VECTOR
 11017 100346  000000                   TPRISAV:        .WORD   0               ;SAVE TTI PRIORITY
 11018
 11019
 11020                                  ;+
 11021                                  ;MENU FOR OPERATOR INPUT FOR SCOPE LOOPS
 11022                                  ;-
 11023
 11024                                          .EVEN
 11025 100350  100402  100455  100503   SCMENU: .WORD   1$,2$,3$,4$,5$,6$
 11026 100364  100654  100712  100760           .WORD   7$,8$,9$,10$,11$,12$,0
 11027
 11028
 11029 100402     012     123     105   1$:     .ASCIZ  <12>'SELECT SCOPE LOOP FROM FOLLOWING OPTIONS:'
 11030 100455     012     011     060   2$:     .ASCIZ  <12>'    0      Display This Menu'
 11031 100503     011     061     011   3$:     .ASCIZ  '        1      TSDA Read Access'
 11032 100527     011     062     011   4$:     .ASCIZ  '        2      TSSR Read Access'
 11033 100553     011     063     011   5$:     .ASCIZ  '        3      Initialize (TSSR Write Access)'
 11034 100615     011     064     011   6$:     .ASCIZ  '        4      TSDB High Byte Write Access'
 11035 100654     011     065     011   7$:     .ASCIZ  '        5      TSDB Low Byte Write Access'
 11036 100712     011     066     011   8$:     .ASCIZ  '        6      TSDB Maintenance Mode Write Access'
 11037 100760     011     067     011   9$:     .ASCIZ  '        7      TSDBX (TSSR High Byte) Write Access'
 11038 101027     011     136     103   10$:    .ASCIZ  '        ^C     To Return to Diagnostic Supervisor'
 11039 101076     000                   11$:    .ASCIZ  ''
 11040 101077     124     171     160   12$:    .ASCIZ  'Type RETURN To Stop Scope Loops'
 11041 101137     045     116     045   EXFMSG: .ASCIZ  '%%%A *** Extended Features Switch Not On *** '
 11042 101215     123     164     141   T4ONE:  .ASCIZ  'Stand-alone Scope Loops Not Executed'
 11043 101262     123     143     157   TST40ID:        .ASCIZ  'Scope Loops'
 11044
 11045 101276                                   .EVEN
                                                ENDTST
       101276                                                                   L10077:
       101276  104401                                                                   TRAP    C$ETST
 11046 101300                                   ENDMOD
 11048                                          .TITLE  TSV6 - PARAMETER CODING
 11054
 11059
 11065
 11066 101300                                   BGNMOD  TSV6
       101300                           TSV6::
 11067
 11068                                          .SBTTL  HARDWARE PARAMETER CODING SECTION
 11069
 11070                                  ;++
 11071                                  ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
 11072                                  ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
```

HARDWARE PARAMETER CODING SECTION

```
11073                                   ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
11074                                   ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
11075                                   ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
11076                                   ; WITH THE OPERATOR.
11077                                   ; -
11078 101300                                    BGNHRD
      101300  000010                            .WORD L10100-L$HARD/2
      101302                            L$HARD::
11079
11080 101302                                    GPRMA   HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
      101302  000031                            .WORD   T$CODE
      101304  101322                            .WORD   HPM1
      101306  160010                            .WORD   T$LOLIM
      101310  177776                            .WORD   T$HILIM
11081 101312                                    GPRMA   HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
      101312  001031                            .WORD   T$CODE
      101314  101356                            .WORD   HPM2
      101316  000000                            .WORD   T$LOLIM
      101320  000776                            .WORD   T$HILIM
11082                                           ;GPRMD  HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
11083 101322                                    ENDHRD
                                                .EVEN
      101322                            L10100:
11084 101322      104     105     126   HPM1:   .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
11085 101356      111     116     124   HPM2:   .ASCIZ  'INTERRUPT VECTOR    '
11086 101402      111     116     124   HPM3:   .ASCIZ  'INTERRUPT PRIORITY     '
11087                                           .EVEN
```

SOFTWARE PARAMETER CODING SECTION

```
11089                                           .SBTTL  SOFTWARE PARAMETER CODING SECTION
11090
11091                                   ;++
11092                                   ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
11093                                   ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
11094                                   ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
11095                                   ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
11096                                   ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
11097                                   ; WITH THE OPERATOR.
11098                                   ;--
11099 101432                                      BGNSFT
       101432  000003                            .WORD L10101-L$SOFT/2
       101434                          L$SOFT::
11100                                   ;          GPRML   SPM1,0,-1,YES                   ; GET TRANSPORT TEST FLAG.
11101 101434                                      GPRML   SPM4,2,-1,YES                   ; GET ITERATION CONTROL.
       101434  001130                            .WORD   T$CODE
       101436  101472                            .WORD   SPM4
       101440  177777                            .WORD   -1
11102                                   ;          GPRMD   SPM6,4,D,7777,0,7777,YES        ; GET LOCAL ERROR LIMIT
11103                                   ;          GPRMD   SPM7,6,D,7777,0,7777,YES        ; GET GLOBAL ERROR LIMIT
11104 101442                                      ENDSFT
                                                  .EVEN
       101442                          L10101:
11105
11106 101442     105     116     101  SPM1:     .ASCIZ  'ENABLE TRANSPORT TESTS '
11107 101472     111     116     110  SPM4:     .ASCIZ  'INHIBIT ITERATIONS     '
11108 101522     120     105     122  SPM6:     .ASCIZ  'PER TEST ERROR LIMIT   '
11109 101552     120     105     122  SPM7:     .ASCIZ  'PER UNIT ERROR LIMIT   '
11110                                           .SBTTL  PATCH AREA
11111
11112                                   ;
11113                                   ; FINALLY A GENEROUS PATCH AREA.
11114                                   ;
11115                                   ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
11116                                   ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
11117                                   ;
11118
11119 101602                            PATCH::
11120
11121 101602                                      .BLKW   32.
11122
11124           102000                            .=.!377+1
11126 102000                                      LASTAD          ;SET LAST USED ADDRESS.
                                                  .EVEN
       102000  000000                            .WORD   0
       102002  000000                            .WORD   0
       102004                          L$LAST::
11127 102004                                      ENDMOD
11128           000001                            .END
```

Symbol table

```
ADDSSR  012166 G        C$AUTO= 000061        DEVNRD  022705        FUSI    004121        INTMAS  016204
ADR   = 000020 G        C$BRK = 000022        DEVNXR  022623        F$AU  = 000015        INTR    016256 G
AMBTSS  006637          C$BSEG= 000004        DEVONL  022553        F$AUTO= 000020        INTREC  002224 G
ASSEMB= 000010          C$BSUB= 000002        DEVSUM  022516        F$BGN = 000040        INTVEC  016206
A1716 = 000003          C$CEFG= 000045        DFPTBL  002156 G      F$CLEA= 000007        INTX    004302
BADDAT  003160 G        C$CLCK= 000062        DIAGMC= 000000        F$DU  = 000016        INVERT  020504 G
BADSSR  015740 G        C$CLEA= 000012        DICEB = 000001        F$END = 000041        IOKCKI= 000200
BDVPCR= 177520 G        C$CLOS= 000035        DSBINT  016244        F$HARD= 000004        IOKSTP= 000001
BENBSW  002232 G        C$CLP1= 000006        DUAD12  004645        F$HW  = 000013        IPRI    002212 G
BIE   = 040000          C$CVEC= 000036        DUFLG   003114 G      F$INIT= 000006        ISR   = 000100 G
BIT0  = 000001 G        C$DCLN= 000044        DUMMY   003064        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  017037        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  016212        F$SRV = 000010        I$MOD = 000041
BIT08 = 000400 G        C$ERSO= 000057        ENVIRN  020152        F$SUB = 000002        I$MSG = 000041
BIT09 = 001000 G        C$ESCA= 000010        EPRTSW  002200 G      F$SW  = 000014        I$PROT= 000040
BIT1  = 000002 G        C$ESEG= 000005        EPRT1   006362        F$TEST= 000001        I$PTAB= 000041
BIT10 = 002000 G        C$ESUB= 000003        EPRT2   006362        GDDAT   003162 G      I$PWR = 000041
BIT11 = 004000 G        C$ETST= 000001        ERCM    011773        GERRMA  002174 G      I$RPT = 000041
BIT12 = 010000 G        C$EXIT= 000032        ERRHI   002240 G      GETPAT  017516 G      I$SEG = 000041
BIT13 = 020000 G        C$GETB= 000026        ERRK    017016        GETSEL  017600 G      I$SETU= 000041
BIT14 = 040000 G        C$GETW= 000027        ERRLO   002242 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$GPLO= 000030        ERTABE  003400        G$EXCP= 000400        I$TST = 000041
BIT4  = 000020 G        C$GPRI= 000040        ERTABL  003200        G$HILI= 000002        J$JMP = 000167
BIT5  = 000040 G        C$INIT= 000011        ESUM    017020        G$LOLI= 000001        KIPAR0= 172340
BIT6  = 000100 G        C$INLP= 000020        EVL   = 000004 G      G$NO  = 000000        KIPAR1= 172342
BIT7  = 000200 G        C$MAINT= 000050       EXBCNT= 000010        G$OFFS= 000400        KIPAR2= 172344
BIT8  = 000400 G        C$MEM = 000031        EXFMSG  101137        G$OFSI= 000376        KIPAR3= 172346
BIT9  = 001000 G        C$MSG = 000023        EXPBRE  015542 G      G$PRMA= 000001        KIPAR4= 172350
BOE   = 000400 G        C$OPEN= 000034        EXPD    002234 G      G$PRMD= 000002        KIPAR5= 172352
BRINIT  004461          C$PNTB= 000014        EXPGOT  004535        G$PRML= 000000        KIPAR6= 172354
BSEL0 = 000000          C$PNTF= 000017        EXPGT2  004571        G$RADA= 000140        KIPAR7= 172356
BSEL1 = 000001          C$PNTS= 000016        EXPMSG  002324 G      G$RADB= 000000        KIPDR0= 172300
CHKAMB  016104          C$PNTX= 000015        EXPREC  015534 G      G$RADD= 000040        KIPDR1= 172302
CHKMAN  020022 G        C$QIO = 000377        EXTA    005774        G$RADL= 000120        KIPDR2= 172304
CHKTSS  016376          C$ROBU= 000007        EXTEND  005772        G$RADO= 000020        KIPDR3= 172306
CKDROP  017242          C$REFG= 000047        EXTFEA  002226 G      G$XFER= 000004        KIPDR4= 172310
CKEMAX  017142          C$RESE= 000033        E$END = 002100        G$YES = 000010        KIPDR5= 172312
CKMSG   011420 G        C$REVI= 000003        E$LOAD= 000035        HIADDR= 001400        KIPDR6= 172314
CKMSG2  011540 G        C$RFLA= 000021        FATERR= 000060        HOE   = 100000 G      KIPDR7= 172316
CKRAM   011154 G        C$RPT = 000025        FATFLG  002222 G      HPM1    101322        KTENAB  003136 G
CKRAM2  011264 G        C$SEFG= 000046        FERCM   011762        HPM2    101356        KTFLG   003134 G
CMDPKT  020650 G        C$SPRI= 000041        FIFEXP  012230 G      HPM3    101402        KTINIT  020332
CONFIG  017310          C$SVEC= 000037        FIF1MS  012302        IBE   = 010000 G      KTOFF   017334
COUNT   002312 G        C$TPRI= 000013        FIF2MS  012351        IDU   = 000040 G      KTON    017316
CSRADD  002206 G        DATA    002314 G      FNOINT  004217        IER   = 020000 G      LERRMA  002172 G
CTAB    003166 G        DATASC  017554        FORCER  002176 G      IFAULT  004260        LERRNO= 000000
CTABE   003200 G        DEBUGM  011672        FREE    003126 G      INCERK  017104        LISTAL= 000001
CTABM   003166 G        DEVCNT  002220 G      FREEHI  003132        INTCPC  016210        LOE   = 040000 G
C$AU  = 000052          DEVDRO  022766        FRESIZ  003130 G      INTFLA  016205        LOOPCN  002216 G
```

J4

Symbol table

```
LOOPCO  013166          L10001  002176          L10073  066254          ONEFIL= 000000          PRMSGE  014612 G
LOOPFL  003164 G        L10002  005770          L10074  064414          O$APTS= 000000          PRMSGO  014772
LOT   = 000010 G        L10003  012104          L10075  074112          O$AU  = 000001          PRMSG1  015037
L$ACP   002110 G        L10004  012122          L10076  077622          O$BGNR= 000001          PRMSG2  015075
L$APT   002036 G        L10005  012140          L10077  101276          O$BGNS= 000001          PROASC  014460
L$AU    021742 G        L10006  012146          L10100  101322          O$DU  = 000001          PR1ASC  014525
L$AUT   002070 G        L10007  012164          L10101  101442          O$ERRT= 000000          PST32W  003154 G
L$AUTO  022146 G        L10010  012202          MEMADD  014014 G        O$GNSW= 000001          PUNIT   021674
L$CCP   002106 G        L10011  012226          MEMCK   020666 G        O$POIN= 000001          PW.D11= 000021
L$CLEA  022226 G        L10012  012300          MENASC  017771          O$SETU= 000000          PW.D13= 000022
L$CO    002032 G        L10013  012450          MENERR  017716          PASRPT  021444          PW.D22= 000020
L$DEPO  002011 G        L10014  013164          MENRES  020020          PATCH   101602 G        PW.NOP= 000000
L$DESC  003412 G        L10015  014012          MIMENU  072772          PATDAT  017552          PW.NO1= 000023
L$DESP  002076 G        L10016  014034          MMVEC = 000250          PC.ERA= 002400          PW.RDE= 000024
L$DEVP  002060 G        L10017  015540          MSA.FR= 000006          PC.IER= 002000          PW.RDR= 000001
L$DISP  002124 G        L10020  015546          MSA.NO= 000000          PC.NOO= 001000          PW.RDS= 000005
L$DLY   002116 G        L10021  015554          MSA.NR= 000004          PC.REL= 000000          PW.RFI= 000003
L$DTP   002040 G        L10022  015566          MSA.VO= 000002          PC.REW= 000400          PW.WCT= 000006
L$DTYP  002034 G        L10023  015610          MSGEXP  012204 G        PKBCNT= 000006          PW.WFI= 000004
L$DU    022040 G        L10024  015636          MSGLOO  013124 G        PKHI  = 000004          PW.WFM= 000007
L$DUT   002072 G        L10025  015776          MSGSTA  012410 G        PKLOW = 000002          PW.WMI= 000010
L$DVTY  003404 G        L10026  016306          MSGSUB  014002 G        PKTADD  007556          PW.WNP= 000011
L$EF    002052 G        L10030  021672          MS.ATT= 000006          PKTFRM  007520          PW.WTR= 000002
L$ENVI  002044 G        L10031  022036          MS.EXT= 000200          PKTGET  012124 G        P.ACK = 100000
L$ETP   002102 G        L10032  022144          MS.RSD= 000001          PKTMES  012150 G        P.CMD = 000037
L$EXP1  002046 G        L10033  022224          MS.RSF= 000020          PKTRAM  004747 G        P.CONT= 000012
L$EXP4  002064 G        L10034  022252          MS.RST= 000010          PKTSSR  012106 G        P.CVC = 040000
L$EXP5  002066 G        L10035  022514          M8186   005556          PNT   = 001000 G        P.FMT = 000140
L$HARD  101302 G        L10036  024014          M8189   005647          PRAMPK  014036          P.FORM= 000011
L$HIME  002120 G        L10037  026006          NBA   = 002000          PRASC   014563          P.GETS= 000017
L$HPCP  002016 G        L10040  024270          NEWPAS  021400          PR8EXP  015530          P.IE  = 000200
L$HPTP  002022 G        L10041  024534          NODEV   003116 G        PR8MSG  015376          P.INIT= 000013
L$HW    002156 G        L10042  031402          NOEXTF  027576          PR8REC  015532          P.MODE= 007400
L$ICP   002104 G        L10043  026362          NOINIT  004337          PR8TOT  015463          P.OPP = 020000
L$INIT  021146 G        L10044  026652          NOINTR  004223          PR8YTE  015162 G        P.POSI= 000010
L$LADP  002026 G        L10045  027150          NOITS   002170 G        PRI   = 002000 G        P.READ= 000001
L$LAST  102004 G        L10046  027602          NOMAN   020056          PRIADD  010162          P.SWB = 010000
L$LOAD  002100 G        L10047  034172          NOMEM   005462          PRIAO   010232          P.WRIT= 000005
L$LUN   002074 G        L10050  031736          NP.IR = 000200          PRIBXO  007614 G        P.WRTC= 000004
L$MREV  002050 G        L10051  032330          NP.LOO= 000040          PRIEQU  010062          P.WRTS= 000006
L$NAME  002000 G        L10052  032736          NP.OUT= 000100          PRIPKT  007372 G        QVP     002204 G
L$PRIO  002042 G        L10053  037764          NP.WRP= 000020          PRIRAM  010070          RAMASC  014216
L$PROT  021136 G        L10054  035256          NSI     004154          PRITAD  010276          RAMDAT  002244 G
L$PRT   002112 G        L10055  036210          NSINIT  004411          PRITSS  006026          RAMERR  015550 G
L$REPP  002062 G        L10056  050076          NUL     004531          PRITO   010360          RAMEXP  015570 G
L$REV   002010 G        L10057  040270          NULCR   004532          PRIT1   010423          RAMFOR  010120
L$RPT   022254 G        L10060  041500          NUMENU  075402          PRIXOR  007744 G        RAMSIZ  002304 G
L$SOFT  101434 G        L10061  043040          NXM   = 004000          PRI00 = 000000 G        RAMTAD  015556 G
L$SPC   002056 G        L10062  043416          NXMFLG  003140 G        PRI01 = 000040 G        RCVHIA  002306 G
L$SPCP  002020 G        L10063  044670          NXMHI   003144 G        PRI02 = 000100 G        RCVLOA  002310 G
L$SPTP  002024 G        L10064  045734          NXMLO   003142 G        PRI03 = 000140 G        RDERR   005210
L$STA   002030 G        L10065  051356          NXMTST  021042          PRI04 = 000200 G        RECMSG  002470 G
L$SW    002166 G        L10066  062204          NXR     003742          PRI05 = 000240 G        RECV    002236 G
L$TEST  002114 G        L10067  053362          NXRERR  005740 G        PRI06 = 000300 G        REGSAV  017462
L$TIML  002014 G        L10070  054650          NXRX    004001          PRI07 = 000340 G        RETERR  005374
L$UNIT  002012 G        L10071  056120          NXTU    021412          PRMESS  014302          REV     002230 G
L10000  002164          L10072  057364          OFL   = 000100          PRMNO   002322 G        REWIND  011054 G
```

K4

SEQ 0256

Symbol table

```
RMCHBE= 000167          S1.IID= 004000          TST40I  101262          T10     066256 G        T15BFR  033002
RMCHEN= 000200          S1.I1R= 020000          TSV2    002000 G        T11     074114 G        T15BF2  033470
RMMSGB= 000215          S1.I2R= 040000          TSV3    002176 G        T12     077624 G        T15BS0  033470
RMMSGE= 000234          S1.PAR= 100000          TSV4    021136 G        T12BFR  027672          T15BS1  033471
RMPKTB= 000201          S2.ATI= 000010          TSV5    023036 G        T12BKG  030347          T15DAT  032770
RMPKTE= 000210          S2.BTI= 000004          TSV6    101300 G        T12BLK  027724          T15L00  031434
RMR   = 010000          S2.DIM= 000200          TTIBFR= 177562 G        T12CHA  031340          T15PAC  032760
RWPACK  011150          S2.ILW= 000100          TTICSR= 177560 G        T12CKR  031120 G        T15PK2  033460
SC    = 100000          S2.INR= 000020          TTION   100342          T12CON  030726          T15RES  034062
SCE   = 020000          S2.OUT= 000040          TTION2  071036          T12DAT  027660          T15RT2  034134
SCHERR  005302          S2.UND= 000003          TTIVEC= 000060 G        T12DPR  030526          T15SSR  033476
SCME    005015          TBLEND= 003064 G        TVECSA  100344          T12GET  030106          T15S2   033472
SCMENU  100350          TCOASC  006500          TVSAV2  071040          T12HIA  027712          T15S3   033474
SDELAY  010660          TCOCOD  006700          T$ARGC= 000001          T12KT   027720          T16BEN  037650
SELASC  017764          TEMP1   003120 G        T$CODE= 001130          T12L0A  027714          T16BFR  037622
SELDAT= 000004          TEMP2   003122 G        T$ERRN= 002261          T12L00  026036          T16BFS  037642
SEL2  = 000002          TERCLS= 000016          T$EXCP= 000000          T12MSG  030251          T16CLR  037466
SETMAP  017356          TESTNO= 000014          T$FLAG= 000040          T12NIN  030435          T16DAT  037610
SETU    021476          TEXASC  006437          T$GMAN= 000000          T12NXM  030617          T16DT2  037660
SFFMSG  012142 G        TFCASC  006541          T$HILI= 000776          T12PAC  027650          T16D01  036242
SFHERR  003707          TIMEXP  015612 G        T$LAST= 000001          T12PAR  027716          T16D28  036244
SFIERR  003654          TIMSGO  015640          T$LOLI= 000000          T12SET  031276          T16D53  036246
SFIMSG  012074 G        TINERR  012061          T$LSYM= 010000          T12SWR  031230          T16D78  036250
SFPTBL  002166 G        TMPBFR  002634 G        T$LTNO= 000014          T12TBE  030056          T16LEN  036572
SIFLAG  003156 G        TNAM    016744          T$NEST= 177777          T12WRT  030162          T16L00  034212
SIMSG   012026          TPRISA  100346          T$NS0 = 000000          T121L0  026150          T16PAC  037600
SKIPT   003402          TPSAV2  071042          T$NS1 = 000005          T122L0  026504          T16PK2  037650
SOFINI  016034 G        TRANST  002166 G        T$NS2 = 000002          T123L0  026724          T16REJ  036704
SPACE   010470 G        TSBA  = 000000 G        T$NS3 = 000003          T124L0  027346          T16RES  037420
SPM1    101442          TSBAH = 000001 G        T$PTNU= 000000          T124TS  027722          T16SEX  037560
SPM4    101472          TSDB  = 000000 G        T$SAVL= 177777          T13BFR  023452          T16SRD  037512
SPM6    101522          TSDBH = 000001 G        T$SEGL= 177777          T13DAT  023440          T16SSR  036322
SPM7    101552          TSFCOD  007240          T$SEKO= 010000          T13L00  023054          T16TSB  036470
SR0   = 177572          TSREJ = 000006          T$SUBN= 000000          T13MEM  023545          T16T01  037021
SR1   = 177574          TSSDEF  006610          T$TAGL= 177777          T13NBA  023604          T16T28  037120
SR2   = 177576          TSSR  = 000002 G        T$TAGN= 010102          T13PAC  023430          T16T53  037220
SR3   = 172516          TSSRBI  003504 G        T$TEMP= 000000          T13RES  023746          T16T78  037320
SSR   = 000200          TSSRFO  006417          T$TEST= 000014          T13SSR  023657          T16WMI  037532
STATCO  012452          TSSRH = 000003 G        T$TSTM= 177777          T14BFR  024566          T162SS  036357
SVCGBL= 000000          TSSX    004022          T$TSTS= 000001          T14BS0  024560          T163SS  036423
SVCINS= 000000          TSTBLK  002754 G        T$$AU = 010031          T14BS1  024561          T17BEN  047752
SVCSUB= 000001          TSTCNT  002214 G        T$$AUT= 010033          T14BS2  024562          T17BFR  047632
SVCTAG= 000000          TSTEND  016760          T$$CLE= 010034          T14DAT  024560          T17BFS  047652
SVCTST= 000001          TSTFLA  002316 G        T$$DU = 010032          T14DTA  025200          T17CLE  047536
S$LSYM= 010000          TSTL00  016516 G        T$$HAR= 010100          T14L00  024034          T17CLR  047350
SO.IDB= 000010          TSTPTR  002320 G        T$$HW = 010000          T14NBA  025212          T17DAT  047620
SO.IFB= 000002          TSTSET  016550 G        T$$INI= 010030          T14NIN  025453          T17DT2  047770
SO.IFP= 000001          TST12I  030060          T$$MSG= 010025          T14PAC  024550          T17EXE  046114
SO.ILD= 000020          TST13I  023472          T$$PRO= 010027          T14PK2  025170          T17EXP  045772
SO.ION= 000040          TST14I  025631          T$$RPT= 010035          T14RES  025676          T17EXS  046012
SO.IRD= 000100          TST15I  034043          T$$SEG= 010000          T14RST  025734          T17L00  040014
SO.IRW= 000004          TST16I  036252          T$$SOF= 010101          T14SSR  025363          T17MSK  045766
SO.ISP= 000200          TST17I  046216          T$$SRV= 010026          T14TSB  025545          T17PAC  047610
S1.ICE= 002000          TST18I  050606          T$$SUB= 010074          T142RE  025266          T17PK2  047760
S1.IEO= 010000          TST19I  057572          T$$SW = 010001          T15AM2  033552          T17RFI  047516
S1.IFM= 001000          TST20I  064572          T$$TES= 010077          T15AM3  033653          T17RSF  047414
S1.IHE= 000400          TST39I  077574          T1      023036 G        T15AM4  033755          T17SET  047556
```

L4

Symbol table

```
T17SNP  047436          T19SSR  057633          T3      026010 G        T39NBA  077075          WC.ISR= 000020
T17SRO  047374          T19WCT  061532          T38FLG  003152 G        T39NE   076373          WF.IED= 000010
T17SSR  046235          T19WFI  061476          T3.1    026036          T39NFL  076370          WF.IER= 000004
T17WFD  046114          T19WFM  061552          T3.2    026376          T39OFF  077436          WF.IHI= 000200
T17WFI  047462          T19WST  061133          T3.3    026666          T39OF2  076634          WF.IRE= 000040
T171CM  046555          T191CM  060270          T3.4    027164          T39ON   077445          WF.IWF= 000020
T172CM  046637          T192CM  060352          T38ASC  074045          T39ON2  076700          WF.IWR= 000100
T172SS  046272          T192SS  057670          T38ASN  074064          T39OUT  077241          WF.I3R= 000002
T173CM  046733          T193CM  060446          T38ASO  073742          T39PAC  075620          WF.I4R= 000001
T173SS  046336          T193SS  057734          T38AS1  074007          T39PK2  076320          WRTCHR  010662 G
T174CM  047017          T194SS  060001          T38BFR  071066          T39PK3  076350          WRTERR  005115
T174SS  046403          T195CM  060534          T38BSO  071060          T39PK4  076360          WRTMSG  005060
T175CM  047102          T195SS  060044          T38BS1  071061          T39RES  077536          WSMBK   020660 G
T175SS  046446          T196CM  060610          T38BS2  071062          T39RL   077532          XFERAS  016000
T176CM  047156          T196SS  060110          T38CNT  074110          T39SBS  077364          XNXM    016436
T176SS  046512          T197CM  060673          T38DAT  073400          T39SFS  077312          XORBFO  007676
T177CM  047264          T197SS  060153          T38DLY  071044          T39SIZ  076366          XORFOR  010014
T18BFR  051312          T198CM  060761          T38DSW  071570          T39SSR  077151          XST0  = 000006 G
T18CMP  051013          T198SS  060222          T38DTA  071560          T39TAD  075630          XST1  = 000010 G
T18DAT  051300          T199CM  061050          T38EAI  071566          T39WPN  077016          XST2  = 000012 G
T18DT2  051350          T2      024016 G        T38EB   071542          T39WR   076362          XST3  = 000014 G
T18EXP  050556          T2.1    024034          T38ID   072745          T39WRT  076743          XST4  = 000016 G
T18LOO  050116          T2.2    024304          T38INT  072336          T4      031404 G        XSOBOT= 000002
T18MSK  050552          T20BEN  066132          T38MBP  073440          T4.1    031434          XSOEOT= 000001
T18PAC  051270          T20BFR  066012          T38MSG  072705          T4.2    031740          XSOIE = 000040
T18PK2  051340          T20BFS  066032          T38MS1  072546          T4.3    032332          XSOILA= 000400
T18SET  051160          T20CLE  065724          T38MS2  072641          T40NE   101215          XSOILC= 001000
T18SMI  051136          T20CLR  065512          T38MS3  071705          T5      034174 G        XSOLET= 020000
T18SRO  051116          T20CWP  065417          T38NBA  072172          T5.1    034212          XSOMOT= 000200
T18SSR  050645          T20DAT  066000          T38NE   071630          T5.2    035272          XSONEF= 002000
T18XS   050602          T20DT2  066150          T38OFL  072422          T6      037766 G        XSOONL= 000100
T182SS  050702          T20EXE  064572          T38ONL  072366          T6.1    040014          XSOPED= 000010
T183SS  050746          T20EXP  064452          T38PAC  071050          T6.2    040304          XSORLL= 010000
T198EN  062062          T20EXS  064472          T38PK2  071550          T6.3    041514          XSORLS= 040000
T198FC  057406          T20LOO  062224          T38PK3  071600          T6.4    043054          XSOTMK= 100000
T198FR  061742          T20MSK  064446          T38PK4  071620          T6.5    043432          XSOVCK= 000020
T198FS  061762          T20PAC  065770          T38RES  073402          T6.6    044704          XSOWLE= 004000
T19CLE  061610          T20PK2  066140          T38SIZ  071626          T7      050100 G        XSOWLK= 000004
T19CLR  061344          T20RFI  065576          T38SSR  072246          T8      051360 G        XXCOMM  003124 G
T19CNV  061654          T20RSF  065316          T38SST  072456          T8.1    051376          X$ALWA= 000000
T19DAT  061730          T20SET  065744          T38TAD  071060          T8.2    053376          X$FALS= 000040
T19DT2  062100          T20SEX  065706          T38WLE  072125          T8.3    054664          X$OFFS= 000400
T19EXE  057572          T20SRO  065536          T38WR   071622          T8.4    056134          X$TRUE= 000020
T19EXP  057452          T20SSR  064621          T38WRL  072064          T9      062206 G        X1.COR= 020000
T19EXS  057472          T20SWP  065207          T38WRT  072000          T9.1    062224          X1.DLT= 100000
T19LOO  051376          T20WFI  065616          T398FR  075636          UAM   = 000200 G        X1.MBZ= 017375
T19MSK  057446          T20WFM  065652          T398SO  075630          UNITN   002202 G        X1.RBP= 000400
T19PAC  061720          T20WMI  065672          T398S1  075631          UNREC = 000006          X1.SPA= 040000
T19PK2  062070          T20WNP  065556          T398S2  075632          USI     004125          X1.UNC= 000002
T19PRE  057404          T202SS  064656          T39DAT  077534          WAITF   016310 G        X2.BUF= 000100
T19RFI  061456          T203SS  064722          T39DLY  075612          WC.IFA= 000200          X2.EXT= 000200
T19RSF  061410          T204SS  064767          T39DSW  076340          WC.IFE= 000002          X2.OPM= 100000
T19RST  061240          T205SS  065032          T39DTA  076330          WC.IGO= 000001          X2.RCE= 040000
T19SET  061630          T206SS  065076          T39EAI  076336          WC.IRE= 000010          X2.REV= 000077
T19SEX  061572          T208SS  065141          T39ETN  076541          WC.IRW= 000004          X2.SPA= 035400
T19SNP  061432          T23A    003146 G        T39ETS  076452          WC.IOT= 000100          X2.UNI= 000007
T19SRO  061370          T23B    003150 G        T39MCL  077454          WC.I1T= 000040          X2.WCF= 002000
```

M4
TSV6 - PARAMETER CODING MACRO V05.03 Tuesday 28-Apr-87 08:29 Page 85-5
SEQ 0258

Symbol table

```
X3.DCK= 000010          X3.OPI= 000100          X3.SPA= 000200          X4.MBZ= 017400          X4.TSM= 020000
X3.MBZ= 000006          X3.REV= 000040          X3.TRF= 000020          X4.RCE= 040000          X4.WRC= 000377
X3.MDE= 177400          X3.RIB= 000001          X4.HSP= 100000

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

*** Assembler statistics


Work  file  reads: 398
Work  file writes: 363
Size of work file: 32288 Words  ( 127 Pages)
Size of core pool: 19684 Words  ( 75 Pages)
Operating  system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:10:06.20
CVTSBE,CVTSBE/-SP=SVC/ML,CVTSBE
```