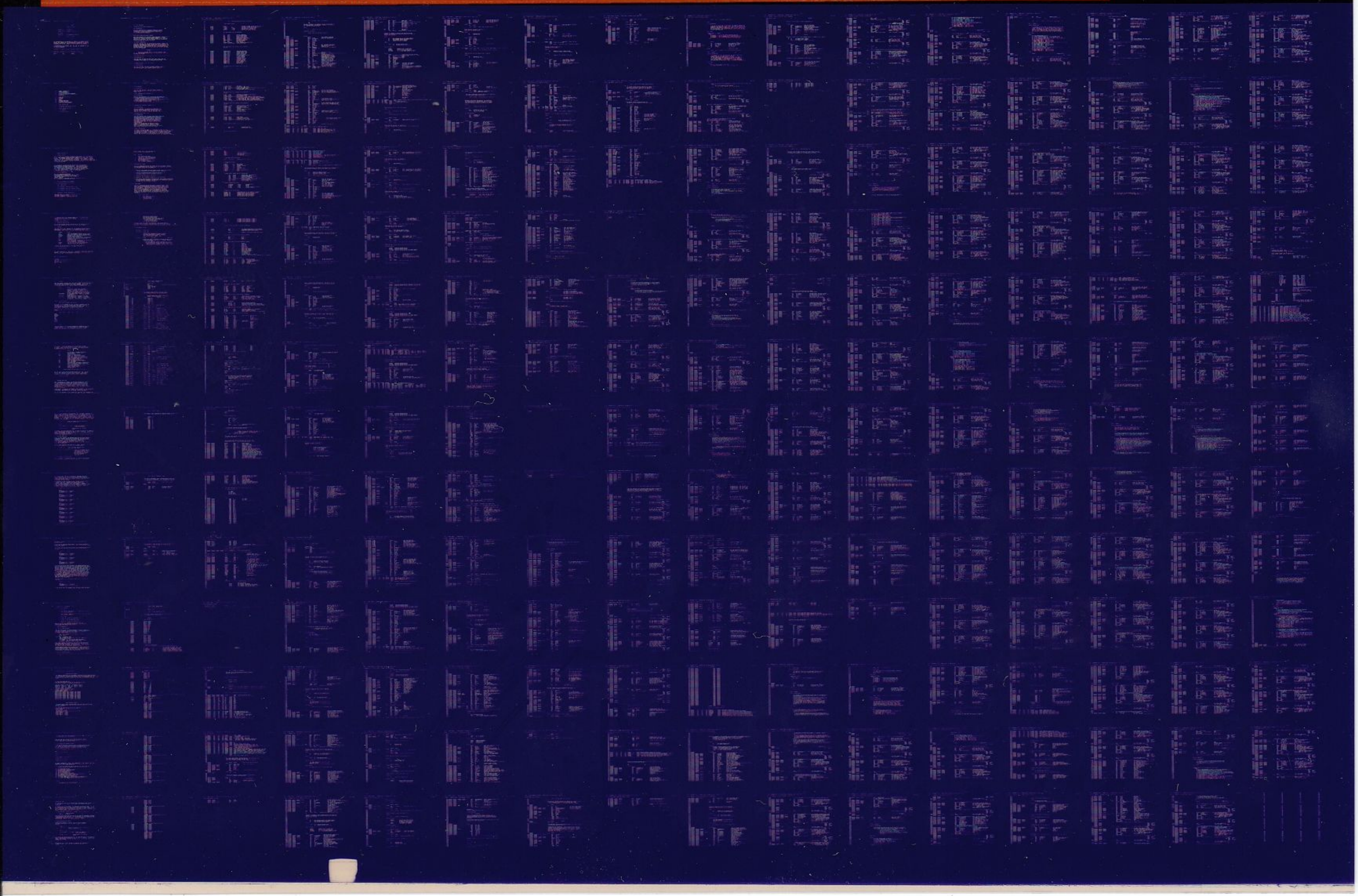
11/21+ TSV05 CTRL LTZ
TSV05 CNTSBA0

AH-T817A-MC
FICHE 01 OF 02

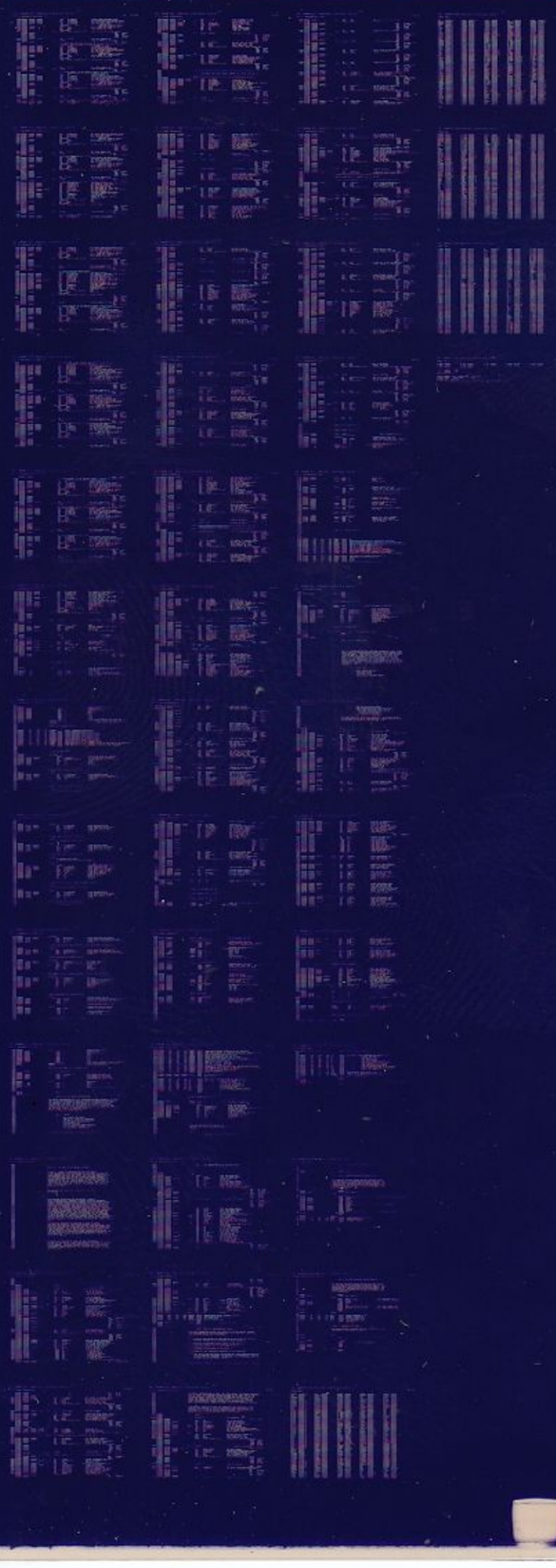
JUL 1984
digital
Made In USA



11/Z1+ TSV05 CTRL LTZ
TSV05 CNTSBA0

AH-T817A-MC
FICHE 02 OF 02

JUL 1984
digital
Made In USA



S

.REM_

IDENTIFICATION

PRODUCT ID: AC-T816A-MC

PRODUCT TITLE: CNTSBAO TSV05 CTRL LT2

DECO/DEPO: 1.0

DEPARTMENT: ISS/DIAGNOSTIC SERVICES

DATE: APRIL 09, 1984

TABLE OF CONTENTS

1.0 GENERAL INFORMATION
1.1 PROGRAM ABSTRACT
1.2 SYSTEM REQUIREMENTS
1.3 RELATED DOCUMENTS AND STANDARDS
1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
1.5 ASSUMPTIONS

2.0 OPERATING INSTRUCTIONS
2.1 COMMANDS
2.2 SWITCHES
2.3 FLAGS
2.4 HARDWARE QUESTIONS
2.5 SOFTWARE QUESTIONS
2.6 EXTENDED P-TABLE DIALOGUE
2.7 QUICK STARTUP PROCEDURE

3.0 ERROR INFORMATION

4.0 PERFORMANCE AND PROGRESS REPORTS

5.0 DEVICE INFORMATION TABLES

6.0 TEST SUMMARIES

7.0 MAINTENANCE HISTORY

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS IS A SBC-11/21+ RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF A TSV05 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A SBC-11/21+ 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

SBC-11/21+ 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. XXDP+ USERS GUIDE
2. TSV05 TRANSPORT SUBSYSTEM USER'S GUIDE
3. TSV05 TRANSPORT SUBSYSTEM TECHNICAL MANUAL
4. TSV05 TRANSPORT SUBSYSTEM INSTALLATION MANUAL

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

FUNCTIONAL SBC-11/21+ 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.
CNTSAA 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.

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 SBC-11/21+ DIAGNOSTIC SUPERVISOR COMPATIBLE
PROGRAM. ALL LOADING AND RUNTIME INSTRUCTIONS CAN BE REFERENCED IN THE
XXDP+ USERS GUIDE. THE USER ENTRY IS IN QUOTES.

BOOT THE DIAGNOSTIC MEDIA

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

2.2 SWITCHES

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

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

EXAMPLE OF SWITCH USAGE:

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

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

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

	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 CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
PNT	PRINT TEST NUMBER AS TEST EXECUTES
BOE	"BELL" ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE
LOT	LOOP ON TEST

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

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

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

2.4 HARDWARE QUESTIONS

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

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

ON A "N" (NO) RESPONSE TO THE "CHANGE HW?" QUESTION, THE DIAGNOSTIC WILL RUN USING THE DEFAULT VALUES FOR ALL QUESTIONS. THE DEFAULT ADDRESS AND VECTOR ARE:

TSBA/TSDB = 176000, VECTOR = 224

ON A "Y" (YES) RESPONSE TO THE QUESTION, THE FOLLOWING QUESTIONS WILL THEN BE ASKED TO ALLOW THE OPERATOR TO SELECT THE UNITS TO BE TESTED. A VALUE, IF PRESENT, LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN IF ONLY A CARRIAGE RETURN IS TYPED AS A RESPONSE. A "(D)" IN A QUESTION INDICATES THAT A DECIMAL NUMBER IS REQUIRED AS A RESPONSE. AN "(O)" INDICATES AN OCTAL NUMBER IS BEING SOLICITED. AN "(L)" INDICATES THAT A LOGICAL RESPONSE IS TO BE MADE: "Y" FOR YES, "N" FOR NO.

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

UNIT 0

DEVICE ADDRESS (O) 176000 ? <ENTER THE ADDRESS OF THE TSBA/TSDB REGISTER>

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

THE ADDRESS AND VECTOR QUESTIONS WILL BE ASKED FOR EACH OF THE NUMBER OF UNITS (CONTROLLERS) SPECIFIED IN THE "# UNITS?" QUESTION. LOGICAL UNIT NUMBERS ARE ASSIGNED IN ORDER, BEGINNING AT 0. UP TO FOUR UNITS CAN BE SELECTED FOR TESTING AS FOLLOWS:
UP TO 4 TSV05 CONTROLLERS PER 11/21+ AND UP TO 2 DRIVES PER CONTROLLER

2.5 SOFTWARE QUESTIONS

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

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

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

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

2.6 EXTENDED P-TABLE DIALOGUE

WHEN YOU ANSWER THE HARDWARE QUESTIONS, YOU ARE BUILDING ENTRIES IN A TABLE THAT DESCRIBES THE DEVICES UNDER TEST. THE SIMPLEST

WAY TO BUILD THIS TABLE IS TO ANSWER ALL QUESTIONS FOR EACH
UNIT TO BE TESTED. IF YOU HAVE A MULTIPLEXED DEVICE SUCH AS
A MASS STORAGE CONTROLLER WITH SEVERAL DRIVES OR A COMMUNICATION
DEVICE WITH SEVERAL LINES, THIS BECOMES TEDIOUS SINCE MOST OF
THE ANSWERS ARE REPETITIOUS.

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

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

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

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

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

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

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

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

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

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

NOTICE THAT THE DEFAULT VALUE FOR THE Q-FACTOR CHANGES WHEN A

NON-DEFAULT RESPONSE IS GIVEN. BE CAREFUL WHEN SPECIFYING MULTIPLE UNITS!

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

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

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

UNIT 1
CSR ADDRESS (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
CNTSB 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.

```
CNTSB 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.

```
CNTSB 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 (SBC-11/21+)

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

 UNITS (D) ?  1

 UNIT 0

 DEVICE ADDRESS  (O) 176000 ? <CR>

 VECTOR (O) 22^ ? <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 FALCON + PROCESSOR.

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).

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

Q.V.	2 MINS 15 SECONDS
DEFAULT	12 MINUTES

MORE EXHAUSTIVE CHECKS ARE AVAILABLE BY ALLOWING THE DIAGNOSTIC PROGRAMS TO RUN FOR MORE THAN ONE PASS. THE SECOND PASS OF THE PROGRAM IS MORE COMPREHENSIVE THAN THE FIRST PASS. ALL ITERATIONS AFTER THE FIRST PASS ARE THE SAME, HOWEVER, THEY ARE SUBSTANTIALLY LONGER.

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) 176000 ? <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 11/21+ 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).

    CVTSBBO => CNTSBAO          JAKI BERG                    9-APR-1984

                 CHANGES WERE MADE TO CVTSBBO TO PRODUCE CNTSBAO FOR THE
                 FALCON-PLUS PROJECT (SBC-11/21+).  CHANGES, MARKED BY
                 ";JB REV A-0", ARE:
                   - SET THE ODT BREAK VECTOR (LOCATION 140) TO THE
                   STARTING ADDRESS OF FALCON'S ODT ROM (170000-OCTAL).
                   - LOWER THE GENERAL INTERRUPT PRIORITY FROM 7 TO 6.
                   - CHANGE DEFAULT CSR ADDRESS FROM 172540 TO 176000.
```

```
                                .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  CNTSB,B,0,655.,0
002000                  L$NAME::                ;DIAGNOSTIC NAME
002000  103                     .ASCII /C/
002001  116                     .ASCII /N/
002002  124                     .ASCII /T/
002003  123                     .ASCII /S/
002004  102                     .ASCII /B/
002005  000                     .BYTE   0
002006  000                     .BYTE   0
002007  000                     .BYTE   0
002010                  L$REV::                 ;REVISION LEVEL
002010  102                     .ASCII  /B/
002011                  L$DEPO::                ;0
002011  060             L$UNIT::        .ASCII  /0/
002012                                          ;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  101412                  .WORD   L$HARD
002020                  L$SPCP::                ;POINTER TO S.W. QUES.
002020  101544                  .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  102404                  .WORD   L$LAST
002030                  L$STA::                 ;RESERVED FOR APT STATS
002030  000000                  .WORD   0
002032                  L$CO::
002032  000000                  .WORD   0
002034                  L$DTYP::                ;DIAGNOSTIC TYPE
002034  000000                  .WORD   0
002036                  L$APT::                 ;APT EXPANSION
002036  000000                  .WORD   0
002040                  L$DTP::                 ;PTR. TO DISPATCH TABLE
002040  002124                  .WORD   L$DISPATCH
002042                  L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
```

PROGRAM HEADER

```
002042  000000                  .WORD   0
002044                  L$ENVI::                ;FLAGS DESCRIBE HOW IT WAS SETUP
002044  000000                  .WORD   0
002046                  L$EXP1::                ;EXPANSION WORD
002046  000000                  .WORD   0
002050                  L$MREV::                ;SVC REV AND EDIT #
002050     003                  .BYTE   C$REVISION
002051     003                  .BYTE   C$EDIT
002052                  L$EF::                  ;DIAG. EVENT FLAGS
002052  000000                  .WORD   0
002054  000000                  .WORD   0
002056                  L$SPC::
002056  000000                  .WORD   0
002060                  L$DEVP::                ; POINTER TO DEVICE TYPE LIST
002060  003402                  .WORD   L$DVTYP
002062                  L$REPP::                ;PTR. TO REPORT CODE
002062  022712                  .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  022400                  .WORD   L$AU
002072                  L$DUT::                 ;PTR. TO DROP UNIT CODE
002072  022476                  .WORD   L$DU
002074                  L$LUN::                 ;LUN FOR EXERCISERS TO FILL
002074  000000                  .WORD   0
002076                  L$DESP::                ;POINTER TO DIAG. DESCRIPTION
002076  003410                  .WORD   L$DESC
002100                  L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
002100  104035                  EMT     E$LOAD
002102                  L$ETP::                 ;POINTER TO ERRTBL
002102  000000                  .WORD   0
002104                  L$ICP::                 ;PTR. TO INIT CODE
002104  021556                  .WORD   L$INIT
002106                  L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  022664                  .WORD   L$CLEAN
002110                  L$ACP::                 ;PTR. TO AUTO CODE
002110  022604                  .WORD   L$AUTO
002112                  L$PRT::                 ;PTR. TO PROTECT TABLE
002112  021546                  .WORD   L$PROT
002114                  L$TEST::                ;TEST NUMBER
002114  000000                  .WORD   0
002116                  L$DLY::                 ;DELAY COUNT
002116  000000                  .WORD   0
002120                  L$HIME::                ;PTR. TO HIGH MEM
002120  000000                  .WORD   0
```

DISPATCH TABLE

```
                                        .SBTTL  DISPATCH TABLE

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

                                        DISPATCH 12
002122  000014                         .WORD   12
002124                          L$DISPATCH::
002124  023474                          .WORD   T1
002126  024456                          .WORD   T2
002130  026450                          .WORD   T3
002132  032044                          .WORD   T4
002134  034634                          .WORD   T5
002136  040426                          .WORD   T6
002140  050540                          .WORD   T7
002142  052020                          .WORD   T8
002144  062646                          .WORD   T9
002146  066716                          .WORD   T10
002150  074560                          .WORD   T11
002152  077732                          .WORD   T12
```

```
                                        .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  176000                          .WORD   176000          ; 1ST (OF 2) REGISTERS.
002160  000224                          .WORD   224             ; INTERRUPT VECTOR
002162  000200                          .WORD   PRI04           ; INTERRUPT PRIORITY.
002164                                  ENDHW
002164                          L10000:
```

SOFTWARE P-TABLE

```
                                .SBTTL  SOFTWARE P-TABLE

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

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

                                ENDMOD
```

SOFTWARE P-TABLE

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

                                BGNMOD  TSV3
002176                  TSV3::
                                .SBTTL  GLOBAL EQUATES SECTION

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

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

GLOBAL EQUATES SECTION

```
                        ; PRIORITY LEVEL DEFINITIONS
                        ;
000340                  PRI07== 340
000300                  PRI06== 300
000240                  PRI05== 240
000200                  PRI04== 200
000140                  PRI03== 140
000100                  PRI02== 100
000040                  PRI01== 40
000000                  PRI00== 0
                        ;
                        ;OPERATOR FLAG BITS
                        ;
000004                  EVL==     4
000010                  LOT==    10
000020                  ADR==    20
000040                  IDU==    40
000100                  ISR==   100
000200                  UAM==   200
000400                  BOE==   400
001000                  PNT==  1000
002000                  PRI==  2000
004000                  IXE==  4000
010000                  IBE== 10000
020000                  IER== 20000
040000                  LOE== 40000
100000                  HOE== 100000

33
34 002176                       KT11                    ;DEFINE MEMORY MANAGEMENT REGISTERS
                        .SBTTL  MEMORY MANAGEMENT DEFINITIONS
                        ;*KT11 VECTOR ADDRESS
000250                  MMVEC=  250
                        ;*KT11 STATUS REGISTER ADDRESSES
177572                  SR0=    177572
177574                  SR1=    177574
177576                  SR2=    177576
172516                  SR3=    172516
                        .IF NB
                        ;*USER "I" PAGE DESCRIPTOR REGISTERS
                        UIPDR0= 177600
                        UIPDR1= 177602
                        UIPDR2= 177604
                        UIPDR3= 177606
                        UIPDR4= 177610
                        UIPDR5= 177612
                        UIPDR6= 177614
                        UIPDR7= 177616
                        .IF NB
                        ;*USER "D" PAGE DESCRIPTOR REGISTORS
                        UDPDR0= 177620
                        UDPDR1= 177622
                        UDPDR2= 177624
                        UDPDR3= 177626
                        UDPDR4= 177630
                        UDPDR5= 177632
                        UDPDR6= 177634
                        UDPDR7= 177636
```

```
.ENDC
;*USER "I" PAGE ADDRESS REGISTERS
UIPAR0= 177640
UIPAR1= 177642
UIPAR2= 177644
UIPAR3= 177646
UIPAR4= 177650
UIPAR5= 177652
UIPAR6= 177654
UIPAR7= 177656
.IF NB
;*USER "D" PAGE ADDRESS REGISTERS
UDPAR0= 177660
UDPAR1= 177662
UDPAR2= 177664
UDPAR3= 177666
UDPAR4= 177670
UDPAR5= 177672
UDPAR6= 177674
UDPAR7= 177676
.ENDC
.ENDC
.IF NB
;*SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS
SIPDR0= 172200
SIPDR1= 172202
SIPDR2= 172204
SIPDR3= 172206
SIPDR4= 172210
SIPDR5= 172212
SIPDR6= 172214
SIPDR7= 172216
.IF NB
;*SUPERVISOR "D" PAGE 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
```

MEMORY MANAGEMENT DEFINITIONS

```
                        SDPAR3= 172266
                        SDPAR4= 172270
                        SDPAR5= 172272
                        SDPAR6= 172274
                        SDPAR7= 172276
                        .ENDC
                        .ENDC
                        ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
172300                  KIPDR0= 172300
172302                  KIPDR1= 172302
172304                  KIPDR2= 172304
172306                  KIPDR3= 172306
172310                  KIPDR4= 172310
172312                  KIPDR5= 172312
172314                  KIPDR6= 172314
172316                  KIPDR7= 172316
                        .IF NB
                        ;*KERNEL "D" PAGE DESCRIPTOR REGISTERS
                        KDPDR0= 172320
                        KDPDR1= 172322
                        KDPDR2= 172324
                        KDPDR3= 172326
                        KDPDR4= 172330
                        KDPDR5= 172332
                        KDPDR6= 172334
                        KDPDR7= 172336
                        .ENDC
                        ;*KERNEL "I" PAGE ADDRESS REGISTERS
172340                  KIPAR0= 172340
172342                  KIPAR1= 172342
172344                  KIPAR2= 172344
172346                  KIPAR3= 172346
172350                  KIPAR4= 172350
172352                  KIPAR5= 172352
172354                  KIPAR6= 172354
172356                  KIPAR7= 172356
                        .IF NB
                        ;*KERNEL "D" PAGE ADDRESS REGISTERS
                        KDPAR0= 172360
                        KDPAR1= 172362
                        KDPAR2= 172364
                        KDPAR3= 172366
                        KDPAR4= 172370
                        KDPAR5= 172372
                        KDPAR6= 172374
                        KDPAR7= 172376
                        .ENDC
```

```
                        .SBTTL  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                  MIADDR= 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= BI16            ;TRANSPORT ON LINE
000040                  XSOIE=  BIT5            ;INTERRUPT ENABLE
000020                  XSOVCK= BIT4            ;VOLUME CHECK BIT
000010                  XSOPED= BIT3            ;PHASE ENCODED DRIVE
000004                  XSOWLK= BIT2            ;WRITE LOCKED
000002                  XSOBOT= BIT1            ;BEGINNING OF TAPE
000001                  XSOEOT= BIT0            ;END OF TAPE

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

100000                  X1.DLT  = BIT15         ;DATA LATE
```

TSV05 REGISTER AND PACKET DEFINITIONS

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

                        ;+
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
                        ;(XST2)
                        ;-
                        X2.OPM  = BIT15          ;OPERATION IN PROGRESS (TAPE MOVING)
                        X2.RCE  = BIT14          ;RAM CHECKSUM ERROR
                        X2.SPARE= BIT13+BIT12+BIT11+BIT9+BIT8   ;NOT USED BY TSV05 (ALWAYS=0)
                        X2.WCF  = BIT10          ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
                        X2.EXTF = BIT7           ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
                        X2.BUFE = BIT6           ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
                        X2.REV  = 000077         ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
                        X2.UNIT = BIT2+BIT1+BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.

                        ;+
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
                        ;(XST3)
                        ;-
                        X3.MDE  = 177400         ;MICRO-DIAGNOSTIC ERROR CODE
                        X3.SPARE= BIT7           ;NOT USED BY TSV05
                        X3.OPI  = BIT6           ;OPERATION INCOMPLETE
                        X3.REV  = BIT5           ;REVERSE
                        X3.TRF  = BIT4           ;TRANSPORT RESPONSE FAILURE
                        X3.DCK  = BIT3           ;DENSITY CHECK
                        X3.MBZ  =BIT2+BIT1       ;NOT USED ALWAYS 0
                        X3.RIB  = BIT0           ;REVERSE INTO BOT

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

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

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

                        ;+
                        ;
                        ;DEVICE REGISTER OFFSETS
                        ;
                        ;-
```

TSV05 REGISTER AND PACKET DEFINITIONS

```
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

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

TSV05 REGISTER AND PACKET DEFINITIONS

```
                        ;
                        ;-

000006                  XST0== 6                ;EXTENDED STATUS REGISTER 0 (WORD 4)
000010                  XST1== 8.               ;EXTENDED STATUS REGISTER 1 (WORD 5)
000012                  XST2== 10.              ;EXTENDED STATUS REGISTER 2 (WORD 6)
000014                  XST3== 12.              ;EXTENDED STATUS REGISTER 3 (WORD 7)
000016                  XST4== 14.              ;EXTENDED STATUS REGISTER 4 (WORD 8)

                        ;*
                        ;
                        ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
                        ;
                        ;-

000002                          PKLOW   = 2             ;LOW ORDER CHARACTERISTIC DATA POINTER
000004                          PKHI    = 4             ;HIGH ORDER CHARACTERISTIC DATA POINTER
000006                          PKBCNT  = 6             ;NUMBER OF BYTES IN DATA PACKET

000010                          EXBCNT=10               ;NUMBER OF BYTES IN EXTENDED DATA PACKET

                        ;*
                        ;DATA PACKET OFFSETS FOR WRITE SUBSYTEM COMMAND
                        ;-
000000                          BSEL0   = 0             ;BYTE 0
000001                          BSEL1   = 1             ;BYTE 1
000002                          SEL2    = 2             ;WORD 2
000004                          SELDATA = 4             ;WORD 3

                        ;*
                        ;BSEL0 SELECT CODES FOR WRITE SUBSYSTEM COMMAND
                        ;-
000000                  PW.NOP          = 0             ;NO-OP
000001                  PW.RDRAM        = 1             ;READ RAM
000002                  PW.WTRAM        = 2             ;WRITE RAM
000003                  PW.RFIFO        = 3             ;READ FIFO
000004                  PW.WFIFO        = 4             ;WRITE FIFO
000005                  PW.RDSTAT       = 5             ;READ STATUS
000006                  PW.WCTL         = 6             ;WRITE TAPE CONTROL
000007                  PW.WFMT         = 7             ;WRITE TAPE FORMAT
000010                  PW.WMISC        = 10            ;WRITE MISCELLANEOUS
000011                  PW.WNPR         = 11            ;WRITE NPR CONTROL
000020                  PW.D22          = 20            ;DO MICROTEST 22
000021                  PW.D11          = 21            ;DO MICROTEST 11
000022                  PW.D13          = 22            ;DO MICROTEST 13
000023                  PW.NO1311       = 23            ;DISABLE MICROTEST 11 AND 13
000024                  PW.RDEXT        = 24            ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS

                        ;*
                        ;BSEL1 CODES FOR WRITE TAPE CONTROL
                        ;-
000200                  WC.IFAD         = BIT7          ;IFAD - FORMATTER ADDRESS
000100                  WC.IOTAD        = BIT6          ;ITAD0  - TRANSPORT ADDRESS BIT 0
000040                  WC.I1TAD        = BIT5          ;ITAD1  - TRANSPORT ADDRESS BIT 1
000020                  WC.I5RESV       = BIT4          ;IRESV5 - RESERVED #5
000010                  WC.IREW         = BIT3          ;IREW   - REWIND
000004                  WC.IRWU         = BIT2          ;IRWU   - REWIND AND UNLOAD
```

TSV05 REGISTER AND PACKET DEFINITIONS

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

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

000200                          S2.DIM          = BIT7          ;WORD #9 BYTE 2 DATA IN MISS
000100                          S2.ILW          = BIT6          ;                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                  SO.IRDY         = BIT6          ;               IRDY L
000040                  SO.IONL         = BIT5          ;               IONL L
000020                  SO.ILDP         = BIT4          ;               ILDP L
000010                  SO.IDBY         = BIT3          ;               IDBY L
000004                  SO.IRWD         = BIT2          ;               IRWD L
000002                  SO.IFBY         = BIT1          ;               IFBY L
000001                  SO.IFPT         = BIT0          ;               IFPT L
                                .SBTTL  SPECIAL MACROS AND OPDEFS.

                        ;+
                        ;SAVE GENERAL REGS 1 TO 5
                        ;-

                                .MACRO  SAVREG
                                JSR     R5,REGSAV
                                .ENDM

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

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

SPECIAL MACROS AND OPDEFS.

```
                                        ERRNO=ERRNO+1
                                ;;;.IIF NDF LISTALL, .LIST
                                .LIST
                                .ENDM

                                ;+
                                ;MACRO TO PERFORM XOR
                                ;-

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

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

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

002176  000000                  FORCER::        0       ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED -
                                                ; - BY THE MACRO "IFERROR").  AN ERROR NEED NOT -
                                                ; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.
                                        .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                  BENBSW::        .WORD   0       ;BUFFER ENABLE SWITCH SW 0=OFF;1=ON
002232  000000                  EXPD::          .WORD   0       ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
002234  000000                  RECV::          .WORD   0       ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
002236  000000                  ERRHI::         .WORD   0       ;HIGH ADDRESS MEMORY ERROR
002240  000000                  ERRLO::         .WORD   0       ;LOW ADDRESS MEMORY ERROR
002242                          RAMDATA::       .BLKW   16.     ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
002302  000000                  RAMSIZ::        .WORD   0       ;RAM DATA SIZE FOR PRAMPKT ROUTINE
```

GLOBAL DATA SECTION

```
                                RCVHIADD::      .WORD   0       ;RECEIVED BUFFER HIGH ADDRESS
                                RCVLOADD::      .WORD   0       ;RECEIVED BUFFER LOW ADDRESS
                                COUNT::         .WORD   0       ;TEST COUNT PATTERN
                                DATA::          .WORD   0       ;TEST DATA
                                TSTFLAG::       .WORD   0       ;TEST FLAG WORD
                                TSTPTR::        .WORD   0       ;TSTBLK POINTER
                                PRMNO::         .WORD   0       ;PRINT ROUTINE TEMP
                                EXPMSG::        .BLKB   100.    ;EXPECTED MESSAGE BUFFER DATA
                                RECMSG::        .BLKB   100.    ;RECEIVED MESSAGE BUFFER DATA
                                TMPBFR::        .BLKB   80.     ;TEMPORARY STORAGE FOR PRINT
                                        .SBTTL  TSTBLK  - TEST DATA TABLE

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

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

J3

TSTBLK - TEST DATA TABLE

```
003046  167777                          .WORD   ↑CBIT12
003050  157777                          .WORD   ↑CBIT13
003052  137777                          .WORD   ↑CBIT14
003054  077777                          .WORD   ↑CBIT15
003056  125252                          .WORD   125252          ;ALTERNATING ONES, ZEROS
003060  052525                          .WORD   052525          ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
        003062                  TBLEND==.
                                        .SBTTL  GLOBAL ENVIRONMENT STORAGE
                                ;
                                ;STORAGE FOR DEVICE REGISTERS
                                ;
003062  000000  100000  000000  DUMMY:  0,100000,0,0      ;DUMMY DEVICE REGISTERS...
003072  000000  000000  000000          0,0,0,0,0,0,0,0
                                                        ;...FOR MULTI-UNIT CHECKOUT.

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

003116  000000                  TEMP1::         .WORD   0               ;SOME TEMP LOCATIONS.
003120  000000                  TEMP2::         .WORD   0
003122  000000                  XXCOMM::        .WORD   0               ;XXDP+ COMM BLOCK POINTER.
003124  000000                  FREE::          .WORD   0               ;1ST FREE MEMORY ADDRESS...
003126  000000                  FRESIZ::        .WORD   0               ;...AND SIZE (IN WORDS).
003130  000000                  FREEHI: .WORD   0               ;LAST WORD IN FREE SPACE
003132  000000                  KTFLG::         .WORD   0               ;KT11, MEM AVAIL FLAG -
                                                                        ;- .WORD        0 = <24K OR NO KT -
                                                                        ;- NZ = >24K AND KT.
003134  000000                  KTENABLE::      .WORD   0               ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
003136  000000                  NXMFLG::        .WORD   0               ;SET IF WE CAN TEST CLEARED OTHERWISE
003140  000000                  NXMLO::         .WORD   0               ;NXM LO ADDRESS BITS
003142  000000                  NXMHI::         .WORD   0               ;NXM HI ADDRESS BITS FOR DAL'S 16-21
003144  000000                  T23A::          .WORD   0               ;11/23A FLAG
003146  000000                  T23B::          .WORD   0               ;11/23B FLAG
003150  000000                  T3BFLG::        .WORD   0               ;TEST 3B FLAG ↑O
003152  002000                  PST32W::        .WORD   2000            ;32W BLOCK ADDRESS FOR 32K START
003154  000000                  SIFLAG::        .WORD   0               ;
003156  000000                  BADDAT::        .WORD   0               ;ACTUAL DATA
003160  000000                  GDDAT::         .WORD   0               ;EXPECTED DATA
003162  000000                  LOOPFL::        .WORD   0
003164                          CTAB::                                  ;CONFIGURATION TABLES.
003164  000000                  CTABM::         .WORD   0               ;CONFIG WORK.
003166  000000                                  .WORD   0
003170  000000                                  .WORD   0
003172  000000                                  .WORD   0
003174  177777                                  .WORD   -1              ;END OF MEM TABLE.
003176                          CTABE::
                                ;ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
                                ;
                                ;       0       =       UNIT NOT TESTED
                                ;       100000  =       UNIT ONLINE, NO ERRORS
                                ;       10XXXX  =       UNIT ONLINE, ENCOUNTERED XXXX ERRORS
                                ;       160000  =       UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
                                ;       160001  =       UNIT DROPPED, NOT IDLE AT START
                                ;       14XXXX  =       UNIT DROPPED, ENCOUNTERED XXXX ERRORS
                                ;
003176                          ERTABL:         .BLKW   64.
```

K3

GLOBAL ENVIRONMENT STORAGE

```
552 003376  000000                ERTABE:         .WORD   0
553
554 003400  000000                SKIPT:  .WORD   0               ;1=SKIP SUBTEST 0=NO SKIP OF SUBTEST
```

GLOBAL TEXT MESSAGES

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

                                ;+
                                ;NAMES OF DEVICES SUPPORTED
                                ;-

003402                                  DEVTYP  <TSV05>
003402                          L$DVTYP::
003402  124  123  126                    .ASCIZ  %TSV05%
                                        .EVEN

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


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

003502  003542  003545  003551  TSSRBIT::        .WORD   1$,2$,3$,4$,5$,6$,7$,8$
003522  003603  003607  003613                  .WORD   9$,10$,11$,12$,13$,14$,15$,16$
003542     123     103     000  1$:      .ASCIZ  'SC'
003545     102     111     105  2$:      .ASCIZ  'BIE'
003551     123     103     105  3$:      .ASCIZ  'SCE'
003555     122     115     122  4$:      .ASCIZ  'RMR'
003561     116     130     115  5$:      .ASCIZ  'NXM'
003565     116     102     101  6$:      .ASCIZ  'NBA'
003571     102     111     124  7$:      .ASCIZ  'BIT9'
003576     102     111     124  8$:      .ASCIZ  'BIT8'
003603     123     123     122  9$:      .ASCIZ  'SSR'
003607     117     106     114  10$:     .ASCIZ  'OFL'
003613     102     111     124  11$:     .ASCIZ  'BIT5'
003620     102     111     124  12$:     .ASCIZ  'BIT4'
003625     102     111     124  13$:     .ASCIZ  'BIT3'
003632     102     111     124  14$:     .ASCIZ  'BIT2'
003637     102     111     124  15$:     .ASCIZ  'BIT1'
003644     102     111     124  16$:     .ASCIZ  'BIT0'
                                .EVEN
003652     124     123     123  SFIERR:  .ASCIZ  'TSSR ERROR AFTER SOFT INIT'
003705     124     123     123  SFHERR:  .ASCIZ  'TSSR ERROR AFTER BUS RESET'
003740     040     040     116  NXR:     .ASCIZ  /  NON-EXISTANT DEVICE REGISTER/
003777     045     101     040  NXRX:    .ASCIZ  /%A  ADDRESS: %O6/
004020     045     101     040  TSSX:    .ASCII  /%A  TSBA,TSSR EXP'D: %O6%A,%O6%N/
004060     045     101     040          .ASCIZ  /%A  TSBA,TSSR REC'D: %O6%A,%O6/
004117     045     116     045  FUSI:    .ASCII  /%N%A/
004123     040     040     125  USI:     .ASCIZ  /  UNEXPECTED INTERRUPT/
004152     040     040     111  NSI:     .ASCIZ  /  INTERRUPT EXPECTED, NOT RECEIVED/
```

GLOBAL TEXT MESSAGES

```
                                                FNOINTR:        .ASCII  /%N%A/
                                                NOINTR: .ASCIZ  /  NO INTERRUPT WAS GENERATED/
                                                IFAULT: .ASCIZ  /  INTERRUPT FAULT/
                                                INTX:   .ASCIZ  /%A  CPU PC: %O6%A  TSBA: %O6/
                                                NOINIT: .ASCIZ  /  "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
                                                NSINIT: .ASCIZ  /  "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
                                                BRINIT: .ASCIZ  /  "BUS-RESET" DIDN'T INITIALIZE THE DPU/

                                                NUL:    .ASCIZ  //
                                                NULCR:  .ASCIZ  /%N/
                                                EXPGOT: .ASCIZ  /%A EXP'D: %O6%A, REC'D: %O6/
                                                EXPGT2: .ASCIZ  /%N%A EXP'D: %O6%A, %O6%N%A REC'D: %O%A, %O6/
                                                DUAD12: .ASCIZ  /%A REG(W) WRITTEN TO: %O6%A REG(R) READ; EXP'D: %O6%A, REC'D: %O6/
                                                PKTRAM::        .ASCIZ  'RAM Contents Do Not Match Packet Sent'
                                                SCME:   .ASCIZ  /  CONFIG DOESN'T MATCH MFG. MASTER/
                                                WRTMSG: .ASCIZ  'WRITE CHARACTERISTICS Failed'
                                                WRTERR: .ASCIZ  'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
                                                RDERR:  .ASCIZ  'TSSR Incorrect After READ Command, More Bits Set Than SSR'
                                                SCHERR: .ASCIZ  'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
                                                RETERR: .ASCIZ  'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
                                                NOMEM:  .ASCIZ  '%N%A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****%N'
                                                M8186:  .ASCIZ  '%N%A ****************** 11/23A SYSTEM ******************%N'
                                                M8189:  .ASCIZ  '%N%A ****************** 11/23B SYSTEM ******************%N'
                                                        .EVEN
                                                        .SBTTL  GLOBAL ERROR REPORT SECTION

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

                                                BGNMSG  NXRERR          ;NON-EXISTANT DEVICE REGISTER.
                                        NXRERR::
005736                                          PRINTX  #NXRX,NODEV     ;NODEV = NEXM ADDRESS.
005736  013746  003114                          MOV     NODEV,-(SP)
005742  012746  003777                          MOV     #NXRX,-(SP)
005746  012746  000002                          MOV     #2,-(SP)
005752  010600                                  MOV     SP,R0
005754  104415                                  TRAP    C$PNTX
005756  062706  000006                          ADD     #6,SP
005762  004737  005770                          JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
005766                                          ENDMSG
005766                                  L10002:
005766  104423                                  TRAP    C$MSG

                                        ;
                                        ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
                                        ; TO ANY OF THE ABOVE ERROR SIGNATURES.
                                        ;
005770  005727                          EXTEND: TST     (PC)+
005772  000000                          EXTA:   0                       ; 0 = NO EXTENSION.
005774  001402                                  BEQ     1$
005776  004777  177770                          JSR     PC,@EXTA        ; APPEND EXTENSION TEXT.
006002                                  1$:     PRINTX  #NULCR          ; PRINT A BLANK LINE
006002  012746  004530                          MOV     #NULCR,-(SP)
006006  012746  000001                          MOV     #1,-(SP)
```

GLOBAL ERROR REPORT SECTION

```
        006012  010600                          MOV     SP,R0
        006014  104415                          TRAP    C$PNTX
        006016  062706  000004                  ADD     #4,SP
    673 006022  000207                          RTS     PC
```

```
                                        .SBTTL  PRITSSR - PRINT TSSR CONTENTS

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

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

PRITSSR - PRINT TSSR CONTENTS

```
                006170  012746  000002                  MOV     #2,-(SP)
                006174  010600                          MOV     SP,R0
                006176  104415                          TRAP    C$PNTX
                006200  062706  000006                  ADD     #6,SP
            719
            720 006204  010403              20$:        MOV     R4,R3           ;GET THE TSSR CONTENTS
            721 006206  042703  177761                  BIC     #^CTERCLS,R3    ;CLEAR ALL BUT TERMINATION
            722 006212  016303  006676                  MOV     TCOCOD(R3),R3   ;GET THE TERMINATION CODE MEANING
            723 006216                                  PRINTX  #TCOASC,R3      ;PRINT THE TERMINATION CODE
                006216  010346                          MOV     R3,-(SP)
                006220  012746  006476                  MOV     #TCOASC,-(SP)
                006224  012746  000002                  MOV     #2,-(SP)
                006230  010600                          MOV     SP,R0
                006232  104415                          TRAP    C$PNTX
                006234  062706  000006                  ADD     #6,SP
            724 006240  010403                          MOV     R4,R3           ;TSSR CONTENTS AGAIN
            725 006242  042703  177717                  BIC     #^CFATERR,R3    ;CLEAR ALL BUT FATAL TERMINATION
            726 006246  001416                          BEQ     25$             ;DON'T PRINT IF ZERO
            727 006250  006203                          ASR     R3
            728 006252  006203                          ASR     R3
            729 006254  006203                          ASR     R3              ;ALINE TERMINATION CODE FOR INDEX
            730 006256  016303  007236                  MOV     TSFCOD(R3),R3   ;GET THE FATAL TERMINATION CODE
            731 006262                                  PRINTX  #TFCASC,R3      ;PRINT THE FATAL TERMINATION CODE
                006262  010346                          MOV     R3,-(SP)
                006264  012746  006537                  MOV     #TFCASC,-(SP)
                006270  012746  000002                  MOV     #2,-(SP)
                006274  010600                          MOV     SP,R0
                006276  104415                          TRAP    C$PNTX
                006300  062706  000006                  ADD     #6,SP
            732 006304  042704  176377              25$:BIC     #^CHIADDR,R4    ;CLEAR ALL BUT EXTENDED ADDRESS
            733 006310  001411                          BEQ     30$             ;DON'T PRINT IF ZERO
            734 006312                                  PRINTX  #TEXASC,R4      ;PRINT THE EXTENDED ADDRESS BITS
                006312  010446                          MOV     R4,-(SP)
                006314  012746  006435                  MOV     #TEXASC,-(SP)
                006320  012746  000002                  MOV     #2,-(SP)
                006324  010600                          MOV     SP,R0
                006326  104415                          TRAP    C$PNTX
                006330  062706  000006                  ADD     #6,SP
            735 006334  013703  002200              30$:MOV     EPRTSW,R3               ;PRINT MEAASGE BUFFER ADDRESS
            736 006340                                  PRINTX  R3                      ;PRINT PROPER MESSAGE
                006340  010346                          MOV     R3,-(SP)
                006342  012746  000001                  MOV     #1,-(SP)
                006346  010600                          MOV     SP,R0
                006350  104415                          TRAP    C$PNTX
                006352  062706  000004                  ADD     #4,SP
            737 006356  000207                          RTS     PC              ;RETURN TO CALLER
            738
            744 006360                          EPRT2:
            745 006360      045     116     045 EPRT1:  .ASCIZ   '%N%A *****REPLACE M7196*****'
            746
            756 006415      045     116     045 TSSRFOR:        .ASCIZ  '%N%A TSSR = %O6'
            757 006435      045     116     045 TEXASC:         .ASCIZ  '%N%A Extended Address Bits = %O6'
            758 006476      045     116     045 TCOASC:         .ASCIZ  '%N%A Termination Class Code = %T'
            759 006537      045     116     045 TFCASC:         .ASCIZ  '%N%A Fatal Termination Class Code = %T'
            760 006606      045     116     045 TSSDEF:         .ASCIZ  '%N%A TSSR Bits Set: %T'
            761 006635      045     116     045 AMBTSSR:        .ASCIZ  '%N%A TSSR Contents Are Ambiguous'
            762                                             .EVEN
```

```
     763 006676  006716  006741  006767  TCOCOD: .WORD   1$,2$,3$,4$,5$,6$,7$,8$
     764 006716    116     157     162  1$:     .ASCIZ  'Normal Termi ation'
     765 006741    124     145     162  2$:     .ASCIZ  'Termination Condition'
     766 006767    124     141     160  3$:     .ASCIZ  'Tape Status Alert'
     767 007011    106     165     156  4$:     .ASCIZ  'Function Reject'
     768 007031    122     145     143  5$:     .ASCIZ  'Recoverable Error - Tape Position One Record Down'
     769 007113    122     145     143  6$:     .ASCIZ  'Recoverable Error - Tape Was Not Moved'
     770 007162    125     156     162  7$:     .ASCIZ  'Unrecoverable Error'
     771 007206    106     141     164  8$:     .ASCIZ  'Fatal Controller Error'
     772                                          .EVEN
     773
     774 007236  007246  007302  007313  TSFCOD: .WORD   1$,2$,3$,4$
     775 007246    111     156     164  1$:     .ASCIZ  'Internal Diagnostic Failure'
     776 007302    122     145     163  2$:     .ASCIZ  'Reserved'
     777 007313    102     165     163  3$:     .ASCIZ  'Bus Interface or Sanity Check Error'
     778 007357    122     145     163  4$:     .ASCIZ  'Reserved'
     779                                          .EVEN
     780                                          .SBTTL  PRIPKT  - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET
     781
     782                                  ;+
     783                                  ;THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
     784                                  ;THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
     785                                  ;
     786                                  ;INPUT:
     787                                  ;
     788                                  ;       R0      NUMBER OF WORDS IN PACKET
     789                                  ;       R3      HIGH ORDER COMMAND PACKET ADDRESS
     790                                  ;       R4      ADDRESS OF COMMAND PACKET
     791                                  ;
     792                                  ;       NOTE:   R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
     793                                  ;-
     794
     795 007370                           PRIPKT::
     796 007370                                   SAVREG                  ;SAVE THE REGISTERS
     797 007374  010005                           MOV     R0,R5           ;SAVE NO. OF WORDS IN PACKET
     798 007376  005737  003134                   TST     KTENABLE        ;ABOVE 28K UNDER TEST?
     799 007402  001001                           BNE     10$             ;BR IF YES
     800 007404  005003                           CLR     R3              ;SET HIGH ORDER ADDRESS TO 0
     801 007406  010301                   10$:    MOV     R3,R1           ;COPY HIGH ORDER ADDRESS
     802 007410  010400                           MOV     R4,R0           ;GET LOWER ADDRESS
     803 007412  006100                           ROL     R0              ;SHIFT BIT 15 INTO C BIT
     804 007414  006101                           ROL     R1              ;AND INTO HIGH ORDER.
     805 007416                                   PRINTB  #PKTADD,R1,R4   ;PRINT PACKET ADDRESS
         007416  010446                           MOV     R4,-(SP)
         007420  010146                           MOV     R1,-(SP)
         007422  012746  007554                   MOV     #PKTADD,-(SP)
         007426  012746  000003                   MOV     #3,-(SP)
         007432  010600                           MOV     SP,R0
         007434  104414                           TRAP    C$PNTB
         007436  062706  000010                   ADD     #10,SP
     806 007442  010300                   15$:    MOV     R3,R0           ;GET HIGH ORDER ADDRESS
     807 007444  001404                           BEQ     20$             ;BR IF NOT ABOVE 28K.
     808 007446  010401                           MOV     R4,R1           ;GET LOW ORDER ADDRESS
     809 007450  004737  017316                   JSR     PC,SETMAP       ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
     810 007454  010004                           MOV     R0,R4           ;GET RETURNED PAR6 ADDRESS BIAS
     811 007456  005001                   20$:    CLR     R1              ;SAVE WORD NUMBER
     812 007460  012402                   25$:    MOV     (R4)+,R2        ;GET PACKET CONTENTS
```

PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

```
    813 007462                                  PRINTB  @PKTFRM,R1,R2   ;PRINT THE DATA
        007462  010246                          MOV     R2,-(SP)
        007464  010146                          MOV     R1,-(SP)
        007466  012746  007516                  MOV     @PKTFRM,-(SP)
        007472  012746  000003                  MOV     @3,-(SP)
        007476  010600                          MOV     SP,R0
        007500  104414                          TRAP    C$PNTB
        007502  062706  000010                  ADD     @10,SP
    814 007506  005201                          INC     R1              ;NEXT WORD NUMBER
    815 007510  020105                          CMP     R1,R5           ;DONE ALL PACKET WORDS?
    816 007512  002762                          BLT     25$             ;LOOP TILL ALL DONE
    817 007514  000207                          RTS     PC              ;RETURN
    818
    819 007516     045     116     045  PKTFRM: .ASCIZ   '%N%A Packet Word @%D1%A = %O6'
    820 007554     045     116     045  PKTADD: .ASCIZ   '%N%A Packet Address = %O1%O5'
    821                                         .EVEN
    822                                         .SBTTL  PRIBXOR - PRINT EXPD, RECV AND XOR BYTE
    823
    824                                 ;*
    825                                 ;
    826                                 ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
    827                                 ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
    828                                 ;
    829                                 ;INPUTS:
    830                                 ;
    831                                 ;       R1      RECEIVED DATA
    832                                 ;       R2      EXPECTED DATA
    833                                 ;
    834                                 ;OUTPUT:
    835                                 ;
    836                                 ;       R0      XOR OF EXPECTED/RECEIVED DATA
    837                                 ;
    838                                 ;-
    839
    840 007612                          PRIBXOR::
    841 007612                                  SAVREG                  ;SAVE THE REGISTERS
    842 007616  010203                          MOV     R2,R3           ;EXPECTED DATA
    843 007620                                  XOR     R1,R3           ;FORM THE EXCLUSIVE OR
    844 007630  012700  177400                  MOV     @^C<377>,R0     ;BYTE MASK
    845 007634  040001                          BIC     R0,R1           ;SAVE LOW BYTE RECV
    846 007636  040002                          BIC     R0,R2           ;SAVE LOW BYTE EXPD
    847 007640  040003                          BIC     R0,R3           ;SAVE LOW BYTE XOR
    848 007642                                  PRINTB  @XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
        007642  010346                          MOV     R3,-(SP)
        007644  010146                          MOV     R1,-(SP)
        007646  010246                          MOV     R2,-(SP)
        007650  012746  007674                  MOV     @XORBFOR,-(SP)
        007654  012746  000004                  MOV     @4,-(SP)
        007660  010600                          MOV     SP,R0
        007662  104414                          TRAP    C$PNTB
        007664  062706  000012                  ADD     @12,SP
    849 007670  010300                          MOV     R3,R0           ;R0 HAS XOR ON RETURN
    850 007672  000207                          RTS     PC              ;RETURN TO CALLER
    851
    852 007674     045     116     045  XORBFOR:        .ASCIZ   '%N%A EXPD: %O3%A RECV: %O3%A XOR: %O3'
    853                                         .EVEN
    854                                         .SBTTL  PRIXOR  - PRINT EXPD, RECV AND XOR
```

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

007742                          PRIXOR::
007742                                  SAVREG                          ;SAVE THE REGISTERS
007746  010203                          MOV     R2,R3                   ;EXPECTED DATA
007750                                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007760                                  PRINTB  #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
007760  010346                          MOV     R3,-(SP)
007762  010146                          MOV     R1,-(SP)
007764  010246                          MOV     R2,-(SP)
007766  012746  010012                  MOV     #XORFOR,-(SP)
007772  012746  000004                  MOV     #4,-(SP)
007776  010600                          MOV     SP,R0
010000  104414                          TRAP    C$PNTB
010002  062706  000012                  ADD     #12,SP
010006  010300                          MOV     R3,R0                   ;R0 HAS XOR ON RETURN
010010  000207                          RTS     PC                      ;RETURN TO CALLER

010012    045   116   045   XORFOR: .ASCIZ  '<N><A EXPD: <06><A RECV: <06><A XOR: <06'
                                .EVEN
                                .SBTTL  PRIEQU  - PRINT BIT NUMBERS AS ASCII EQUIVALENT

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

010060                          PRIEQU:
010060                                  SAVREG                                  ;SAVE THE REGISTERS
010064  000207                          RTS     PC                              ;RETURN TO CALLER

                                .SBTTL  PRIRAM  - PRINT RAM ADDRESS
                                ;*
                                ;
                                ;PRINT CONTROLLER RAM ADDRESS.
```

PRIRAM - PRINT RAM ADDRESS

```
                                        ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R4      RAM ADDRESS
                                        ;
                                        ;-
                                        PRIRAM:
                                                SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
                                                PRINTB  @RAMFOR,R4              ;PRINT RAM ADDRESS IN ERROR
010072  010446                                  MOV     R4,-(SP)
010074  012746  010116                          MOV     @RAMFOR,-(SP)
010100  012746  000002                          MOV     #2,-(SP)
010104  010600                                  MOV     SP,R0
010106  104414                                  TRAP    C$PNTB
010110  062706  000006                          ADD     #6,SP
010114  000207                                  RTS     PC                      ;RETURN

010116      045     116     045 RAMFOR: .ASCIZ  '%N%A CONTROLLER RAM ADDRESS = %06'
                                        .EVEN

                                                .SBTTL  PRIADD  - PRINT MEMORY ERROR ADDRESS
                                        ;*
                                        ;
                                        ;PRINT MEMORY ADDRESS
                                        ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                                        ;
                                        ; IMPLICIT INPUTS
                                        ;
                                        ;       ERRHI   - HIGH ORDER ADDRESS
                                        ;       ERRLO   - LOW ORDER ADDRESS
                                        ;
                                        ;-
                                        PRIADD:
                                                SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
                                                MOV     ERRHI,R0                ;GET HIGH ADDRESSS
                                                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
                                                PRINTB  @PRIA0,R0,R2            ;PRINT MEMORY ADDRESS IN ERROR
010202  010246                                  MOV     R2,-(SP)
010204  010046                                  MOV     R0,-(SP)
010206  012746  010230                          MOV     @PRIA0,-(SP)
010212  012746  000003                          MOV     #3,-(SP)
010216  010600                                  MOV     SP,R0
010220  104414                                  TRAP    C$PNTB
010222  062706  000010                          ADD     #10,SP
010226  000207                                  RTS     PC                      ;RETURN

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

                                                .SBTTL  PRITADD - PRINT MEMORY TEST ADDRESS
                                        ;*
                                        ;
                                        ;PRINT MEMORY ADDRESS
```

```
                        ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                        ;
                        ; IMPLICIT INPUTS
                        ;
                        ;       ERRHI   - HIGH ORDER ADDRESS
                        ;       ERRLO   - LOW ORDER ADDRESS
                        ;
                        ;-
                        PRITADD:
                                SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
010274  013702  002236          MOV     ERRHI,R2                ;GET HIGH ADDRESSS
010304  013701  002240          MOV     ERRLO,R1                ;GET LOW ADDRESS
                                ;MOV    R1,R2                   ;COPY LOW ADDRESS
                                ;ROL    R1                      ;SHIFT BIT 15 TO C BIT
                                ;ROL    R0                      ;SHIFT INTO HIGH ORDER
010310                          PRINTB  #PRIT0,R1               ;PRINT MEMORY ADDRESS LOW IN ERROR
010310  010146                  MOV     R1,-(SP)
010312  012746  010356          MOV     #PRIT0,-(SP)
010316  012746  000002          MOV     #2,-(SP)
010322  010600                  MOV     SP,R0
010324  104414                  TRAP    C$PNTB
010326  062706  000006          ADD     #6,SP
010332                          PRINTB  #PRIT1,R2               ;PRINT MEMORY ADDRESS HIGH IN ERROR
010332  010246                  MOV     R2,-(SP)
010334  012746  010421          MOV     #PRIT1,-(SP)
010340  012746  000002          MOV     #2,-(SP)
010344  010600                  MOV     SP,R0
010346  104414                  TRAP    C$PNTB
010350  062706  000006          ADD     #6,SP
010354  000207                  RTS     PC                      ;RETURN

010356  045  116  045  PRIT0:  .ASCIZ  '%N%A MEMORY TEST ADDRESS LOW = %06'
010421  045  116  045  PRIT1:  .ASCIZ  '%N%A MEMORY TEST ADDRESS HIGH = %06'
                                .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:
                                        ;
                                        ;
                                        ;-
                                        SPACE::
010466                                          SAVREG                          ;SAVE THE GENERAL REGISTERS
010466  012737  000764  010660                   MOV     #500.,SDELAY            ;SET UP DELAY
010472  012737  140010  010650                   MOV     #140010,80$             ;SET UP COMMAND, SPACE FORWARD
010500  005703                                  TST     R3                      ;CHECK FOR DIRECTION
010506  100403                                  BMI     5$                      ;BR, IF REVERSE INDICATED
010510  010337  010652                          MOV     R3,90$                  ;LOAD UP NUMBER OF RECORDS TO SPACE
010512  000407                                  BR      10$                     ;GO DO COMMAND
010516  042703  100000                  5$:     BIC     #BIT15,R3               ;CLEAR DIRECTION BIT
010520  010337  010652                          MOV     R3,90$                  ;LOAD UP NUMBER OF RECORDS TO SPACE
010524  052737  000400  010650                  BIS     #BIT8,80$               ;SET REVERSE BIT IN COMMAND PACKET
010530  012704  010650                  10$:    MOV     #80$,R4                 ;SET UP R4 WITH PACKET ADDRESS
010536  010465  000000                          MOV     R4,TSDB(R5)             ;SEND OUT COMMAND
010542  004737  016250                  15$:    JSR     PC,WAITF                ;WAIT FOR SSR
010546  103420                                  BCS     20$                     ;BR, IF SSR IS SET AND OK
010550                                          DELAY   250             ;DELAY ABOUT .25 SECONDS
010554  012727  000250                          MOV     #250,(PC)+
010560  000000                                  .WORD   0
010562  013727  002116                          MOV     L$DLY,(PC)+
010566  000000                                  .WORD   0
010570  005367  177772                          DEC     -6(PC)
010574  001375                                  BNE     .-4
010576  005367  177756                          DEC     -22(PC)
010602  001367                                  BNE     .-20
010604  005337  010660                          DEC     SDELAY                  ;BUMP DELAY COUNTER DOWN
010610  001356                                  BNE     15$                     ;BR, IF MORE DELAY
010612  000411                                  BR      60$                     ;BR IF TROUBLE CARRY = CLEAR
010614  016501  000002                  20$:    MOV     TSSR(R5),R1             ;READ TSSR
010620  012702  000200                          MOV     #SSR,R2                 ;SET UP EXPECTED
010624  020201                          25$:    CMP     R2,R1                   ;ARE THEY OK
010626  001401                                  BEQ     40$                     ;BR, IF EQUAL = OK
010630  000402                                  BR      60$                     ;TROUBLE EXIT
010632  000261                  40$:    SEC                                     ;SET CARRY NO TROUBLE
010634  000401                                  BR      70$                     ;EXIT
010636  000241                  60$:    CLC                                     ;CARRY CLEAR = ERROR
010640                          70$:
010640  010400                                  MOV     R4,R0                   ;PASS PACKET ADDRESS
010642  000207                                  RTS     PC                      ;RETURN
```

```
                                        ;
                                        ;
                                        ;
                                        ;PACKET FOR SPACE COMMAND
                                        ;
010650                                          .=<.+10>&177770
                                        ;
                                        ;COMMAND WORD
010650  000000                          80$:    .WORD
                                        ;NUMBER OF RECORDS TO BE SPACED OVER WORD
010652  000000                          90$:    .WORD
010654  000000                                  .WORD
010656  000000                                  .WORD
010660  000000                          SDELAY: .WORD   0                       ;DELAY COUNTER
                                                .EVEN
                                                .SBTTL  WRTCHR  - WRITE CHARACTERISTICS COMMAND

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

010662                                  WRTCHR::
010662                                          SAVREG                          ;SAVE THE GENERAL REGISTERS
005037  002230                                  CLR     BENBSW          ;CLEAR BUFFER ENABLE SWITCH
005037  002226                                  CLR     EXTFEA                  ;CLEAR EXTENDED FEATURES SW SWITCH
010465  000000                          10$:    MOV     R4,TSDB(R5)             ;SEND OUT COMMAND
004737  016336                                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
103401                                          BCS     20$                     ;BR, IF SSR IS SET AND OK
000435                                          BR      60$                     ;BR IF TROUBLE CARRY = CLEAR
016501  000002                          20$:    MOV     TSSR(R5),R1             ;READ TSSR
012702  000200                                  MOV     #SSR,R2                 ;SET UP EXPECTED
```

WRTCHR - WRITE CHARACTERISTICS COMMAND

```
                                        BIT     #OFL,R1                 ;WAS OFF LINE SET IN TSSR
                                        BEQ     25$                     ;BR, IF NO OFL SET
                                        BIS     #OFL,R2                 ;MAKE THEM LOOK ALIKE
                                25$:    CMP     R2,R1                   ;ARE THEY OK
                                        BEQ     40$                     ;BR, IF EQUAL = OK
                                        BR      60$                     ;TROUBLE EXIT
                                40$:    ADD     #8.,R4                  ;POINT TO WRT CHARA DATA PACKET
                                        MOV     (R4),R3                 ;GET ADDRESS OF MESSAGE BUFFER
                                        BIT     #X2.EXTF,XST2(R3)       ;EXTENDED FEATURES BIT SET?
                                        BEQ     45$                     ;BR IF NO
                                        INC     EXTFEA                  ;SET EXTENDED FEATURES SW SWITCH
                                45$:
                                        BIT     #X2.BUFE,XST2(R3)                       ;BUFFER ENABLE SWITCH SET
                                        BEQ     50$                     ;BR, IF SWITCH NOT SET
                                        INC     BENBSW                  ;SET SOFTWARE SWITCH FOR ENABLED
                                50$:
                                        SEC                             ;SET CARRY NO TROUBLE
                                        BR      70$                     ;EXIT
                                60$:    CLC                             ;CARRY CLEAR = ERROR
                                70$:    MOV     TSSR(R5),R0             ;RETURN TSSR CONTENTS
                                        RTS     PC                      ;RETURN
                                        .SBTTL  REWIND  - POSITION TAPE (REWIND) COMMAND

                                ;+
                                ;
                                ;THIS ROUTINE WILL REWIND THE SELECTED TAPE.
                                ;
                                ;       CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
                                ;                TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
                                ;                SSR TO SET IN THE TSSR
                                ;
                                ;
                                ;CALLING SEQUENCE:
                                ;
                                ;       DO A SOFT INIT
                                ;       DO A WRITE CHARACTERISTICS
                                ;       JSR     PC,REWIND
                                ;
                                ;INPUT:
                                ;
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;
                                ;
                                ;OUTPUT
                                ;
                                ;       R0      THE CONTENTS OF R4 IS PASSED TO R0
                                ;
                                ;
                                ;
011014                          REWIND::
011014                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011020  012704  011110                  MOV     #RWPACK,R4              ;GET PACKET ADDRESS
011024  010465  000000                  MOV     R4,TSDB(R5)             ;SEND PACKET ADDRESS TO EXECUTE
011030  012703  000550                  MOV     #360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
011034  004737  016250          10$:    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
011040  103417                          BCS     20$                     ;LEAVE WHEN SSR IS SET
011042                                  DELAY   250.                    ;WAIT FOR .25 SECONDS
```

REWIND - POSITION TAPE (REWIND) COMMAND

```
        011042  012727  000372                  MOV     #250.,(PC)+
        011046  000000                          .WORD   0
        011050  013727  002116                  MOV     L$DLY,(PC)+
        011054  000000                          .WORD   0
        011056  005367  177772                  DEC     -6(PC)
        011062  001375                          BNE     .-4
        011064  005367  177756                  DEC     -22(PC)
        011070  001367                          BNE     .-20
   1152 011072  005303                          DEC     R3                      ;BUMP COUNTER DOWN
   1153 011074  001357                          BNE     10$                     ;KEEP GOING
   1154 011076  000241                          CLC                             ;CLEAR CARRY TO SET ERROR
   1155 011100  010400                  20$:    MOV     R4,R0                   ;PASS THE PACKET ADDRESS
   1156 011102  000207                          RTS     PC                      ;RETURN
   1157
   1159         011110                          .=<.+10>&177770
   1161 011110                          RWPACK:
   1162 011110  102010                          .WORD   102010                  ;POSTION COMMAND (REWIND)
   1163 011112  000000                          .WORD   0                       ;NOT USED
   1164                                         .SBTTL  CKRAM   - COMPARE RAM TO I/O PACKET
   1165
   1166                                 ;+
   1167                                 ;
   1168                                 ;ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
   1169                                 ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
   1170                                 ;
   1171                                 ;INPUT:
   1172                                 ;
   1173                                 ;       R4      ADDRESS OF THE COMMAND PACKET
   1174                                 ;       R5      FIRST DEVICE UNIBUS ADDRESS
   1175                                 ;
   1176                                 ;OUTPUT:
   1177                                 ;
   1178                                 ;       CARRY   SET - RAM MATCHES PACKET
   1179                                 ;               CLR - RAM DOES NOT MATCH PACKET
   1180                                 ;
   1181                                 ;IMPLICIT OUTPUT:
   1182                                 ;
   1183                                 ;       THE TABLE RAMDATA IS FILLED WITH THE
   1184                                 ;       DATA HELD IN RAM.
   1185                                 ;       RAMSIZ IS SET TO 8. FOR PRAMPKT ROUTINE
   1186                                 ;
   1187                                 ;SIDE EFFECTS:
   1188                                 ;
   1189                                 ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
   1190                                 ;
   1191                                 ;-
   1192
   1193 011114                          CKRAM::
   1194 011114                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
   1195 011120  012701  002242                  MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
   1196 011124  012702  000201                  MOV     #RMPKTBEG,R2            ;BYTE ADDRESS OF FIRST RAM DATA
   1197 011130  005003                          CLR     R3                      ;CLEAR THE ERROR FLAG
   1198 011132  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
   1199 011136  112765  000000  000000          MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
   1200 011144  004737  016336          10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
   1201 011150  010265  000000                  MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
   1202 011154  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
```

CKRAM - COMPARE RAM TO I/O PACKET

```
                                        MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
                                        CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
                                        BEQ     20$                     ;BRANCH IF OK
                                        INC     R3                      ;SET ERROR FLAG
                                20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
                                        CMP     R2,#RMPKTEND            ;REACHED END YET ?
                                        BLE     10$                     ;BRANCH TILL ALL READ
                                        TST     R3                      ;WAS AN ERROR FOUND ?
                                        BEQ     30$                     ;BRANCH IF NOT
                                        CLC                             ;CLEAR CARRY TO SHOW ERROR
                                        BR      50$                     ;AND EXIT
                                30$:    SEC                             ;SHOW GOOD COMPARE
                                50$:    MOV     #8.,RAMSIZ              ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
                                        RTS     PC                      ;RETURN
                                        .SBTTL  CKRAM2  - COMPARE RAM TO I/O CHARACTERISTICS DATA
                                ;+
                                ;
                                ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
                                ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
                                ;
                                ;INPUT:
                                ;
                                ;       R4      ADDRESS OF THE CHARACTERISTICS DATA
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;
                                ;OUTPUT:
                                ;
                                ;       CARRY   SET - RAM MATCHES PACKET
                                ;               CLR - RAM DOES NOT MATCH PACKET
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       THE TABLE RAMDATA IS FILLED WITH THE
                                ;       DATA HELD IN RAM.
                                ;       RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
                                ;
                                ;-

011224                          CKRAM2::
011224                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
011230  012701  002242                  MOV     @RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
011234  012702  000167                  MOV     @RMCHBEG,R2             ;BYTE ADDRESS OF FIRST RAM DATA
011240  005003                          CLR     R3                      ;CLEAR THE ERROR FLAG
011242  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
011246  112765  000000  000000          MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
011254  004737  016336          10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011260  010265  000000                  MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
011264  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011270  116511  000000                  MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
011274  122124                          CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
011276  001401                          BEQ     20$                     ;BRANCH IF OK
011300  005203                          INC     R3                      ;SET ERROR FLAG
011302  005202                  20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
```

```
1260 011304  012737  000010  002302          MOV     #8.,RAMSIZ              ;ASSUME EXTFEA NOT SET
1261 011312  005737  002226                  TST     EXTFEA                  ;IS THE SOFTWARE EXTENDED FEATURES SET
1262 011316  001407                          BEQ     25$                     ;BR, IF NOT SET
1263 011320  012737  000012  002302          MOV     #10.,RAMSIZ             ;SET RAMSIZ FOR EXTEND FEATURES
1264 011326  020227  000200                  CMP     R2,#RMCHEND             ;AT END OF EXTENDED BUFFER
1265 011332  003750                          BLE     10$                     ;BR, IF NOT AT END YET
1266 011334  000403                          BR      27$                     ;AT END BRANCH
1267 011336  020227  000176          25$:    CMP     R2,#RMCHEND-2           ;REACHED END YET ?
1268 011342  003744                          BLE     10$                     ;BRANCH TILL ALL READ
1269 011344  005703                  27$:    TST     R3                      ;WAS AN ERROR FOUND ?
1270 011346  001402                          BEQ     30$                     ;BRANCH IF NOT
1271 011350  000241                          CLC                             ;CLEAR CARRY TO SHOW ERROR
1272 011352  000401                          BR      50$                     ;AND EXIT
1273 011354  000261                  30$:    SEC                             ;SHOW GOOD COMPARE
1274 011356  000207                  50$:    RTS     PC                      ;RETURN
1275                                         .SBTTL  CKMSG   - COMPARE WRITE CHAR. MESSAGE BUFFERS
1276                                 ;+
1277                                 ;
1278                                 ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
1279                                 ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1280                                 ;ERROR PRINT ROUTINES.
1281                                 ;
1282                                 ;INPUT:
1283                                 ;
1284                                 ;       R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1285                                 ;       R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
1286                                 ;       R2      EXPD MESSAGE BUFFER ADDRESS
1287                                 ;OUTPUT:
1288                                 ;
1289                                 ;       CARRY   SET - MESSAGE BUFFERS MATCH
1290                                 ;               CLR -MESSAGE BUFFERS DON'T MATCH
1291                                 ;
1292                                 ;IMPLICIT OUTPUT:
1293                                 ;
1294                                 ;       EXPMSG          BUFFER IS SET TO EXPD DATA
1295                                 ;       RECMSG          BUFFER IS SET TO RECV DATA
1296                                 ;       RCVHIADD        SET TO HIGH ORDER ADDRESS OF RECV
1297                                 ;       RCVLOADD        SET TO LOW ORDER ADDRESS OF RECV
1298                                 ;
1299                                 ;-
1300 011360                         CKMSG::
1301 011360                                 SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
1302 011364  010037  002304                 MOV     R0,RCVHIADD             ;SAVE RECV HIGH ADDRESS
1303 011370  010137  002306                 MOV     R1,RCVLOAD              ;SAVE RECV LOW ADDRESS
1304 011374  005737  003134                 TST     KTENABLE                ;TESTING ABOVE 28K?
1305 011400  001403                         BEQ     10$                     ;BR IF NO
1306 011402  004737  017316                 JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
1307 011406  010001                         MOV     R0,R1                   ;GET RETURNED ADDRESS BIASED TO PAR6
1308 011410  005004                 10$:    CLR     R4                      ;WORD IN BUFFER
1309 011412  005003                         CLR     R3                      ;CLEAR ERROR SEEN FLAG
1310 011414  010205                         MOV     R2,R5                   ;GET EXPD BUFFER ADDRESS
1311 011416  011264  002322         15$:    MOV     (R2),EXPMSG(R4)         ;SAVE EXPD FOR ERROR REPORT
1312 011422  011164  002466                 MOV     (R1),RECMSG(R4)         ;SAVE RECV FOR ERROR REPORT
1313 011426  022221                         CMP     (R2)+,(R1)+             ;EXPD EQUAL RECV?
1314 011430  001401                         BEQ     25$                     ;BR IF YES
1315 011432  005203                         INC     R3                      ;SET ERROR SEEN FLAG
1316 011434  062704  000002         25$:    ADD     #2,R4                   ;POINT TO NEXT WORD ADDRESS
```

```
1317 011440  020427  000014                    CMP     R4,#14          ;DONE FIRST 7 WORDS?
1318 011444  003764                            BLE     15$             ;BR IF NO
1319 011446  032765  000200  000012            BIT     #X2.EXTF,XST2(R5);IS EXTENDED FEATURES SET IN EXPD?
1320 011454  001403                            BEQ     50$             ;BR IF NO
1321 011456  020427  000016                    CMP     R4,#16          ;DONE EXTENDED FEATURES WORD?
1322 011462  003755                            BLE     15$             ;BR IF NO
1323 011464  005703                 50$:       TST     R3              ;ANY ERRORS SEEN?
1324 011466  001402                            BEQ     55$             ;BR IF NO
1325 011470  000241                            CLC                     ;SET FAILURE
1326 011472  000401                            BR      60$             ;
1327 011474  000261                 55$:       SEC                     ;SET SUCCESS
1328 011476  000207                 60$:       RTS     PC              ;RETURN
1329                                            .SBTTL  CKMSG2  - COMPARE EXPD RECV MESSAGE BUFFERS
1330                                ;+
1331                                ;
1332                                ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
1333                                ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1334                                ;ERROR PRINT ROUTINES.
1335                                ;
1336                                ;INPUT:
1337                                ;
1338                                ;       R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1339                                ;       R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
1340                                ;       R2      EXPD MESSAGE BUFFER ADDRESS
1341                                ;       R3      NUMBER OF BYTES TO COMPARE
1342                                ;
1343                                ;OUTPUT:
1344                                ;
1345                                ;       CARRY   SET - MESSAGE BUFFERS MATCH
1346                                ;               CLR - MESSAGE BUFFERS DON'T MATCH
1347                                ;
1348                                ;IMPLICIT OUTPUT:
1349                                ;
1350                                ;       EXPMSG          BUFFER IS SET TO EXPD DATA
1351                                ;       RECMSG          BUFFER IS SET TO RECV DATA
1352                                ;       RCVHIADD        SET TO HIGH ORDER ADDRESS OF RECV
1353                                ;       RCVLOADD        SET TO LOW ORDER ADDRESS OF RECV
1354                                ;
1355                                ;-
1356 011500                        CKMSG2::
1357 011500                                SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
1358 011504  020327  000144                CMP     R3,#RECMSG-EXPMSG;@@D IS COUNT ABOVE MAX ALLOWED?
1359 011510  003412                        BLE     5$                      ;@@D BR IF NO
1360 011512  012703  000144                MOV     #RECMSG-EXPMSG,R3;@@D
1361 011516                                PRINTF  #DEBUGMSG               ;@@D
     011516  012746  011632                MOV     #DEBUGMSG,-(SP)
     011522  012746  000001                MOV     #1,-(SP)
     011526  010600                        MOV     SP,R0
     011530  104417                        TRAP    C$PNTF
     011532  062706  000004                ADD     #4,SP
1362 011536  010037  002304        5$:     MOV     R0,RCVHIADD             ;SAVE RECV HIGH ADDRESS
1363 011542  010137  002306                MOV     R1,RCVLOAD              ;SAVE RECV LOW ADDRESS
1364 011546  005737  003134                TST     KTENABLE                ;TESTING ABOVE 28K?
1365 011552  001403                        BEQ     10$                     ;BR IF NO
1366 011554  004737  017316                JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
1367 011560  010001                        MOV     R0,R1                   ;GET RETURNED ADDRESS BIASED TO PAR6
1368 011562  005004                10$:    CLR     R4                      ;WORD IN BUFFER
```

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```
1369 011564  005005                        CLR     R5                  ;CLEAR ERROR SEEN FLAG
1370 011566  111264  002322          15$:   MOVB    (R2),EXPMSG(R4)     ;SAVE EXPD FOR ERROR REPORT
1371 011572  111164  002466                 MOVB    (R1),RECMSG(R4)     ;SAVE RECV FOR ERROR REPORT
1372 011576  122221                        CMPB    (R2)+,(R1)+         ;EXPD EQUAL RECV?
1373 011600  001401                        BEQ     25$                 ;BR IF YES
1374 011602  005205                        INC     R5                  ;SET ERROR SEEN FLAG
1375 011604  062704  000001          25$:   ADD     #1,R4               ;POINT TO NEXT BYTE
1376 011610  020403                        CMP     R4,R3               ;DONE ALL BYTES?
1377 011612  002001                        BGE     50$                 ;BR IF YES
1378 011614  000764                        BR      15$                 ;DO NEXT BYTE
1379 011616  005705                  50$:  TST     R5                  ;ANY ERRORS SEEN?
1380 011620  001402                        BEQ     55$                 ;BR IF NO
1381 011622  000241                        CLC                         ;SET FAILURE
1382 011624  000401                        BR      60$                 ;
1383 011626  000261                  55$:  SEC                         ;SET SUCCESS
1384 011630  000207                  60$:  RTS     PC                  ;RETURN
1385
1386 011632  120  122  117  DEBUGMSG:       .ASCIZ  'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-';@@D
1387 011722  045  116  045  FERCM:   .ASCII  /%N%A  %%%/
1388 011733  040  040  124  ERCM:    .ASCIZ  /  TSSR ERROR CODE REC'D = /
1389 011766  056  056  056  SIMSG:   .ASCIZ  /.... AFTER DOING SOFT INIT/
1390 012021  124  105  123  TINERR:  .ASCIZ  /TEST: .../
1391                                         .EVEN
1392
1393                                 ;+
1394                                 ;
1395                                 ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
1396                                 ;
1397                                 ;INPUT:
1398                                 ;
1399                                 ;       R1      CONTENTS OF TSSR AT ERROR
1400                                 ;
1401                                 ;SIDE EFFECTS:
1402                                 ;
1403                                 ;       EXECUTES DROP UNIT TO CEASE TESTING
1404                                 ;
1405                                 ;-
1406
1407 012034                                      BGNMSG  SFIMSG
      012034                         SFIMSG::
1408 012034  004737  006024                     JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1409 012040  004737  017202                     JSR     PC,CKDROP       ;DROP UNIT, IF ALLOWED
1410 012044                                     ENDMSG
      012044                         L10003:
      012044  104423                           TRAP    C$MSG
1411
1412                                 ;+
1413                                 ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1414                                 ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
1415                                 ;
1416                                 ;INPUTS:
1417                                 ;
1418                                 ;       R1      TSSR CONTENTS
1419                                 ;       R4      ADDRESS OF COMMAND PACKET
1420                                 ;
1421                                 ;-
1422
```

```
1423 012046                                    BGNMSG  PKTSSR
     012046                                PKTSSR::
1424 012046  004737  006024                       JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1425 012052  012700  000004                       MOV     #4,R0           ;NO. OF WORDS IN PACKET
1426 012056  004737  007370                       JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
1427 012062                                       ENDMSG
     012062                                L10004:
     012062  104423                               TRAP    C$MSG
1428
1429                                           ;+
1430                                           ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1431                                           ;TSSR AND A GET STATUS COMMAND PACKET.
1432                                           ;
1433                                           ;INPUTS:
1434                                           ;
1435                                           ;       R1      TSSR CONTENTS
1436                                           ;       R4      ADDRESS OF COMMAND PACKET
1437                                           ;
1438                                           ;-
1439
1440 012064                                    BGNMSG  PKTGETS
     012064                                PKTGETS::
1441 012064  004737  006024                       JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1442 012070  012700  000002                       MOV     #2,R0           ;NO. OF WORDS IN GET STATUS PACKET
1443 012074  004737  007370                       JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
1444 012100                                       ENDMSG
     012100                                L10005:
     012100  104423                               TRAP    C$MSG
1445
1446                                           ;+
1447                                           ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
1448                                           ;
1449                                           ;INPUTS:
1450                                           ;
1451                                           ;       R1      TSSR CONTENTS
1452                                           ;       R4      ADDRESS OF COMMAND PACKET
1453                                           ;-
1454
1455 012102                                    BGNMSG  SFFMSG
     012102                                SFFMSG::
1456 012102  004737  006024                       JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1457 012106                                       ENDMSG
     012106                                L10006:
     012106  104423                               TRAP    C$MSG
1458
1459                                           .SBTTL  PKTMES  - PRINT TSSR AND MESSAGE BUFFER
1460                                           ;+
1461                                           ;
1462                                           ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
1463                                           ;BUFFER FOR ERROR REPORTS
1464                                           ;
1465                                           ;INPUTS:
1466                                           ;
1467                                           ;       R1      CONTENTS OF TSSR
1468                                           ;       R2      LOW ORDER MESSAGE BUFFER
1469                                           ;       R3      HIGH ORDER MESSAGE BUFFER ADDRESS
1470                                           ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
```

```
1471
1472 012110                                ;-
     012110                                        BGNMSG  PKTMES
                                           PKTMES::
1473 012110  004737  006024                        JSR     PC,PRITSSR              ;PRINT CONTENTS OF TSSR
1474 012114  010200                        MOV     R2,R0                   ;LOW ORDER ADDRESS
1475 012116  010301                        MOV     R3,R1                   ;HIGH ORDER ADDRESS
1476 012120  004737  014242                JSR     PC,PRMESS               ;PRINT THE MESSAGE BUFFER
1477 012124                                ENDMSG
     012124                                L10007:
     012124  104423                        TRAP    C$MSG
1478                                       .SBTTL  ADDSSR  - PRINT TEST ADDRESS AND TSSR
1479                                       ;+
1480                                       ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1481                                       ;TSSR AND A MEMORY TEST ADDRESS
1482                                       ;
1483                                       ;INPUTS:
1484                                       ;
1485                                       ;       R5      FIRST DEVICE UNIBUS ADDRESS
1486                                       ;       ERRHI   HIGH ORDER MEMORY TEST ADDRESS
1487                                       ;       ERRLO   LOW ORDER MEMORY TEST ADDRESS
1488                                       ;-
1489
1490 012126                                        BGNMSG  ADDSSR
     012126                                ADDSSR::
1491 012126  004737  010274                        JSR     PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
1492 012132  016501  000002                        MOV     TSSR(R5),R1     ;GET CURRENT TSSR
1493 012136  004737  006024                        JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1494 012142                                ENDMSG
     012142                                L10010:
     012142  104423                        TRAP    C$MSG
1495
1496                                       .SBTTL  MSGEXP - PRINT WRITE CHAR. EXPD-RECV MESSAGE BUFFERS
1497                                       ;+
1498                                       ;
1499                                       ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
1500                                       ;
1501                                       ;IMPLICIT INPUTS:
1502                                       ;
1503                                       ;       EXPMSG  - EXPECTED MESSAGE BUFFER
1504                                       ;       RECMSG  - RECEIVED MESSAGE BUFFER
1505                                       ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1506                                       ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1507                                       ;-
1508 012144                                        BGNMSG  MSGEXP
     012144                                MSGEXP::
1509 012144  012700  000007                        MOV     #7,R0           ;ASSUME NO EXT FEATURES
1510 012150  005737  002226                        TST     EXTFEA          ;EXT FEATURES SET?
1511 012154  001402                        BEQ     5$              ;BR IF NO
1512 012156  012700  000010                        MOV     #8.,R0          ;EXT FEATURE BUFFER IS 8 WORDS
1513 012162  004737  014552                5$:     JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
1514 012166                                ENDMSG
     012166                                L10011:
     012166  104423                        TRAP    C$MSG
1515                                       .SBTTL  FIFEXP  - PRINT FIFO EXP/RECV DATA
1516                                       ;+
1517                                       ;
1518                                       ;PRINT ROUTINE TO PRINT FIFO EXP/RECV DATA
```

```
                                        ;
                                        ;       R1      - BYTE COUNT
                                        ;
                                        ;IMPLICIT INPUTS:
                                        ;
                                        ;       EXPMSG  - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY
                                        ;       RECMSG  - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
                                        ;-
                                                BGNMSG  FIFEXP
012170                                  FIFEXP::
                                                PRINTX  #FIF1MSG,R1     ;PRINT BYTES TRANSFERRED
012170  010146                                 MOV     R1,-(SP)
012172  012746  012242                         MOV     #FIF1MSG,-(SP)
012176  012746  000002                         MOV     #2,-(SP)
012202  010600                                 MOV     SP,R0
012204  104415                                 TRAP    C$PNTX
012206  062706  000006                         ADD     #6,SP
                                                PRINTX  #FIF2MSG        ;PRINT HEADER MSG
012212  012746  012311                         MOV     #FIF2MSG,-(SP)
012216  012746  000001                         MOV     #1,-(SP)
012222  010600                                 MOV     SP,R0
012224  104415                                 TRAP    C$PNTX
012226  062706  000004                         ADD     #4,SP
012232  010100                                 MOV     R1,R0           ;GET BYTE COUNT
012234  004737  015122                         JSR     PC,PRBYTEXP     ;PRINT FIFO BYTES IN ERROR
012240                                          ENDMSG
012240                                  L10012:
012240  104423                                 TRAP    C$MSG
012242     045     116     045        FIF1MSG:        .ASCIZ  '%N%A NUMBER OF BYTES TRANSFERRED = %D2'
012311     045     116     045        FIF2MSG:        .ASCIZ  '%N%A FIFO DATA BYTES IN ERROR:'
                                                .EVEN
                                                .SBTTL  MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
                                        ;*
                                        ;
                                        ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
                                        ;
                                        ;
                                        ;IMPLICIT INPUTS:
                                        ;
                                        ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                        ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                        ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                        ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                        ;-
                                                BGNMSG  MSGSTAT
012350                                  MSGSTAT::
012350  012701  012412                         MOV     #STATCOD,R1     ;ASCII ADDRESS TABLE
012354  012100                          10$:    MOV     (R1)+,R0        ;DONE ALL MSG LINES?
012356  001410                                 BEQ     20$             ;BR IF YES
012360                                          PRINTX  R0              ;PRINT STATUS BIT NAMES
012360  010046                                 MOV     R0,-(SP)
012362  012746  000001                         MOV     #1,-(SP)
012366  010600                                 MOV     SP,R0
012370  104415                                 TRAP    C$PNTX
012372  062706  000004                         ADD     #4,SP
012376  000766                                 BR      10$             ;DO ANOTHER MSG LINE
012400  012700  000012                  20$:    MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
```

MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS

```
1556 012404 004737 014552                  JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
1557 012410                                ENDMSG
     012410                          L10013:
     012410 104423                         TRAP    C$MSG
1558
1559 012412 012430 012472 012563  STATCOD:        .WORD   1$,2$,3$,4$,5$,6$,0
1560 012430    045    116    045  1$:.ASCIZ  '<N><A Tape Bus Signals in Word #8:'
1561 012472    045    116    045  2$:.ASCIZ  '<N><A          PARERR<15>  IEOT  <12>  IFMK  <9>  IRDY<6>  IRWD<2>'
1562 012563    045    116    045  3$:.ASCIZ  '<N><A          IRESV2<14>  IIDENT<11>  IHER  <8>  IONL<5>  IFBY<1>'
1563 012654    045    116    045  4$:.ASCIZ  '<N><A          IRESV1<13>  ICER  <10>  ISPEED<7>  ILDP<4>  IFPT<0>'
1564 012745    045    116    045  5$:.ASCIZ  '<N><A Tape Bus Signals in Word #9:'
1565 013007    045    116    045  6$:.ASCIZ  '<N><A          DATMIS<7>  ILW<6>  OUTRDY<5>  INRDY<4>'
1566                                       .EVEN
1567
1568                                       .SBTTL  MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
1569                                ;+
1570                                ;
1571                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
1572                                ;
1573                                ;IMPLICIT INPUTS:
1574                                ;
1575                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
1576                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
1577                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1578                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1579                                ;-
1580 013064                               BGNMSG  MSGLOOP
     013064                          MSGLOOP::
1581 013064 012701 013126                 MOV     #LOOPCOD,R1     ;ASCII ADDRESS TABLE
1582 013070 012100                 10$:   MOV     (R1)+,R0        ;DONE ALL MSG LINES?
1583 013072 001410                        BEQ     20$             ;BR IF YES
1584 013074                               PRINTX  R0              ;PRINT STATUS BIT NAMES
     013074 010046                        MOV     R0,-(SP)
     013076 012746 000001                 MOV     #1,-(SP)
     013102 010600                        MOV     SP,R0
     013104 104415                        TRAP    C$PNTX
     013106 062706 000004                 ADD     #4,SP
1585 013112 000766                        BR      10$             ;DO ANOTHER MSG LINE
1586 013114 012700 000012          20$:   MOV     #10.,R0         ;NUMBER OF WCRDS IN A READ STATUS BUFFER
1587 013120 004737 014552                 JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
1588 013124                               ENDMSG
     013124                          L10014:
     013124 104423                        TRAP    C$MSG
1589
1590 013126 013146 013221 013320  LOOPCOD:        .WORD   1$,2$,3$,4$,5$,6$,7$,0
1591 013146    045    116    045  1$:.ASCIZ  '<N><A Tape Bus Loopback Signals in Word #8:'
1592 013221    045    116    045  2$:.ASCIZ  '<N><A       PARERR<15>          IRESV2<14>          IRESV1<13>'
1593 013320    045    116    045  3$:.ASCIZ  '<N><A IHISP=>IEOT<12>    IWRT=>IIDENT<11>    IREV  =>ICER  <10>'
1594 013417    045    116    045  4$:.ASCIZ  '<N><A IWFM =>IFMK<09>    IEDIT=>IHER <08>    IFAD  =>ISPEED<07>'
1595 013516    045    116    045  5$:.ASCIZ  '<N><A ITADO=>IRDY<06>    ITAD1=>IONL <05>    IERASE=>ILDP  <04>'
1596 013615    045    116    045  6$:.ASCIZ  '<N><A IREW =>IDBY<03>    IRWU =>IRWD <02>    IFEN  =>IFBY  <01>'
1597 013714    045    116    045  7$:.ASCIZ  '<N><A IGO  =>IFPT<00>'
1598                                      .EVEN
1599                                      .SBTTL  MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
1600                                ;+
1601                                ;
1602                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
```

MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER

```
1603                                    ;
1604                                    ;
1605                                    ;IMPLICIT INPUTS:
1606                                    ;
1607                                    ;       EXPMSG  - EXPECTED MESSAGE BUFFER
1608                                    ;       RECMSG  - RECEIVED MESSAGE BUFFER
1609                                    ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1610                                    ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1611                                    ;-
1612 013742                                     BGNMSG  MSGSUB
     013742                             MSGSUB::
1613 013742  012700  000012                     MOV     #10.,R0         ;SIZE OF WRITE SUBSYSTEM BUFFER
1614 013746  004737  014552                     JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
1615 013752                                     ENDMSG
     013752                             L10015:
     013752  104423                             TRAP    C$MSG
1616
1617                                            .SBTTL  MEMADD - PRINT MEMORY ADDRESS DATA ERROR
1618                                    ;*
1619                                    ;
1620                                    ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
1621                                    ;
1622                                    ;IMPLICIT INPUTS:
1623                                    ;
1624                                    ;       ERRHI   - MEMORY ERROR HIGH ORDER ADDRESS
1625                                    ;       ERRLO   - MEMORY ERROR LOW ORDER ADDRESS
1626                                    ;       EXP     - EXPECTED DATA
1627                                    ;       RECV    - RECEIVED DATA
1628                                    ;-
1629 013754                                     BGNMSG  MEMADD
     013754                             MEMADD::
1630 013754  004737  010160                     JSR     PC,PRIADD       ;PRINT MEMORY ADDRESS IN ERROR
1631 013760  013701  002232                     MOV     EXPD,R1         ;GET EXPD DATA
1632 013764  013702  002234                     MOV     RECV,R2         ;GET RECEIVED DATA
1633 013770  004737  007742                     JSR     PC,PRIXOR       ;PRINT EXPD/RECV
1634 013774                                     ENDMSG
     013774                             L10016:
     013774  104423                             TRAP    C$MSG
1635                                            .SBTTL  PRAMPKT - PRINT RAM AND PACKET DATA
1636                                    ;*
1637                                    ;
1638                                    ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1639                                    ;WHEN THE RAM DATA DOES NOT MATCH.
1640                                    ;
1641                                    ;INPUTS:
1642                                    ;
1643                                    ;       R4      POINTER TO COMMAND PACKET
1644                                    ;
1645                                    ;IMPLICIT INPUTS:
1646                                    ;
1647                                    ;       RAMDATA         DATA AS READ FROM THE RAM
1648                                    ;       RAMSIZ          NUMBER OF BYTES IN PACKET
1649                                    ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
1650                                    ;
1651                                    ;IMPLICIT OUTPUTS:
1652                                    ;
1653                                    ;       RAMSIZ  SET TO 0
```

```
                                        ;-
                                        ;
013776                                  PRAMPKT:
013776                                          SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
014002  012701  002242                          MOV     #RAMDATA,R1             ;DATA FROM THE RAM
014006  005002                                  CLR     R2                      ;INIT BYTE NUMBER
014010  122124                          5$:     CMPB    (R1)+,(R4)+             ;COMPARE EXPECTED, RECEIVED
014012  001005                                  BNE     7$                      ;BR IF NO MATCH
014014                                          FORCERROR       7$,NOTSSR
014024  000436                                  BR      10$                     ;BRD
014026  116105  177777                  7$:     MOVB    -1(R1),R5               ;GET RECV RAM DATA
014032  116403  177777                          MOVB    -1(R4),R3               ;GET EXPD PACKET DATA
014036                                          XOR     R5,R3                   ;XOR EXPD/RECV
014046  042703  177400                          BIC     #177400,R3              ;LOW BYTE ONLY
014052  116137  177777  002234                  MOVB    -1(R1),RECV             ;GET RECEIVED RAM DATA
014060  116437  177777  002232                  MOVB    -1(R4),EXPD             ;GET EXPECTED RAM DATA
014066                                          PRINTB  #RAMASC,R2,RECV,EXPD,R3
014066  010346                                  MOV     R3,-(SP)
014070  013746  002232                          MOV     EXPD,-(SP)
014074  013746  002234                          MOV     RECV,-(SP)
014100  010246                                  MOV     R2,-(SP)
014102  012746  014156                          MOV     #RAMASC,-(SP)
014106  012746  000005                          MOV     #5,-(SP)
014112  010600                                  MOV     SP,R0
014114  104414                                  TRAP    C$PNTB
014116  062706  000014                          ADD     #14,SP
014122  005202                          10$:    INC     R2                      ;UPDATE BYTE COUNT
014124  005737  002302                          TST     RAMSIZ                  ;DEFAULT TO 8.?
014130  001404                                  BEQ     15$                     ;BR IF YES
014132  020237  002302                          CMP     R2,RAMSIZ               ;DONE ALL BYTES?
014136  003724                                  BLE     5$                      ;BR IF NO
014140  000403                                  BR      25$                     ;
014142  020227  000010                  15$:    CMP     R2,#8.                  ;DONE DEFAULT NUMBER OF BYTES?
014146  002720                          20$:    BLT     5$                      ;BR IF NO
014150  005037  002302                  25$:    CLR     RAMSIZ                  ;SET DEFAULT RAMSIZ
014154  000207                                  RTS     PC                      ;RETURN
                                        ;
014156      045     116     045         RAMASC: .ASCIZ  '%N%A  BYTE: %D2%A RAM: %03%A Packet: %03%A XOR:%03'
                                                .EVEN
                                                .SBTTL  PRMESS  - PRINT CONTENTS OF MESSAGE BUFFER
                                        ;+
                                        ;
                                        ;THIS ROUTINE PRINTS THE CONTENTS OF
                                        ;THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE
                                        ;TSV-05.
                                        ;
                                        ;INPUT:
                                        ;
                                        ;       R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
                                        ;       R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
                                        ;       NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
                                        ;
                                        ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
                                        ;
                                        ;-
                                        ;
014242                                  PRMESS:
```

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```
                                                SAVREG                  ;SAVE THE REGISTERS
                                                MOV     R0,R5           ;SAVE LOW ORDER ADDRESS
                                                TST     KTENABLE        ;ADDRESS ABOVE 28K?
                                                BNE     10$             ;BR IF YES
                                                CLR     R1              ;SET HIGH ORDER ADDRESS TO 0
                                        10$:    MOV     R1,R3           ;SAVE HIGH ORDER ADDRESS
                                                ROL     R0              ;SHIFT BIT15 TO C BIT
                                                ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
                                                PRINTX  #PROASC,R1,R5    ;PRINT MESSAGE BUFFER ADDRESS
                                                MOV     R5,-(SP)
                                                MOV     R1,-(SP)
                                                MOV     #PROASC,-(SP)
                                                MOV     #3,-(SP)
                                                MOV     SP,R0
                                                TRAP    C$PNTX
                                                ADD     #10,SP
                                                PRINTX  #PR1ASC         ;PRINT HEADER FOR CONTENTS
                                                MOV     #PR1ASC,-(SP)
                                                MOV     #1,-(SP)
                                                MOV     SP,R0
                                                TRAP    C$PNTX
                                                ADD     #4,SP
                                                CLR     R4              ;NUMBER OF THE NEXT WORD
                                                MOV     R5,R1           ;COPY LOW ORDER ADDRESS
                                                MOV     R3,R0           ;COPY HIGH ORDER ADDRESS
                                                BEQ     20$             ;BR IF NOT ABOVE 28K
                                                JSR     PC,SETMAP       ;SETUP PAR ADDRESS IN R0
                                                MOV     R0,R5           ;GET PAR FORMAT ADDRESS ABOVE 28K
                                        20$:    PRINTX  #PRASC,R4,(R5)+ ;PRINT THE CONTENTS OF MEMORY BUFFER
                                                MOV     (R5)+,-(SP)
                                                MOV     R4,-(SP)
                                                MOV     #PRASC,-(SP)
                                                MOV     #3,-(SP)
                                                MOV     SP,R0
                                                TRAP    C$PNTX
                                                ADD     #10,SP
                                                INC     R4              ;NUMBER OF THE NEXT
                                                CMP     R4,#7           ;DONE ALL YET ?
                                                BGT     50$             ;BRANCH IF ALL DONE
                                                BLT     20$             ;PRINT FIRST 7 WORDS
                                                BIT     #X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
                                                BNE     20$             ;PRINT EXTENDED STATUS WORD
                                        50$:    RTS     PC              ;RETURN

 014420     045     116     045  PROASC: .ASCIZ  '%N%A Message Buffer Address = %O1%O5'
 014465     045     116     045  PR1ASC: .ASCIZ  '%N%A Message Buffer Contents:'
 014523     045     116     045  PRASC:  .ASCIZ  '%N%A    Word%D1%A: %O'
                                         .EVEN
                                         .SBTTL  PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
                                ;+
                                ;
                                ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
                                ;
                                ;       R0      - NUMBER OF WORDS IN BUFFER
                                ;
                                ;IMPLICIT INPUTS:
                                ;
```

PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS

```
1740                                    ;       EXPMSG  - EXPECTED MESSAGE BUFFER
1741                                    ;       RECMSG  - RECEIVED MESSAGE BUFFER
1742                                    ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1743                                    ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1744                                    ;-
1745 014552                            PRMSGEXP::
1746 014552                                    SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
1747 014556  010005                            MOV     R0,R5           ;SAVE NUMBER OF WORDS
1748 014560  013700  002306                    MOV     RCVLOADD,R0     ;GET RECV LOW ADDRESS
1749 014564  010004                            MOV     R0,R4           ;COPY LOW ADDRESS
1750 014566  013701  002304                    MOV     RCVHIADD,R1     ;GET RECV HIGH ADDRESS
1751 014572  006100                            ROL     R0              ;SHIFT BIT15 TO C BIT
1752 014574  006101                            ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
1753 014576                                    PRINTX  #PRMSG0,R1,R4   ;PRINT MESSAGE BUFFER ADDRESS
     014576  010446                            MOV     R4,-(SP)
     014600  010146                            MOV     R1,-(SP)
     014602  012746  014732                    MOV     #PRMSG0,-(SP)
     014606  012746  000003                    MOV     #3,-(SP)
     014612  010600                            MOV     SP,R0
     014614  104415                            TRAP    C$PNTX
     014616  062706  000010                    ADD     #10,SP
1754 014622                                    PRINTX  #PRMSG1         ;PRINT HEADER FOR CONTENTS
     014622  012746  014777                    MOV     #PRMSG1,-(SP)
     014626  012746  000001                    MOV     #1,-(SP)
     014632  010600                            MOV     SP,R0
     014634  104415                            TRAP    C$PNTX
     014636  062706  000004                    ADD     #4,SP
1755 014642  005004                            CLR     R4              ;NUMBER OF THE CURRENT WORD
1756 014644  012701  002322                    MOV     #EXPMSG,R1      ;GET EXPD BUFFER ADDRESS
1757 014650  012702  002466                    MOV     #RECMSG,R2      ;GET RECV BUFFER ADDRESS
1758 014654  011100                    20$:    MOV     (R1),R0         ;GET EXPD
1759 014656  011203                            MOV     (R2),R3         ;GET RECV
1760 014660                                    XOR     R0,R3           ;XOR EXPD/RECV
1761 014670                                    PRINTX  #PRMSG2,R4,(R1)+,(R2)+,R3
     014670  010346                            MOV     R3,-(SP)
     014672  012246                            MOV     (R2)+,-(SP)
     014674  012146                            MOV     (R1)+,-(SP)
     014676  010446                            MOV     R4,-(SP)
     014700  012746  015035                    MOV     #PRMSG2,-(SP)
     014704  012746  000005                    MOV     #5,-(SP)
     014710  010600                            MOV     SP,R0
     014712  104415                            TRAP    C$PNTX
     014714  062706  000014                    ADD     #14,SP
1762 014720  005204                            INC     R4              ;NUMBER OF THE NEXT
1763 014722  020405                            CMP     R4,R5           ;DONE ALL YET?
1764 014724  002001                            BGE     50$             ;BR IF YES
1765 014726  000752                            BR      20$             ;DO ANOTHER
1766 014730  000207                    50$:    RTS     PC              ;RETURN
1767
1768 014732     045     116     045    PRMSG0: .ASCIZ  '%N%A Message Buffer Address = %01%05'
1769 014777     045     116     045    PRMSG1: .ASCIZ  '%N%A Message Buffer Contents:'
1770 015035     045     116     045    PRMSG2: .ASCIZ  '%N%A   WORD #%D2%A  EXPD: %06%A  RECV: %06%A  XOR: %06'
1771                                            .EVEN
1772                                            .SBTTL  PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
1773                                    ;+
1774                                    ;
1775                                    ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
```

```
1776                                    ;       ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
1777                                    ;
1778                                    ;       R0      - NUMBER OF BYTES IN BUFFER
1779                                    ;
1780                                    ;IMPLICIT INPUTS:
1781                                    ;
1782                                    ;       EXPMSG  - EXPECTED MESSAGE BUFFER
1783                                    ;       RECMSG  - RECEIVED MESSAGE BUFFER
1784                                    ;-
1785 015122                            PRBYTEXP::
1786 015122                                    SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
1787 015126  010005                            MOV     R0,R5                   ;SAVE NUMBER OF BYTES
1788 015130  005037  002320                    CLR     PRMNO                   ;INIT ERROR COUNT
1789 015134  005004                            CLR     R4                      ;NUMBER OF THE CURRENT BYTE
1790 015136  012701  002322                    MOV     #EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
1791 015142  012702  002466                    MOV     #RECMSG,R2              ;GET RECV BUFFER ADDRESS
1792 015146  111100                    20$:    MOVB    (R1),R0                 ;GET EXPD BYTE
1793 015150  042700  177400                    BIC     #^C<377>,R0             ;CLEAR UPPER BYTE
1794 015154  110037  015470                    MOVB    R0,PRBEXP               ;SAVE FOR ERROR REPORT
1795 015160  111203                            MOVB    (R2),R3                 ;GET RECV BYTE
1796 015162  042703  177400                    BIC     #^C<377>,R3             ;CLEAR UPPER BYTE
1797 015166  110337  015472                    MOVB    R3,PRBREC               ;FOR ERROR REPORT
1798 015172                                    XOR     R0,R3                   ;XOR EXPD/RECV
1799 015202  122122                            CMPB    (R1)+,(R2)+             ;EXPD = RECV?
1800 015204  001431                            BEQ     30$                     ;BR IF YES
1801 015206  005237  002320                    INC     PRMNO                   ;UPDATE ERROR COUNT
1802 015212  023727  002320  000010            CMP     PRMNO,#8.               ;PRINTED 8?
1803 015220  101023                            BHI     30$                     ;BR IF YES
1804 015222                            27$:    PRINTX  #PRBMSG,R4,PRBEXP,PRBREC,R3
     015222  010346                            MOV     R3,-(SP)
     015224  013746  015472                    MOV     PRBREC,-(SP)
     015230  013746  015470                    MOV     PRBEXP,-(SP)
     015234  010446                            MOV     R4,-(SP)
     015236  012746  015336                    MOV     #PRBMSG,-(SP)
     015242  012746  000005                    MOV     #5,-(SP)
     015246  010600                            MOV     SP,R0
     C15250  104415                            TRAP    C$PNTX
     015252  062706  000014                    ADD     #14,SP
1805 015256                                    FORCEXIT        50$             ;@@D
1806 015266  000404                            BR      35$                     ;@@D
1807 015270                            30$:
1808 015270                                    FORCERROR  27$,NOTSSR           ;@@D
1809 015300                            35$:                                    ;@@D
1810 015300  005204                            INC     R4                      ;NUMBER OF THE NEXT
1811 015302  020405                            CMP     R4,R5                   ;DONE ALL YET?
1812 015304  002001                            BGE     50$                     ;BR IF YES
1813 015306  000717                            BR      20$                     ;DO ANOTHER
1814 015310                            50$:    PRINTX  #PRBTOT,PRMNO           ;PRINT TOTAL ERROR COUNT
     015310  013746  002320                    MOV     PRMNO,-(SP)
     015314  012746  015423                    MOV     #PRBTOT,-(SP)
     015320  012746  000002                    MOV     #2,-(SP)
     015324  010600                            MOV     SP,R0
     015326  104415                            TRAP    C$PNTX
     015330  062706  000006                    ADD     #6,SP
1815 015334  000207                            RTS     PC                      ;RETURN
1816
1817 015336      045      116      045  PRBMSG: .ASCIZ  '%N%A   BYTE #%D2%A  EXPD: %03%A  RECV: %03%A  XOR: %03'
```

```
                                        PRBTOT: .ASCIZ  '#N#A NUMBER OF BYTES IN ERROR = #D2'
                                                .EVEN
015470  000000                          PRBEXP: .WORD   0                       ;EXPD
015472  000000                          PRBREC: .WORD   0                       ;RECV
                                                .SBTTL  EXPREC  - PRINT EXPD/RECV WORD DATA
                                        ;+
                                        ;
                                        ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      RECEIVED DATA
                                        ;       R2      EXPECTED DATA
                                        ;
                                        ;-

                                                BGNMSG  EXPREC
015474                                  EXPREC::
015474  004737  007742                          JSR     PC,PRIXOR               ;PRINT THE DATA
015500                                          ENDMSG
015500                                  L10017:
015500  104423                                  TRAP    C$MSG
                                                .SBTTL  EXPBREC - PRINT EXPD/RECV BYTE DATA
                                        ;+
                                        ;
                                        ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      RECEIVED DATA BYTE
                                        ;       R2      EXPECTED DATA BYTE
                                        ;
                                        ;-

                                                BGNMSG  EXPBREC
015502                                  EXPBREC::
015502  004737  007612                          JSR     PC,PRIBXOR              ;PRINT THE DATA
015506                                          ENDMSG
015506                                  L10020:
015506  104423                                  TRAP    C$MSG


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

```
                                ;
                                ;IMPLICIT OUTPUTS:
                                ;
                                ;       RAMSIZ  SET TO 0
                                ;-
                                
                                        BGNMSG  RAMERR
                                RAMERR::
015510  004737  013776                  JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
                                        ENDMSG
                                L10021:
015514  104423                          TRAP    C$MSG
                                
                                        .SBTTL  RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
                                ;*
                                ;
                                ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
                                ;
                                ;INPUTS:
                                ;
                                ;       R4      POINTER TO COMMAND PACKET
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       RAMDATA         DATA AS READ FROM THE RAM
                                ;       RAMSIZ          NUMBER OF BYTES IN PACKET
                                ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
                                ;       ERRHI           HIGH ORDER TEST ADDRESS
                                ;       ERRLO           LOW ORDER TEST ADDRESS
                                ;
                                ;IMPLICIT OUTPUTS:
                                ;
                                ;       RAMSIZ  SET TO 0
                                ;-
                                
                                        BGNMSG  RAMTADD
                                RAMTADD::
015516  004737  010274                  JSR     PC,PRITADD      ;PRINT TEST ADDRESS
015522  004737  013776                  JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
                                        ENDMSG
                                L10022:
015526  104423                          TRAP    C$MSG
                                
                                        .SBTTL  RAMEXP  - PRINT RAM EXPD/RECV DATA
                                ;*
                                ;
                                ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA
                                ;       R2      EXPECTED DATA
                                ;       R4      CONTROLLER RAM ADDRESS
                                ;-
                                
                                        BGNMSG  RAMEXP
                                RAMEXP::
```

RAMEXP - PRINT RAM EXPD/RECV DATA

```
1919 015530  042701  177400          BIC     #^C<377>,R1             ;SAVE EXPD RAM DATA BYTE
1920 015534  042702  177400          BIC     #^C<377>,R2             ;SAVE EXPD RAM DATA BYTE
1921 015540  004737  010066          JSR     PC,PRIRAM               ;PRINT THE RAM ADDRESS
1922 015544  004737  007742          JSR     PC,PRIXOR               ;PRINT THE DATA
1923 015550                          ENDMSG
     015550                     L10023:
     015550  104423                 TRAP    C$MSG
1924
1925                                .SBTTL  TIMEXP  - PRINT TIMER A,B AND EXP/REC
1926                           ;+
1927                           ;
1928                           ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1929                           ;AND TIMER A,B HEADER MESSAGE
1930                           ;
1931                           ;INPUTS:
1932                           ;
1933                           ;       R1      RECEIVED DATA
1934                           ;       R2      EXPECTED DATA
1935                           ;-
1936
1937 015552                          BGNMSG  TIMEXP
     015552                     TIMEXP::
1938 015552                          PRINTX  #TIMSGO                 ;PRINT HEADER
     015552  012746  015600          MOV     #TIMSGO,-(SP)
     015556  012746  000001          MOV     #1,-(SP)
     015562  010600                  MOV     SP,R0
     015564  104415                  TRAP    C$PNTX
     015566  062706  000004          ADD     #4,SP
1939 015572  004737  007742          JSR     PC,PRIXOR               ;PRINT THE DATA
1940 015576                          ENDMSG
     015576                     L10024:
     015576  104423                 TRAP    C$MSG
1941
1942 015600    045    116    045  TIMSGO: .ASCIZ  '<NL>A TIMER A STATUS IS IN BIT 3<NL>A TIMER B STATUS IS IN BIT 2'
1943                                .EVEN
1944                                .SBTTL  BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS
1945
1946                           ;+
1947                           ;
1948                           ;PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
1949                           ;
1950                           ;INPUTS:
1951                           ;
1952                           ;       R1      CONTENTS OF TSSR
1953                           ;       R2      DATA WRITTEN (8 BITS)
1954                           ;
1955                           ;-
1956
1957 015700                          BGNMSG  BADSSR
     015700                     BADSSR::
1958 015700  010246                  MOV     R2,-(SP)                ;SAVE DATA TRANSFERRED
1959 015702  042702  177400          BIC     #177400,R2              ;GET JUST ONE BYTE
1960 015706                          PRINTB  #XFERASC,R2
     015706  010246                  MOV     R2,-(SP)
     015710  012746  015740          MOV     #XFERASC,-(SP)
     015714  012746  000002          MOV     #2,-(SP)
     015720  010600                  MOV     SP,R0
```

BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

```
        015722  104414                          TRAP    C$PNTB
        015724  062706  000006                  ADD     #6,SP
  1961  015730  012602                          MOV     (SP)+,R2                ;RESTORE R2
  1962  015732  004737  006024                  JSR     PC,PRITSSR              ;DECODE TSSR CONTENTS
  1963  015736                                  ENDMSG
        015736                          L10025:
        015736  104423                          TRAP    C$MSG
  1964  015740     045     116     045  XFERASC:        .ASCIZ  '%N%A Data Transferred = %03'
  1965                                          .SBTTL  GLOBAL SUBROUTINES SECTION
  1966
  1967                                  ;++
  1968                                  ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
  1969                                  ; THAT ARE USED IN MORE THAN ONE TEST.
  1970                                  ;--
  1971                                          .SBTTL  SOFINIT - SOFT INITIALIZE OF CONTROLLER
  1972
  1973                                  ;+
  1974                                  ;
  1975                                  ;ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
  1976                                  ;BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
  1977                                  ;THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
  1978                                  ;DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
  1979                                  ;
  1980                                  ;INPUTS:
  1981                                  ;
  1982                                  ;       R5      ADDRESS OF FIRST REGISTER
  1983                                  ;
  1984                                  ;OUTPUTS:
  1985                                  ;
  1986                                  ;       R0      CONTENTS OF TSSR, IF ERROR
  1987                                  ;       CARRY   SET IF INIT WAS OKAY
  1988                                  ;               CLEAR IF FATAL ERROR
  1989                                  ;
  1990                                  ;CALLING SEQUENCE:
  1991                                  ;
  1992                                  ;       MOV     #ADDRESS,R5
  1993                                  ;       JSR     PC,SOFINIT
  1994                                  ;       BCS     CONTINUE
  1995                                  ;       ERRDF                           ;REPORT FATAL ERROR
  1996                                  ;
  1997                                  ;-
  1998
  1999  015774                          SOFINIT::
  2000  015774                                  SAVREG                          ; SAVE THE REGISTERS
  2001  016000  012765  000000  000002          MOV     #0,TSSR(R5)             ; DO THE INIT.
  2002  016006  004737  016250                  JSR     PC,WAITF                ; WAIT FOR SSR
  2003  016012  016500  000002                  MOV     TSSR(R5),R0             ;GET THE TSSR REGISTER
  2004  016016  010004                          MOV     R0,R4                   ;TSSR CONTENTS
  2005  016020  042704  176277                  BIC     #^C<HIADDR!OFL>,R4
  2006  016024  052704  002200                  BIS     #SSR!NBA,R4             ;R4 HAS EXPECTED CONTENTS
  2007  016030  020400                          CMP     R4,R0                   ;ONLY EXPECTED BITS SET ?
  2008  016032  001402                          BEQ     5$                      ;BRANCH IF OKAY
  2009  016034  000241                          CLC                             ;CLEAR THE CARRY FOR ERROR
  2010  016036  000401                          BR      10$                     ;GO TO EXIT
  2011  016040  000261                  5$:     SEC                             ;SET THE CARRY BIT
  2012  016042  000207                  10$:    RTS     PC                      ;RETURN TO CALLER
  2013                                          .SBTTL  CHKAMB  - CHECK TSSR FOR AMBIGUITY
```

CHKAMB - CHECK TSSR FOR AMBIGUITY

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

016044                          CHKAMB:
016044                                  SAVREG                  ;SAVE THE GENERAL REGISTERS
016050  010004                          MOV     R0,R4           ;CONTENTS OF TSSR
016052  032700  100000                  BIT     #SC,R0          ;IS BIT 15 SET ?
016056  001004                          BNE     5$              ;BRANCH IF YES
016060  032700  174077                  BIT     #^C<NBA!OFL!SSR!HIADDR>,R0      ;ANY OTHER BITS SET ?
016064  001023                          BNE     40$             ;MUST BE AN ERROR
016066  000424                          BR      45$             ;RETURN WITH SUCCESS
016070  032700  000200          5$:     BIT     #SSR,R0         ;IS READY BIT SET ?
016074  001011                          BNE     10$             ;BRANCH IF READY BIT IS SET.
016076  032700  000040                  BIT     #BIT5,R0        ;IS FATAL ERROR BIT SET ?
016102  001414                          BEQ     40$             ;ERROR IF NOT
016104  042704  177761                  BIC     #^CTERCLS,R4    ;CLEAR ALL BUT TERMINATION CODE
016110  020427  000016                  CMP     R4,#16          ;ALL THREE BITS MUST BE SET
016114  001007                          BNE     40$             ;ERROR IF NOT SET
016116  000410                          BR      45$             ;OK IF ALL ARE SET
016120  032700  000040          10$:    BIT     #BIT5,R0        ;IS FATAL ERROR BIT SET ?
016124  001405                          BEQ     45$             ;ERROR IF BIT IS SET WITH SSR
016126  032700  000006                  BIT     #BIT2!BIT1,R0   ;IS THIS A FUNCTION REJECT
016132  001002                          BNE     45$             ;BR, IF TSSR IS OK
016134  000241                  40$:    CLC                     ;AMBIGUOUS CONTENTS
016136  000401                          BR      50$
016140  000261                  45$:    SEC                     ;SHOW SUCCESS - NO AMBIGUITY
016142  000207                  50$:    RTS     PC              ;RETURN TO CALLER
                                        .SBTTL  ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS
                                ;
                                ; DEFAULT DISPLAY INTERRUPT HANDLERS.
                                ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
                                ; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
                                ;
                                ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
                                ;
                                        IOKCKIN=BIT7    ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
                                        IOKSTP=BIT0     ; EXPECT "STOP" INTERRUPT.
                                ;
                                ;INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
016144  000             INTMASK:        .BYTE   0
```

```
2071                                    ;INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
2072 016145     000                     INTFLAG:        .BYTE   0
2073
2074                                    ;SAVED INTERRUPT VECTOR:
2075 016146  000000                     INTVEC: .WORD   0
2076                                    ;SAVE CPU PC
2077 016150  000000                     INTCPC: .WORD   0
2078
2079                                    ;SUBROUTINE TO ENABLE INTERRUPTS:
2080 016152  010046                     ENAINT: MOV     R0,-(SP)        ;SAVE R0
2081 016154  013700  002210                     MOV     IVEC,R0         ;GET POINTER TO VECTORS
2082 016160  012720  016216                     MOV     #INTR,(R0)+     ;SET UP INTERRUPT VECTOR
2083 016164  012720  000300                     MOV     #PRI06,(R0)+
2084 016170  012600                             MOV     (SP)+,R0        ;RESTORE R0
2085 016172  011646                             MOV     (SP),-(SP)
2086 016174  012766  000000  000002             MOV     #0,2(SP)        ;SET CPU TO LEVEL 0
2087 016202  000002                             RTI
2088
2089                                    ;SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 6)
2090 016204  011646                     DSBINT: MOV     (SP),-(SP)
2091 016206  012766  000300  000002             MOV     #PRI06,2(SP)
2092 016214  000002                             RTI
2093                                            .SBTTL  INTR    - INTERRUPT HANDLERS
2094
2095 016216                                     BGNSRV  INTR            ;DEFINE INTERRUPT ENTRY
     016216                             INTR::
2096 016216  012737  000001  002224             MOV     #1,INTRECV      ;SET FLAG TO SHOW INTERRUPT RECEIVED
2097 016224  105037  016145                     CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2098 016230  132737  000001  016144             BITB    #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
2099 015236  001003                             BNE     1$              ;BR IF YES
2100 016240  152737  000001  016145             BISB    #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.
2101
2102                                    ;SAVE REGISTERS, MSG BUFFER, ETC.
2103 016246                             1$:
2104 016246                                     ENDSRV
     016246                             L10026:
     016246  000002                             RTI
2105                                            .SBTTL  WAITF   - WAIT FOR SUBSYSTEM READY
2106                                    ;
2107                                    ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
2108                                    ;
2109                                    ;INPUTS:
2110                                    ;
2111                                    ;       R5      ADDRESS OF FIRST DEVICE REGISTER
2112                                    ;
2113                                    ;OUTPUTS:
2114                                    ;
2115                                    ;       R0      CONTENTS OF LAST TSSR READ
2116                                    ;       CARRY   SET - READY BIT SET
2117                                    ;               CLR - TIMEOUT WAITING FOR READY
2118                                    ;
2119 016250  000401                     WAITF:: BR      1$              ;NOP WHEN SUPER FIXED
2120 016252                                     BREAK                   ; DO A SUPVSR BREAK FIRST.
     016252  104422                             TRAP    C$BRK
2121 016254  012746  011000             1$:     MOV     #11000,-(SP)    ;25-APRIL-83 REV B  - 1100 MSEC TIMER
2122 016260  016500  000002             2$:     MOV     TSSR(R5),R0     ;READ THE TSSR REGISTER
2123 016264  105700                             TSTB    R0              ;TEST FOR READY BIT SET
```

WAITF - WAIT FOR SUBSYSTEM READY

```
                                        BMI     3$              ; EXIT ON STOP FLAG.
                                        DELAY   1               ; WAIT 100 USEC
016270  012727  000001                  MOV     #1,(PC)+
016274  000000                          .WORD   0
016276  013727  002116                  MOV     L$DLY,(PC)+
016302  000000                          .WORD   0
016304  005367  177772                  DEC     -6(PC)
016310  001375                          BNE     .-4
016312  005367  177756                  DEC     -22(PC)
016316  001367                          BNE     .-20
016320  005316                          DEC     (SP)            ;REDUCE DELAY COUNT
016322  001356                          BNE     2$              ;RETRY UNTIL TIMER EXPIRES
016324  000241                          CLC                     ; C = 0, CONTROLLER STILL RUNNING...
016326  000401                          BR      4$              ;...OR HUNG-UP AFTER 300 MSEC.
016330  000261                  3$:     SEC                     ; C = 1, CONTROLLER IS STOPPED.
016332  005326                  4$:     DEC     (SP)+           ;RESTORE STACK WITHOUT CHANGING CARRY BIT
016334  000207                          RTS     PC
                                        .SBTTL  CHKTSSR - CHECK TSSR FOR READY


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

016336                          CHKTSSR:
016336  004737  016250                  JSR     PC,WAITF                ;WAIT FOR READY
016342  103014                          BCC     20$                     ;BRANCH IF TIME OUT
016344  004737  016044                  JSR     PC,CHKAMB               ;TSSR AMBIGOUS?
016350  103006                          BCC     10$                     ;BR IF YES
016352  032700  100000                  BIT     #SC,R0                  ;SPECIAL CONDITION SET?
016356  001405                          BEQ     15$                     ;BR IF NO
016360  032700  074000                  BIT     #<SCE!BIE!RMR!NXM>,R0   ;ANY ERROR BITS SET?
016364  001402                          BEQ     15$                     ;BR IF NO
016366  000241                  10$:    CLC                             ;SET FAILURE
016370  000401                          BR      20$                     ;
016372  000261                  15$:    SEC                             ;SET SUCCESS
016374  000207                  20$:    RTS     PC                      ;RETURN TO CALLER
                                        .SBTTL  XNXM    - CHECK FOR NONEXISTENT MEMORY
                                ;+
                                ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
                                ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
                                ;               "C" = 0, ALL ADDRESSES OK.
                                ;
                                ;CALL:  MOV ADR1,R1
```

XNXM - CHECK FOR NONEXISTENT MEMORY

```
                                        ;       MOV ADR2,R2
                                        ;       JSR PC,NXM
                                        ;       RETURN          ;TEST "C" AND PROCEED.
                                        ;
016376  012737  016430  000004   XNXM:  MOV     #2$,@#4          ; SET BUSERR VECTOR.
016404  012737  000200  000006         MOV     #PRI04,@#6
016412  005003                         CLR     R3               ;FLAG.
016414  005711                 1$:     TST     (R1)             ;TEST THE ADDRESS(ES).
                                                                 ;IF ANY TRAP, CONTINUE AT 2$.
016416  020102                         CMP     R1,R2            ;OTHERWISE, CONTINUE HERE.
016420  001407                         BEQ     3$               ;BR IF FINISHED (NO NEXM'S).
016422  062701  000002                 ADD     #2,R1            ;SET NEXT ADDRESS...
016426  000772                         BR      1$               ;...AND CONTINUE.

016430  005103                 2$:     COM     R3               ;GOT ONE, SET FLAG...
016432  012716  016440                 MOV     #3$,(SP)
016436  000002                         RTI                      ;...AND DISMISS INTERRUPT...
016440                          3$:     CLRVEC  #4               ;...AND GIVE BACK THE VECTOR.
016440  012700  000004                 MOV     #4,R0
016444  104436                         TRAP    C$CVEC
016446  005703                         TST     R3               ;DID WE CATCH ONE ??
016450  001401                         BEQ     .+4              ;NO, "C" = 0, SKIP NEXT.
016452  000261                         SEC                      ;YES, "C" = 1, (R1) = NEXM ADDR.
016454  000207                         RTS     PC


                                                .SBTTL  TSTLOOP - CHECK ITERATION COUNT
                                        ;+
                                        ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
                                        ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
                                        ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
                                        ;
                                        ; CALL: LOOPTO  ARG
                                        ;
016456                                  TSTLOOP::
016456  005737  002170                         TST     NOITS            ; ITERATIONS INHIBITED?
016462  001006                                 BNE     1$               ; YES.
016464  005737  002204                         TST     QVP              ; NO.
016470  100403                                 BMI     1$               ;LOOPS DISALLOWED IN QUICK PASS.
016472  005337  002216                         DEC     LOOPCNT          ; BUMP LOOP COUNTER.
016476  001002                                 BNE     2$
016500  000241                          1$:    CLC                      ;LOOP DISALLOWED, OR DONE.
016502  000401                                 BR      3$
016504  000261                          2$:    SEC                      ;LOOP ENABLED.
016506  000207                          3$:    RTS     PC

                                                .SBTTL  TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
                                        ;+
                                        ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
                                        ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
                                        ; IN THE CURRENT RUN SEQUENCE.
                                        ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
                                        ;
                                        ;INPUT:
                                        ;
                                        ;       R0      POINTER TO TEST ID ASCIZ STRING
                                        ;
```

```
                                        ;OUTPUT:
                                        ;
                                        ;       R5      ADDRESS OF FIRST DEVICE REGISTER
                                        ;
                                        ;IMPLICIT OUTPUTS:
                                        ;
                                        ;       TSTCNT  UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
                                        ;
                                        ;SIDE EFFECTS:
                                        ;
                                        ;       INTERRUPT LEVEL IS RASIED TO LEVEL OF
                                        ;       THE DEVICE UNDER TEST
                                        ;
                                        ;-

016510                                  TSTSETUP::
016510  010046                                  MOV     R0,-(SP)        ;SAVE THE TEST ID MESSAGE
016512  005037  003154                          CLR     SIFLAG          ; CLEAR "SOFT INIT" FLAG
016516  005037  016756                          CLR     ERRK            ; CLEAR LOCAL ERROR COUNTER.
016522  005037  005772                          CLR     EXTA            ; CLEAR ERROR EXTENSION FLAG.
016526  105037  016144                          CLRB    INTMASK         ; CLEAR INTERRUPT MASK (CHECK ERROR)
016532  013700  002202                          MOV     UNITN,R0        ; GET THE UNIT NUMBER,
016536  006300                                  ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
016540  005737  003114                          TST     NODEV           ; DID STARTUP FIND THE DEVICE?
016544  001450                                  BEQ     4$              ; BR IF YES
016546  100010                                  BPL     3$              ; BR IF NOT IDLE
016550  052760  160000  003176                  BIS     #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
016556                                          ERRDF   1,NXR,NXRERR    ; NO DEVICE HERE -- PRINT IT
016556  104455                                  TRAP    C$ERDF
016560  000001                                  .WORD   1
016562  003740                                  .WORD   NXR
016564  005736                                  .WORD   NXRERR
016566  000407                                  BR      2$
016570  052760  160001  003176  3$:             BIS     #160001,ERTABL(R0)      ; FLAG ERROR IN THE ERROR TABLE
016576                                          ERRDF   2,NOINIT        ; DEVICE NOT IDLE
016576  104455                                  TRAP    C$ERDF
016600  000002                                  .WORD   2
016602  004335                                  .WORD   NOINIT
016604  000000                                  .WORD   0
016606  012737  177777  003112  2$:             MOV     #-1,DUFLG       ; DROP THE UNIT
016614                                          DODU    UNITN
016614  013700  002202                          MOV     UNITN,R0
016620  104451                                  TRAP    C$DODU
016622                                          DOCLN                   ; ABORT THE PASS
016622  104444                                  TRAP    C$DCLN
016624  000423                                  BR      5$

016626                                  4$:     RFLAGS  R0              ; GET THE OPERATOR FLAGS.
016626  104421                                  TRAP    C$RFLA
016630  032700  001000                          BIT     #PNT,R0         ; PRINT THE TEST NUMBERS?
016634  001412                                  BEQ     1$              ; BR IF NO
016636  011600                                  MOV     (SP),R0         ;GET THE ID MESSAGE
016640                                          PRINTF  #TNAM,R0                ;DISPLAY THE TEST ID
016640  010046                                  MOV     R0,-(SP)
016642  012746  016704                          MOV     #TNAM,-(SP)
016646  012746  000002                          MOV     #2,-(SP)
016652  010600                                  MOV     SP,R0
```

TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS

```
                016654  104417                          TRAP    C$PNTF
                016656  062706  000006                  ADD     #6,SP
     2269       016662  005237  002214          1$:     INC     TSTCNT          ; BUMP TEST COUNTER.
     2270       016666                          5$:     SETPRI  IPRI            ;PRIORITY THAT OF DEVICE
                016666  013700  002212                  MOV     IPRI,R0
                016672  104441                          TRAP    C$SPRI
     2271       016674  005726                          TST     (SP)+           ;FIX UP THE STACK
     2272       016676  013705  002206                  MOV     CSRADDR,R5      ; ADDRESS OF TSV REGISTERS ON UNIBUS
     2273       016702  000207                          RTS     PC
     2274       016704     045     123     045  TNAM:  .ASCIZ  '%S%T%A Test'
     2275                                               .EVEN
     2276                                               .SBTTL  TSTEND  - PRINT ERRORS RECEIVED
     2277                                       ;
     2278                                       ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
     2279                                       ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
     2280                                       ;
     2281       016720                          TSTEND: RFLAGS  R0
                016720  104421                          TRAP    C$RFLA
     2282       016722  030027  020000                  BIT     R0,#IER
     2283       016726  001412                          BEQ     1$              ; BR IF "IER" NOT SET.
     2284       016730                                  PRINTF  #ESUM,ERRK      ; PRINT ERROR COUNT.
                016730  013746  016756                  MOV     ERRK,-(SP)
                016734  012746  016760                  MOV     #ESUM,-(SP)
                016740  012746  000002                  MOV     #2,-(SP)
                016744  010600                          MOV     SP,R0
                016746  104417                          TRAP    C$PNTF
                016750  062706  000006                  ADD     #6,SP
     2285       016754  000207                  1$:     RTS     PC
     2286
     2287       016756  000000                  ERRK:   0                       ; LOCAL ERROR COUNT.
     2288       016760     045     101     040  ESUM:  .ASCIZ  /%A %D%A ERRORS/
     2289       016777     105     122     122  EMAXDU: .ASCIZ  /ERROR LIMIT REACHED -- DROPPING UNIT/
     2290                                               .EVEN
     2291
     2292                                               .SBTTL  INCERK  - INCREMENT LOCAL ERROR COUNT
     2293                                       ;+
     2294                                       ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
     2295                                       ;-
     2296       017044  005237  016756          INCERK: INC     ERRK            ; INCREMENT LOCAL ERROR COUNT
     2297       017050  010046                          MOV     R0,-(SP)        ; SAVE R0
     2298       017052  013700  002202                  MOV     UNITN,R0        ; GET UNIT NUMBER.
     2299       017056  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
     2300       017060  062700  003176                  ADD     #ERTABL,R0      ; R0 GETS ADDRESS OF ERROR TABLE ENTRY.
     2301       017064  005210                          INC     (R0)            ; INCREMENT THE DEVICE ERROR COUNT
     2302       017066  032710  007777                  BIT     #7777,(R0)      ; DID WE OVERFLOW THE FIELD?
     2303       017072  001001                          BNE     1$              ; BR IF NO.
     2304       017074  005310                          DEC     (R0)            ; YES -- BACK IT UP TO 7777.
     2305       017076  012600                  1$:     MOV     (SP)+,R0        ; RESTORE R0
     2306       017100  000207                          RTS     PC              ; RETURN TO CALLER.
     2307
     2308       017102  010046                  CKEMAX: MOV     R0,-(SP)        ; SAVE R0
     2309       017104  013700  002202                  MOV     UNITN,R0        ; GET UNIT NUMBER
     2310       017110  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET
     2311       017112  016000  003176                  MOV     ERTABL(R0),R0   ; GET ERROR TABLE ENTRY
     2312       017116  042700  170000                  BIC     #170000,R0      ; EXTRACT ERROR COUNT FIELD
     2313       017122  020037  002174                  CMP     R0,GERRMAX      ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
     2314       017126  103004                          BHIS    1$              ; BR IF YES
```

INCERK - INCREMENT LOCAL ERROR COUNT

```
2315 017130  023737  016756  002172          CMP     ERRK,LERRMAX    ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
2316 017136  103417                          BLO     2$              ; BR IF NO
2317 017140                          1$:     RFLAGS  R0              ; GET OPERATOR FLAGS
     017140  104421                          TRAP    C$RFLA
2318 017142  032700  000040                  BIT     #IDU,R0         ; IS DROPPING INHIBITED?
2319 017146  001013                          BNE     2$              ; BR IF YES.
2320 017150  012737  177777  003112          MOV     #-1,DUFLG       ; NO -- DROP THE UNIT
2321 017156                                  ERRDF   4,EMAXDU
     017156  104455                          TRAP    C$ERDF
     017160  000004                          .WORD   4
     017162  016777                          .WORD   EMAXDU
     017164  000000                          .WORD   0
2322 017166                                  DODU    UNITN
     017166  013700  002202                  MOV     UNITN,R0
     017172  104451                          TRAP    C$DODU
2323 017174                                  DOCLN
     017174  104444                          TRAP    C$DCLN
2324 017176  012600                  2$:     MOV     (SP)+,R0        ; RESTORE R0
2325 017200  000207                          RTS     PC              ; RETURN TO CALLER
2326                                         .SBTTL  CKDROP  - CHECK IF UNIT SHOULD BE DROPPED
2327                                 ;+
2328                                 ; CHECK IF UNIT SHOULD BE DROPPED
2329                                 ;-
2330 017202  010046                  CKDROP: MOV     R0,-(SP)
2331 017204                                  FORCERROR       1$,NOTSSR
2332 017214                                  RFLAGS  R0
     017214  104421                          TRAP    C$RFLA
2333 017216  032700  000040                  BIT     #IDU,R0
2334 017222  001010                          BNE     1$
2335 017224  011600                          MOV     (SP),R0
2336 017226  012737  177777  003112          MOV     #-1,DUFLG
2337 017234                                  DODU    UNITN
     017234  013700  002202                  MOV     UNITN,R0
     017240  104451                          TRAP    C$DODU
2338 017242                                  DOCLN                   ;ABORT THE PASS
     017242  104444                          TRAP    C$DCLN
2339 017244  012600                  1$:     MOV     (SP)+,R0
2340 017246  000207                          RTS     PC
2341
2342
2343                                         .SBTTL  CONFIG  - DETERMINE CONFIGURATION OF SYSTEM
2344                                 ;
2345                                 ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
2346                                 ;
2347 017250                          CONFIG:
2348 017250  004737  015774                  JSR     PC,SOFINIT
2349 017254  000207                          RTS     PC
2350                                         .SBTTL  KTON,KTOFF      - ENABLE/DISABLE MEMORY MANAGEMENT
2351                                 ;
2352                                 ; SUBROUTINE - ENABLE MEM MGT.
2353                                 ;
2354 017256  005737  003132          KTON:   TST     KTFLG           ; GOT KT?
2355 017262  001403                          BEQ     1$              ; NO.
2356 017264  012737  000001  177572          MOV     #1,SR0          ; YES. ENABLE KT11.
2357 017272  000207                  1$:     RTS     PC
2358
2359                                 ;
```

KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT

```
                                        ; SUBROUTINE - DISABLE MEM MGT.
                                        ;
017274  005737  003132          KTOFF:  TST     KTFLG           ; GOT KT11?
017300  001405                          BEQ     1$              ; NO.
017302  000240                          NOP
017304  000240                          NOP
017306  012737  000000  177572          MOV     #0,SR0          ; DISABLE KT.
017314  000207                  1$:     RTS     PC
                                        .SBTTL  SETMAP  - SETUP PAR6 MAPPING


                                        ;+
                                        ;
                                        ;THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
                                        ;AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
                                        ;IS RETURNED BIASED TO PAR6.
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R0      HIGH ORDER ADDRESS BITS
                                        ;       R1      LOW ORDER ADDRESS BITS
                                        ;
                                        ;OUTPUTS:
                                        ;
                                        ;       R0      OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
                                        ;       CARRY   SET IF SUCCESS
                                        ;               CLR IF ERROR
                                        ;-
017316                                  SETMAP:
017316                                          SAVREG                          ;SAVE R1-R4 UNTIL NEXT RETURN
017322  005737  003132                          TST     KTFLG                   ;SYSTEM HAVE ABOVE 28K?
017326  001433                                  BEQ     10$                     ;BR IF NO
017330  010102                                  MOV     R1,R2                   ;SAVE LOW ORDER BITS
        000006                                  .REPT   6
                                                ASR     R0                      ;CONVERT WORD ADDRESS TO 32W BLOCKS
                                                ROR     R1                      ;MAKE IT DOUBLE PRECISION
                                                .ENDR
017362  042701  000177                          BIC     #177,R1                 ;ALINE FOR LOWER 4K BOUNDARY
017366  020137  003132                          CMP     R1,KTFLG                ;HIGHER THAN EXISTING MEMORY?
017372  103011                                  BHIS    10$                     ;BR IF YES
017374  010137  172354                          MOV     R1,@#KIPAR6             ;SETUP MAPPING REGISTER PAR6
017400  042702  160000                          BIC     #160000,R2              ;SETUP DISPLACEMENT IN PAGE
017404  062702  140000                          ADD     #140000,R2              ;ADD IN PAR6 BIAS
017410  010200                                  MOV     R2,R0                   ;RETURN IN R0
017412  000261                                  SEC                             ;SET SUCCESS
017414  000401                                  BR      15$                     ;
017416  000241                          10$:    CLC                             ;SET FAILURE
017420  000207                          15$:    RTS     PC                      ;RETURN
                                        .SBTTL  FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN


                                        ;+
                                        ;  FILL MEMORY WITH A BACKGROUND PATTERN
                                        ;
                                        ; INPUTS:
                                        ;
                                        ;       R0 = BACKGROUND PATTERN
                                        ;       FREE    = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
                                        ;       KTFLG   = SET TO HIGHEST MEMORY LOCATION IF > 28K.
                                        ;
```

FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN

```
                                        ; OUTPUTS:
                                        ;
                                        ;       NONE
                                        ;-
                                        ;
017422                                  FILLMEM:
017422                                          SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
017426  004737  017274                          JSR     PC,KTOFF        ;DISABLE KT.
017432  010003                                  MOV     R0,R3           ;COPY TEST PATTERN
017434  013701  003124                          MOV     FREE,R1         ;GET FIRST FREE LOCATION
017440  013702  003126                          MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
017444  010321                          10$:    MOV     R3,(R1)+        ;STORE A BACKGROUND WORD
017446  005302                                  DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
017450  003375                                  BGT     10$             ;BR IF NO
017452  005737  003132                          TST     KTFLG           ; GOT KT?
017456  001477                                  BEQ     55$             ; NO. GET OUT.
017460  004737  017256                          JSR     PC,KTON         ; YES. ENABLE KT.
017464  005000                                  CLR     R0              ;HIGH ORDER ADDRESS START
017466  013701  003152                          MOV     PST32W,R1       ;GET >28K START ADDRESS (IN 32W BLOCKS)
        000006                                  .REPT   6
                                                CLC                     ;CLEAR C BIT
                                                ROL     R1              ;CONVERT BLOCKS TO WORDS
                                                ROL     R0              ;MAKE IT DOUBLE PRECISION
                                                .ENDR
017536  004737  017316                          JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
017542  010320                          30$:    MOV     R3,(R0)+        ;STORE TEST PATTERN IN >28K ADDRESS
017544  020027  160000                          CMP     R0,#160000      ;END OF PAR6 MAPPING AREA?
017550  103774                                  BLO     30$             ;BR IF NO
017552  162700  020000                          SUB     #20000,R0       ;BACKUP INTO PAR6 MAPPING BEGIN
017556  062737  000200  172354                  ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
017564  023737  172354  003132                  CMP     @#KIPAR6,KTFLG  ;END OF MEMORY?
017572  001427                                  BEQ     50$             ;BR IF YES
017574  005737  003144                          TST     T23A            ;11/23A?
017600  001407                                  BEQ     35$             ;NO KEEP GOING
017602  013704  177572                          MOV     SR0,R4          ;GET SR0 CONTENTS
017606  042704  177761                          BIC     #177761,R4      ;CLEAR ALL BUT PAGE NUMBER
017612  022704  000016                          CMP     #16,R4          ;SEE IF PAGE 7
017616  001415                                  BEQ     50$             ;EXIT IF THERE
017620  005737  003146                  35$:    TST     T23B            ;11/23B?
017624  001410                                  BEQ     45$             ;NO KEEP GOING
017626  023727  172354  007600                  CMP     @#KIPAR6,#7600  ;REACHED 18 BITS?
017634  103001                                  BHIS    40$             ;YES
017636  000403                                  BR      45$             ;NO KEEP GOING
017640  012737  000020  172516          40$:    MOV     #20,SR3         ;SET 22 BIT RELOCATION
017646  000137  017542                  45$:    JMP     30$             ;KEEP GOING ON ETC.
017652  004737  017274                  50$:    JSR     PC,KTOFF        ; DISABLE KT.
017656  000207                          55$:    RTS     PC
                                                .SBTTL  CMPMEM  - COMPARE MEMORY TO BACKGROUND PATTERN
                                        ;+
                                        ; COMPARE MEMORY WITH A BACKGROUND PATTERN
                                        ;
                                        ; INPUTS:
                                        ;
                                        ;       R0 = BACKGROUND PATTERN
                                        ;       FREE    = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
                                        ;       KTFLG   = SET TO HIGHEST MEMORY LOCATION IF > 28K.
                                        ;
```

CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN

```
                                        ; OUTPUTS:
                                        ;
                                        ;       CARRY   - SET IF NO ERROR
                                        ;       CARRY   - CLR IF ERROR
                                        ;
                                        ; IMPLICIT OUTPUTS:
                                        ;
                                        ;       ERRHI   - ERROR HIGH ADDRESS
                                        ;       ERRLO   - ERROR LOW ADDRESS
                                        ;       EXPD    - EXPECTED DATA
                                        ;       RECV    - RECEIVED DATA
                                        ;-
017660                                  CMPMEM:
017660                                          SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
017664  010003                                  MOV     R0,R3           ;COPY TEST PATTERN
017666  004737  017274                          JSR     PC,KTOFF        ;DISABLE KT.
017672  013701  003124                          MOV     FREE,R1         ;GET FIRST FREE LOCATION
017676  013702  003126                          MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
017702  020311                          10$:    CMP     R3,(R1)         ;FREE SPACE LOCATION EQUAL TO EXPD?
017704  001411                                  BEQ     15$             ;BR IF YES
017706  010137  002240                          MOV     R1,ERRLO        ;SAVE ADDRESS IN ERROR
017712  005037  002236                          CLR     ERRHI           ;NO HIGH ADDRESS
017716  010337  002232                          MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
017722  011137  002234                          MOV     (R1),RECV       ;SAVE RECV FOR ERROR REPORT
017726  000474                                  BR      50$             ;
017730  005721                          15$:    TST     (R1)+           ;POINT TO NEXT ADDRESS
017732  005302                                  DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
017734  003362                                  BGT     10$             ;BR IF NO
017736  005737  003132                          TST     KTFLG           ; GOT KT?
017742  001472                                  BEQ     55$             ; NO. GET OUT.
017744  004737  017256                          JSR     PC,KTON         ; YES. ENABLE KT.
017750  005000                                  CLR     R0              ;HIGH ORDER ADDRESS START
017752  013701  003152                          MOV     PST32W,R1       ;GET >28K START ADDRESS (IN 32W BLOCKS)
        000006                                  .REPT   6
                                                ROL     R1              ;CONVERT BLOCKS TO WORDS
                                                ROL     R0              ;MAKE IT DOUBLE PRECISION
                                                .ENDR
020006  042701  000177                          BIC     #177,R1         ;ALINE 4K BOUNDARY
020012  010046                                  MOV     R0,-(SP)        ;SAVE HIGH ORDER
020014  010146                                  MOV     R1,-(SP)        ;SAVE LOW ORDER
020016  004737  017316                          JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
020022  010004                                  MOV     R0,R4           ;COPY ADDRESS BIASED TO PAR6
020024  012601                                  MOV     (SP)+,R1        ;RESTORE LOW ORDER IN NON PAR6 FORMAT
020026  012600                                  MOV     (SP)+,R0        ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
020030  020314                          30$:    CMP     R3,(R4)         ;ABOVE 28K LOCATION EQUAL EXPD?
020032  001411                                  BEQ     32$             ;BR IF YES
020034  010037  002236                          MOV     R0,ERRHI        ;SAVE HIGH ORDER IN ERROR
020040  010137  002240                          MOV     R1,ERRLO        ;SAVE LOW ORDER IN ERROR
020044  010337  002232                          MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
020050  011437  002234                          MOV     (R4),RECV       ;SAVE RECV FOR ERROR REPORT
020054  000421                                  BR      50$             ;
020056  062701  000002                  32$:    ADD     #2,R1           ;UPDATE NON PAR6 ADDRESS
020062  005500                                  ADC     R0              ;MAKE IT DOUBLE PRECISION ADD
020064  062704  000002                          ADD     #2,R4           ;UPDATE PAR FORMAT ADDRESS
020070  020427  160000                          CMP     R4,#160000      ;END OF PAR6 MAPPING AREA?
020074  103755                                  BLO     30$             ;BR IF NO
020076  162704  020000                          SUB     #20000,R4       ;BACKUP INTO PAR6 MAPPING BEGIN
```

CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN

```
2531 020102  062737  000200  172354          ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
2532 020110  023737  172354  003132          CMP     @#KIPAR6,KTFLG  ;END OF MEMORY?
2533 020116  101744                          BLOS    30$             ;BR IF NO
2534 020120  004737  017274          50$:    JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
2535 020124  000241                          CLC                     ;SET FAILURE
2536 020126  000403                          BR      60$             ;
2537 020130  004737  017274          55$:    JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
2538 020134  000261                          SEC                     ;SET SUCCESS
2539 020136  000207          60$:    RTS     PC
2540                                        .SBTTL  REGSAV  - SAVE R1-R5 ON STACK
2541                                ;+
2542                                ;
2543                                ;ROUTINE TO
2544                                ;SAVE R1 THROUGH R5 ON THE STACK
2545                                ;
2546                                ;CALLING SEQUENCE:
2547                                ;
2548                                ;       JSR     R5,REGSAV
2549                                ;
2550                                ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
2551                                ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
2552                                ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
2553                                ;REGISTERS.
2554                                ;
2555                                ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
2556                                ;CALLED VIA A JSR PC INSTRUCTION
2557                                ;
2558                                ;-
2559
2560 020140                    REGSAV:
2561 020140  010446                          MOV     R4,-(SP)
2562 020142  010346                          MOV     R3,-(SP)
2563 020144  010246                          MOV     R2,-(SP)
2564 020146  010146                          MOV     R1,-(SP)
2565 020150  010546                          MOV     R5,-(SP)
2566 020152  016605  000012                  MOV     10.(SP),R5
2567 020156  004736                          JSR     PC,@(SP)+
2568 020160  012601                          MOV     (SP)+,R1
2569 020162  012602                          MOV     (SP)+,R2
2570 020164  012603                          MOV     (SP)+,R3
2571 020166  012604                          MOV     (SP)+,R4
2572 020170  012605                          MOV     (SP)+,R5
2573 020172  000207                          RTS     PC
2574                                        .SBTTL  GETPAT  - GET 8 BIT PATTERN FROM OPERATOR
2575                                ;+
2576                                ;
2577                                ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
2578                                ;
2579                                ;INPUTS:
2580                                ;
2581                                ;       NONE.
2582                                ;
2583                                ;OUTPUTS:
2584                                ;
2585                                ;       R0      OCTAL NUMBER FROM THE OPERATOR
2586                                ;
2587                                ;CALLING SEQUENCE:
```

```
                                        ;
                                        ;       JSR     PC,GETPAT
                                        ;
                                        ;-

                                        GETPAT::
                                                SAVREG                          ;SAVE THE GENERAL REGISTERS
020200                                  1$:     GMANID  DATASC,PATDAT,0,377,0,377,NO
020200  104443                                  TRAP    C$GMAN
020202  000406                                  BR      10000$
020204  020230                                  .WORD   PATDAT
020206  000022                                  .WORD   T$CODE
020210  020232                                  .WORD   DATASC
020212  000377                                  .WORD   377
020214  000000                                  .WORD   T$LOLIM
020216  000377                                  .WORD   T$HILIM
020220                                  10000$:
                                                BNCOMPLETE      1$      ;RETRY IF ERROR
020220  103367                                  BCC     1$
020222  013700  020230                          MOV     PATDAT,R0       ;DATA PATTERN FROM OPERATOR
020226  000207                                  RTS     PC              ;RETURN TO CALLER

                                        ;+
                                        ;LOCAL DATA AREA
                                        ;-

020230  000000                          PATDAT: .WORD   0               ;TEMPORARY STORAGE FOR DATA
020232      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
                                        ;
                                        ;-

                                        GETSEL::
                                                SAVREG                          ;SAVE GENERAL REGISTERS
                                                MOV     R0,R2                   ;SAVE THE MENU ADDRESS
                                        1$:     MOV     R2,R3                   ;START OF MENU STRING
                                        2$:     TST     (R3)                    ;END OF ASCII ?
                                                BEQ     3$                      ;BRANCH IF ALL LINES DISPLAYED
                                                PRINTF  @SELASC,(R3)+           ;DISPLAY THE MENU
                                                MOV     (R3)+,-(SP)
                                                MOV     @SELASC,-(SP)
                                                MOV     #2,-(SP)
                                                MOV     SP,R0
```

GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE

```
020306  104417                          TRAP    C$PNTF
020310  062706  000006                  ADD     #6,SP
2631 020314  000764                     BR      2$
2632 020316                     3$:     GMANID  MENASC,MENRES,D,-1,0,-1,NO
020316  104443                          TRAP    C$GMAN
020320  000406                          BR      10001$
020322  020476                          .WORD   MENRES
020324  000042                          .WORD   T$CODE
020326  020447                          .WORD   MENASC
020330  177777                          .WORD   -1
020332  000000                          .WORD   T$LOLIM
020334  177777                          .WORD   T$HILIM
020336                          10001$:
2633 020336                             BNCOMPLETE      1$      ;RETRY IF ERROR
020336  103352                          BCC     1$
2634 020340  013700  020476             MOV     MENRES,R0       ;GET THE OPERATOR'S REPLY
2635 020344  020001                     CMP     R0,R1           ;COMPARE TO MAXIMUM ALLOWED
2636 020346  101411                     BLOS    5$              ;BRANCH IF OK
2637 020350                             PRINTF  #MENERR         ;DISPLAY ERROR MESSAGE
020350  012746  020374                  MOV     #MENERR,-(SP)
020354  012746  000001                  MOV     #1,-(SP)
020360  010600                          MOV     SP,R0
020362  104417                          TRAP    C$PNTF
020364  062706  000004                  ADD     #4,SP
2638 020370  000735                     BR      1$              ;RETRY
2639 020372  000207                     RTS     PC              ;RETURN TO CALLER
2640 020374     045     116     045     MENERR: .ASCIZ  '%N%A *** Menu Selection Too Large ***'
2641 020442     045     116     045     SELASC: .ASCIZ  '%N%T'
2642 020447     105     156     164     MENASC: .ASCIZ  'Enter Menu Selection: '
2643                                    .EVEN
2644 020476  000000                     MENRES:         .WORD   0
2645                                    .SBTTL  CHKMAN  - CHECK MANUAL INTERVENTION LEGALITY
2646                                    ;*
2647                                    ;
2648                                    ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
2649                                    ;
2650                                    ;INPUT:
2651                                    ;
2652                                    ;       NONE.
2653                                    ;
2654                                    ;OUTPUT:
2655                                    ;
2656                                    ;       CARRY   0       MANUAL INTERVENTION NOT ALLOWED
2657                                    ;               1       MANUAL INTERVENTION IS OK
2658                                    ;
2659                                    ;SIDE EFFECTS:
2660                                    ;
2661                                    ;       A MESSAGE IS DISPLAYED WARNING THAT TEST IS
2662                                    ;       NOT EXECUTED IF MANUAL INTERVENTION IS NOT
2663                                    ;       ALLOWED.
2664                                    ;
2665                                    ;-
2666
2667 020500                      CHKMAN::
2668 020500                             SAVREG                  ;SAVE THE REGISTERS
2669 020504                             MANUAL                  ;SEE IF MANUAL INTERVENTION OK
020504  104450                          TRAP    C$MANI
```

```
2670 020506                                          BCOMPLETE 1$            ;BRANCH IF ALLOWED
     020506  103411                                 BCS     1$
2671 020510                                          PRINTF  #NOMAN          ;PRINT THE WARNING MESSAGE
     020510  012746  020534                         MOV     #NOMAN,-(SP)
     020514  012746  000001                         MOV     #1,-(SP)
     020520  010600                                 MOV     SP,R0
     020522  104417                                 TRAP    C$PNTF
     020524  062706  000004                         ADD     #4,SP
2672 020530  000241                                 CLC                     ;CLEAR CARRY FOR ERROR
2673 020532  000207                          1$:    RTS     PC              ;RETURN
2674
2675 020534      045     116     045     NOMAN:    .ASCIZ  '%N%A *** Manual Intervention not Allowed - Test Aborted ***'
2676                                                .even
2677                                                .SBTTL  ENVIRN  - SETUP FREE DIAGNOSTIC SPACE
2678                                         ;
2679                                         ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
2680                                         ;
2681 020630                                 ENVIRN: MEMORY  R0
     020630  104431                                 TRAP    C$MEM
2682 020632  010037  003124                         MOV     R0,FREE         ; GET 1ST FREE ADDRESS...
2683 020636  062737  000002  003124                 ADD     #2,FREE
2684 020644  011037  003126                         MOV     (R0),FRESIZ     ;...AND WORD COUNT.
2685 020650  162737  000004  003126                 SUB     #4,FRESIZ
2686 020656  013702  002012                         MOV     L$UNIT,R2       ; GET NUMBER OF UNITS
2687 020662  162737  000007  003126          10$:   SUB     #7,FRESIZ       ; TAKE AWAY 7 WORDS PER UNIT
2688 020670  005302                                 DEC     R2
2689 020672  001373                                 BNE     10$
2690 020674  013700  003124                         MOV     FREE,R0         ;GET FIRST FREE ADDRESS
2691 020700  063700  003126                         ADD     FRESIZ,R0       ;POINT TO LAST FREE ADDRESS
2692 020704  162700  000002                         SUB     #2,R0           ;BACKUP 1 WORD
2693 020710  010037  003130                         MOV     R0,FREEHI       ;STORE LAST FREE ADDRESS
2694 020714  000240                                 NOP                     ;**********************************************
2695 020716  012701  177520                         MOV     #BDVPCR,R1      ;GET BDV11 PCR ADDRESS
2696 020722  010102                                 MOV     R1,R2           ;COPY TO R2
2697 020724  062702  000002                         ADD     #2,R2           ;SET THE RANGE
2698 020730  004737  016376                         JSR     PC,XNXM         ;SEE IF WE HAVE ONE
2699 020734  103001                                 BCC     15$             ;OK TO SET FLAGS
2700 020736  000445                                 BR      40$             ;RETURN WITH FLAGS CLEAR
2701 020740  013701  177520                  15$:   MOV     BDVPCR,R1       ;SAVE PCR CONTENTS
2702 020744  062701  000001                         ADD     #1,R1           ;ADD ONE TO IT
2703 020750  012702  177520                         MOV     #BDVPCR,R2      ;GET BDV11 PCR ADDRESS
2704 020754  005212                                 INC     (R2)            ;TRY TO WRITE TO IT
2705 020756  013703  177520                         MOV     BDVPCR,R3       ;GET RESULTS
2706 020762  020103                                 CMP     R1,R3           ;DID IT CHANGE?
2707 020764  001017                                 BNE     20$             ;NO, MUST BE 11/23B
2708 020766  005237  003144                         INC     T23A            ;SET THE FLAG
2709 020772  042737  170000  002120                 BIC     #170000,L$HIME  ;SUPERVISOR COULD BE WRONG
2710 021000  000240                                 NOP                     ;BR 40$ FOR RELEASE
2711 021002                                         PRINTF  #M8186          ;TELL THE SYSTEM TYPE
     021002  012746  005554                         MOV     #M8186,-(SP)
     021006  012746  000001                         MOV     #1,-(SP)
     021012  010600                                 MOV     SP,R0
     021014  104417                                 TRAP    C$PNTF
     021016  062706  000004                         ADD     #4,SP
2712 021022  000413                                 BR      40$             ;RETURN
2713 021024  005237  003146                  20$:   INC     T23B            ;SET THE FLAG
2714 021030  000240                                 NOP                     ;BR 40$ FOR RELEASE
```

```
2715 021032                                          PRINTF  #M8189          ;TELL THE SYSTEM TYPE
     021032  012746  005645                          MOV     #M8189,-(SP)
     021036  012746  000001                          MOV     #1,-(SP)
     021042  010600                                  MOV     SP,R0
     021044  104417                                  TRAP    C$PNTF
     021046  062706  000004                          ADD     #4,SP
2716 021052  000207                        40$:      RTS     PC              ;RETURN
2717                                                .SBTTL  KTINIT  - SETUP KT11 MEMORY MANAGEMENT REGISTERS
2718                                        ;*
2719                                        ;
2720                                        ;ROUTINE TO INIT KT-11
2721                                        ;
2722                                        ;-
2723
2724 021054                                 KTINIT:
2725 021054  005037  003132                          CLR     KTFLG           ; INIT >28K MEMORY FLAG
2726 021060  005037  003134                          CLR     KTENABLE        ; INIT TEST >28K FLAG
2727 021064  023727  002120  001577                  CMP     L$HIME,#1577    ; GOT ENOUGH MEMORY (>28K)?
2728 021072  101444                                  BLOS    9$              ; NO.
2729 021074  013700  000004                          MOV     @#ERRVEC,R0     ; SAVE OLD ERR VEC PTR.
2730 021100  012737  021172  000004                  MOV     #2$,@#ERRVEC    ; SET ERR VEC PTR.
2731 021106  005737  177572                          TST     @#SR0           ; GOT KT11?
2732 021112  000240                                  NOP                     ;  (TRAP IF NO).
2733 021114  013737  002120  003132                  MOV     L$HIME,KTFLG    ; YES. SET KT FLAG.
2734 021122  042737  000177  003132                  BIC     #177,KTFLG      ;
2735 021130  010037  000004                          MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
2736 021134  005000                                  CLR     R0              ; R0 = AR DATA.
2737 021136  012701  172340                          MOV     #KIPAR0,R1      ; R1 = KI REGS PTR.
2738 021142  012761  077406  177740          1$:     MOV     #77406,-40(R1)  ; SET DESCRIPTOR REG.
2739 021150  010021                                  MOV     R0,(R1)+        ; SET KIPAR REG.
2740 021152  062700  000200                          ADD     #200,R0         ; BUMP AR DATA BY "4K".
2741 021156  020027  002000                          CMP     R0,#2000        ; AT "I/O"?
2742 021162  001367                                  BNE     1$              ; NO.
2743 021164  012741  177600                          MOV     #177600,-(R1)   ; YES. SET KTPAR7 FOR I/O.
2744 021170  000405                                  BR      9$
2745
2746 021172  012716  021200                  2$:     MOV     #6$,(SP)        ; SET UP RETURN
2747 021176  000002                                  RTI                     ; RTI TO NEXT LOCATION
2748
2749 021200  010037  000004                  6$:     MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
2750
2751 021204  000207                          9$:     RTS     PC
2752                                        ;*
2753                                        ;       SUBROUTINE TO SET EXTENDED FEATURES SWITCH
2754                                        ;
2755                                        ;       Requires that SOFINIT and WRTCHR have been done previous to call.
2756                                        ;
2757                                        ;
2758                                        ;INPUTS:
2759                                        ;       R5      CURRENT UNIT NUMBER
2760                                        ;OUTPUTS:
2761                                        ;       The Extended Features Switch is set.
2762                                        ;
2763                                        ;-
2764
2765 021206                                 INVERT::
2766
```

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```
2767 021206  005737  002226                    TST     EXTFEA                  ; IS SWITCH SET?
2768 021212  001020                            BNE     1$                      ; YES,EXIT STAGE RIGHT!(or the next one outa town!)
2769 021214  012737  100206  021260            MOV     #100206,CMDPKT          ; WRT SUB-SYS MEM CMD
2770 021222  012737  021270  021262            MOV     #WSMBK,CMDPKT+2         ; MSG BUF ADDR
2771 021230  012737  000006  021266            MOV     #6,CMDPKT+6             ; BYTE COUNT
2772 021236  012737  100010  021270            MOV     #100010,WSMBK           ; INVERT THE SWITCH
2773 021244  012704  021260                    MOV     #CMDPKT,R4              ; SET CMDPKT INTO R4
2774 021250  004737  010662                    JSR     PC,WRTCHR               ; DO IT
2775 021254  000207                    1$:     RTS     PC                      ; RETURN
2776
2777                                   ;       COMMAND PACKET.
2778
2779         021260                    .       .       <.+3>&177774    ;MUST BE ON MOD 4 BOUNDRY.
2780
2781 021260  000000                    CMDPKT:: 0                      ;1ST WORD IS TSO5 COMMAND.
2782 021262  000000                             0                      ;2ND WORD IS THE BUFFER LOW ADDRESS.
2783 021264  000000                             0                      ;3RD WORD IS THE BUFFER HIGH ADDRESS.
2784 021266  000000                             0                      ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
2785
2786                                   ;       WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
2787
2788 021270  000000                    WSMBK::  0                      ;1ST WORD:: SEL 0
2789 021272  000000                             0                      ;2ND WORD:: SEL 2
2790 021274  000000                             0                      ;3RD WORD:: SEL 4
2791                                            .EVEN
2792                                   ;*
2793                                   ;       SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
2794                                   ;
2795                                   ;
2796                                   ;INPUTS:
2797                                   ;OUTPUTS:
2798                                   ;       The NXMFLG is set if we can test.
2799                                   ;       The NXMLO and NXMHI addresses are setup.
2800                                   ;-
2801
2802 021276                            MEMCK::
2803
2804 021276                                    SAVREG                  ;SAVE THE REGISTERS
2805 021302  005037  003136                    CLR     NXMFLG          ;CLEAR THE FLAG
2806 021306  005037  003140                    CLR     NXMLO           ;CLEAR THE TEST ADDRESS LO
2807 021312  005037  003142                    CLR     NXMHI           ;CLEAR THE TEST ADDRESS HI
2808 021316  005737  003146                    TST     T23B            ;IS IT A 11/23B?
2809 021322  001407                            BEQ     1$              ;NO
2810 021324  023727  002120  007777            CMP     L$HIME,#7777    ; GREATER THAN 128K
2811 021332  103406                            BLO     2$              ; NO
2812 021334  004737  021452                    JSR     PC,NXMTST       ;SETUP THE ADDRESS
2813 021340  000427                            BR      13$             ;SET THE FLAG AND EXIT
2814 021342  005737  003144            1$:     TST     T23A            ;IS IT A 11/23A?
2815 021346  001413                            BEQ     4$              ;NO
2816 021350  023727  002120  005777    2$:     CMP     L$HIME,#5777    ;GREATER THAN 96K
2817 021356  101023                            BHI     14$             ;YES,23A/23B WITH 128K MEMORY
2818 021360  023727  002120  003777            CMP     L$HIME,#3777    ;GREATER THAN 64K BUT LESS THAN 92K?
2819 021366  103403                            BLO     4$              ;NO, CHECK 24K
2820 021370  004737  021452                    JSR     PC,NXMTST       ;SETUP THE ADDRESS
2821 021374  000411                            BR      13$             ;SET THE FLAG AND EXIT
2822 021376  023727  002120  001577    4$:     CMP     L$HIME,#1577    ;GREATER THAN 24K BUT LESS THAN 64K?
2823 021404  103410                            BLO     14$             ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
```

```
2824 021406  004737  021452            JSR     PC,NXMTST               ;SETUP THE ADDRESS
2825 021412  062737  000077  003142    ADD     #77,NXMHI               ;FOOL THE 11/02 & 11/03
2826 021420  005237  003136      13$:  INC     NXMFLG                  ;SET THE FLAG
2827 021424  000411              BR      15$                           ;EXIT
2828 021426  000410              14$:  BR      15$                     ;NOP FOR PRINTOUT
2829 021430                     PRINTF  #NOMEM                        ;TELL THEM & EXIT ***NO PRINT*****
     021430  012746  005460      MOV     #NOMEM,-(SP)
     021434  012746  000001      MOV     #1,-(SP)
     021440  010600              MOV     SP,R0
     021442  104417              TRAP    C$PNTF
     021444  062706  000004      ADD     #4,SP
2830 021450  000207              15$:  RTS     PC                      ;RETURN
2831
2832                             ;+
2833                             ;       SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
2834                             ;
2835                             ;OUTPUTS:NXMLO,NXMHI            ;SETUP WITH NXM ADDRESS
2836                             ;
2837                             ;-
2838
2839 021452  013701  002120      NXMTST: MOV     L$HIME,R1               ;GET TOP OF MEMORY
2840 021456  062701  000200              ADD     #200,R1                 ;MAKE IT I/O BLOCK OR OTHER NXM
2841 021462  042701  000177              BIC     #177,R1
2842 021466  010102              MOV     R1,R2                   ;RESAVE RESULTS
2843         000006              .REPT   6
2844                             ASL     R1                      ;PUT IN PLACE FOR XFER
2845                             .ENDR
2846 021504  010137  003140      MOV     R1,NXMLO                ;SAVE TEST ADDRESS LOW
2847         000012              .REPT   10.
2848                             ASR     R2                      ;PUT IN PLACE FOR XFER
2849                             .ENDR
2850 021534  042702  177700      BIC     #177700,R2              ;DON'T WANT ILA!
2851 021540  010237  003142      MOV     R2,NXMHI                ;SAVE TEST ADDRESS HIGH
2852 021544  000207              RTS     PC                      ;RETURN
2853
2854
2855
2856 021546                      ENDMOD
```

```
                                        .TITLE  TSV4 - MISCELLANEOUS SECTIONS
                                        BGNMOD  TSV4
021546                          TSV4::
021546
```

PROTECTION TABLE

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

INITIALIZE SECTION

```
                                .SBTTL  INITIALIZE SECTION

                        ;++
                        ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                        ;AT THE BEGINNING OF EACH PASS.
                        ;
                        ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
                        ;IF "CONTINUE", NOTHING IS REQUIRED.
                        ;
                        ;--
                        ;+
                        ;INSERT TEMPORARY JUMP TO ODT
                        ;-
021556                          BGNINIT
021556                  L$INIT::

021556                          SETVEC  #140,#170000,#340       ;ODT ROM ADDRESS                ;JB REV A-0
021556  012746  000340          MOV     #340,-(SP)
021562  012746  170000          MOV     #170000,-(SP)
021566  012746  000140          MOV     #140,-(SP)
021572  012746  000003          MOV     #3,-(SP)
021576  104437                  TRAP    C$SVEC
021600  062706  000010          ADD     #10,SP

021604  005037  002226  40$:     CLR     EXTFEA
021610  005037  003136          CLR     NXMFLG
021614  012737  006360  002200  MOV     #EPRT1,EPRTSW           ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
021622  005037  003154          CLR     SIFLAG                  ;CLEAR "SOFT INIT" FLAG
021626  005037  003134          CLR     KTENABLE                ;CLEAR TEST ABOVE 28K FLAG
021632  005037  002302          CLR     RAMSIZ                  ;CLEAR RAM SIZE FOR RAMERR ROUTINE
021636                          READEF  #EF.CONTINUE
021636  012700  000036          MOV     #EF.CONTINUE,R0
021642  104447                  TRAP    C$REFG
021644                          BNCOMPLETE 1$
021644  103023                  BCC     1$
021646  023737  002202  002012  CMP     UNITN,L$UNIT            ;UNIT IN RANGE?
021654  103070                  BHIS    4$                      ;BR IF NO.
021656  005737  003112          TST     DUFLG                   ;DROPPED UNIT?
021662  100472                  BMI     NXTU                    ;BR IF YES
021664  013701  002202          MOV     UNITN,R1
021670  006301                  ASL     R1
021672  005761  003176          TST     ERTABL(R1)
021676  001516                  BEQ     SETU
021700  032761  040000  003176  BIT     #BIT14,ERTABL(R1)       ;DROPPED?
021706  001060                  BNE     NXTU
021710                          EXIT    INIT                    ;DO NOTHING IF "CONTINUE".
021710  104432                  TRAP    C$EXIT
021712  000416                  .WORD   L10030-.
021714                  1$:     READEF  #EF.NEW
021714  012700  000035          MOV     #EF.NEW,R0
021720  104447                  TRAP    C$REFG
021722                          BNCOMPLETE NXTU                 ;TAKE NEXT UNIT IF NOT NEW PASS.
021722  103052                  BCC     NXTU
021724                          READEF  #EF.START
021724  012700  000040          MOV     #EF.START,R0
021730  104447                  TRAP    C$REFG
021732                          BCOMPLETE 2$
```

INITIALIZE SECTION

```
    021732  103404                          BCS     2$
64  021734                                  READEF  @EF.RESTART
    021734  012700  000037                  MOV     @EF.RESTART,R0
    021740  104447                          TRAP    C$REFG
65  021742                                  BNCOMPLETE 31$
    021742  103031                          BCC     31$
66  021744                          2$:                             ;1ST PASS, BUS-INIT...
67  021744                                  BRESET                  ;BUS RESET.
    021744  104433                          TRAP    C$RESET
68  021746  005037  002214                  CLR     TSTCNT          ;NUMBER OF TESTS RUN IN PASS
69  021752  005037  002222                  CLR     FATFLG          ;CLEAR FATAL ERROR COUNT
70  021756  005037  003144                  CLR     T23A            ;CLEAR 11/23A FLAG
71  021762  005037  003146                  CLR     T23B            ;CLEAR 11/23B FLAG
72                                  ;       MOV     #340,-(SP)
73                                  ;       MOV     #20$,-(SP)      ;RETURN TO DEBUGGER
74                                  ;       JMP     0.ODT       ;;@@ENTER THE DEBUGGER
75  021766  005037  003400                  CLR     SKIPT       ;CLEAR THE SUBTEST "SKIPPER"
76  021772                          20$:
77  021772  012737  177777  002204          MOV     #-1,QVP         ;...QUICK VERIFY...
78  022000  004737  020630                  JSR     PC,ENVIRN       ;SET ENVIRONMENT.
79  022004  004737  021054                  JSR     PC,KTINIT       ;INITIALIZE KT MEMORY MANAGEMENT
80  022010  012700  003176                  MOV     #ERTABL,R0
81  022014  005020                  30$:    CLR     (R0)+           ;CLEAR THE ERROR TABLE
82  022016  020027  003376                  CMP     R0,#ERTABE
83  022022  103774                          BLO     30$
84  022024  000404                          BR      4$
85  022026  005037  002204          31$:    CLR     QVP
86  022032  000137  022102                  JMP     PASRPT          ;GO REPORT THE STATUS
87
88  022036                          4$:
89  022036  012737  177777  002202  NEWPAS: MOV     #-1,UNITN       ;INIT UNIT NUMBER...
90  022044  005037  002220                  CLR     DEVCNT          ;CLEAR COUNT OF DEVICES RUNNING
91  022050                          NXTU:   BREAK
    022050  104422                          TRAP    C$BRK
92  022052  005237  002202                  INC     UNITN           ;...AND SET NEXT UNIT NUMBER.
93  022056  023737  002202  002012          CMP     UNITN,L$UNIT
94  022064  103423                          BLO     SETU
95  022066  012737  177777  003112          MOV     #-1,DUFLG
96  022074  000401                          BR      11$
97  022076                                  DOCLN                   ;ABORT, NO MORE UNITS.
    022076  104444                          TRAP    C$DCLN
98  022100  000240                  11$:    NOP
99  022102                          PASRPT:
100 022102  023727  002012  000001          CMP     L$UNIT,#1       ;HOW MANY UNITS SELECTED?
101 022110  101752                          BLOS    NEWPAS          ;BR IF ONLY 1
102 022112  005737  002220                  TST     DEVCNT          ;ARE ANY STILL RUNNING?
103 022116  001747                          BEQ     NEWPAS          ;BR IF NO
104 022120                                  RFLAGS  R0
    022120  104421                          TRAP    C$RFLA
105 022122  032700  000100                  BIT     #ISR,R0         ;SHOULD WE PRINT STATISTICS
106 022126  001343                          BNE     NEWPAS          ;BR IF NO
107
108 022130                                  DORPT
    022130  104424                          TRAP    C$DRPT
109 022132  000741                          BR      NEWPAS
110 022134                          10$:
111
```

INITIALIZE SECTION

```
112 022134                          SETU:   GPHARD  UNITN,R0              ;GET UNIT N P-TABLE POINTER.
    022134  013700  002202                  MOV     UNITN,R0
    022140  104442                          TRAP    C$GPHRD
113 022142                                  BNCOMPLETE NXTU              ;BR IF UNIT NOT AVAILABLE.
    022142  103342                          BCC     NXTU
114 022144  005037  003112                  CLR     DUFLG                ;CLEAR "DROPPED" FLAG.
115 022150  005237  002220                  INC     DEVCNT
116 022154  012001                          MOV     (R0)+,R1             ;GET 1ST REGISTER ADDRESS.
117 022156  010137  002206                  MOV     R1,CSRADDR           ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
118
119 022162  012001                          MOV     (R0)+,R1             ;GET VECTOR ADDRESS.
120                                         ;MOV    (R0),R2              ;GET INTERRUPT PRIORITY
121                                         ;MOV    R2,IPRI              ;SET INTERRUPT PRIORITY.
122 022164  010137  002210                  MOV     R1,IVEC              ;SET INTERRUPT VECTOR POINTER...
123 022170  012721  016216                  MOV     #INTR,(R1)+          ;...VECTOR...
124 022174  013721  002212                  MOV     IPRI,(R1)+           ;...AND PRIORITY.
125
126 022200                          1$:
127                                 ;       TST     QVP             ;1ST PASS ??
128                                 ;       BEQ     5$              ;NO, SKIP THE PASS 1 STUFF.
129
130                                 ;
131                                 ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
132                                 ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
133                                 ;
134 022200  013701  002202                  MOV     UNITN,R1
135 022204  006301                          ASL     R1
136 022206  052761  100000  003176          BIS     #BIT15,ERTABL(R1)       ;SAY DEVICE RUNNING
137 022214  005037  005772                  CLR     EXTA                    ;CLEAR ERROR EXTENSION FLAG.
138 022220  023727  002012  000001          CMP     L$UNIT,#1               ;ARE WE TESTING MULTIPLE UNITS?
139 022226  101416                          BLOS    10$                     ;BR IF NO.
140 022230                                  RFLAGS  R0                      ;YES -- GET OPERATOR FLAGS.
    022230  104421                          TRAP    C$RFLA
141 022232  032700  001000                  BIT     #PNT,R0                 ;SHOULD WE PRINT UNIT #?
142 022236  001412                          BEQ     10$                     ;BR IF NOT.
143 022240                                  PRINTF  #PUNIT,UNITN            ;PRINT THE UNIT #
    022240  013746  002202                  MOV     UNITN,-(SP)
    022244  012746  022332                  MOV     #PUNIT,-(SP)
    022250  012746  000002                  MOV     #2,-(SP)
    022254  010600                          MOV     SP,R0
    022256  104417                          TRAP    C$PNTF
    022260  062706  000006                  ADD     #6,SP
144 022264                          10$:
145 022264  005037  003114                  CLR     NODEV
146 022270  013701  002206                  MOV     CSRADDR,R1              ;ADDRESS OF FIRST REGISTER
147 022274  010102                          MOV     R1,R2                   ;START OF REGISTERS
148 022276  062702  000002                  ADD     #TSSR,R2                ;ADDRESS OF TSSR REGISTER
149 022302  004737  016376                  JSR     PC,XNXM                 ;TEST BOTH CONTROLLER REGISTERS...
150 022306  103005                          BCC     2$                      ;...AND BR IF ALL OK.
151 022310  010137  003114                  MOV     R1,NODEV                ;FLAG DEVICE AS NON-EXISTENT
152 022314  012737  177777  003112          MOV     #-1,DUFLG               ;DROP THIS UNIT.
153 022322                          2$:
154                                 ;
155                                 ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
156                                 ;
157 022322                          5$:     SETPRI  #PRI00          ;ENABLE INTERRUPTS.
    022322  012700  000000                  MOV     #PRI00,R0
```

INITIALIZE SECTION

```
        022326  104441                          TRAP    C$SPRI
    158 022330                                  ENDINIT
        022330                          L10030:
        022330  104411                          TRAP    C$INIT
    159
    160 022332      045     116     045 PUNIT:  .ASCIZ  /%N%N%A****** TESTING UNIT %D2%A ******/
    161                                         .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.
                                ;--
022400                                  BGNAU
022400                          L$AU::
022400  010001                        MOV     R0,R1                   ; GET UNIT TO BE ADDED (R0)
022402  006301                        ASL     R1                      ; MAKE IT A WORD INDEX
022404  052761  100000  003176        BIS     #100000,ERTABL(R1)      ; SET THE "ACTIVE" BIT
022412  042761  040000  003176        BIC     #40000,ERTABL(R1)       ; CLEAR THE "DROPPED" BIT
022420                                  PRINTF  #1$,R0
022420  010046                        MOV     R0,-(SP)
022422  012746  022446                MOV     #1$,-(SP)
022426  012746  000002                MOV     #2,-(SP)
022432  010600                        MOV     SP,R0
022434  104417                        TRAP    C$PNTF
022436  062706  000006                ADD     #6,SP
022442                                  EXIT    AU
022442  000167                        .WORD   J$JMP
022444  000026                        .WORD   L10031-2-.
022446     045     116     045  1$:     .ASCIZ  /%N%A  UNIT %D%A ADDED/
                                        .EVEN

                                        ENDAU           ; UNUSED.
022474                          L10031:
022474  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.

022476                                  BGNDU
022476                          L$DU::
022476  012737  177777  003112        MOV     #-1,DUFLG
022504  010001                        MOV     R0,R1
022506  006301                        ASL     R1
022510  052761  140000  003176        BIS     #140000,ERTABL(R1)      ; SAY DROPPED
022516  000240  000240  000240        240,240,240                     ; ?????????
022524                                  PRINTF  #1$,R0
022524  010046                        MOV     R0,-(SP)
022526  012746  022552                MOV     #1$,-(SP)
022532  012746  000002                MOV     #2,-(SP)
022536  010600                        MOV     SP,R0
022540  104417                        TRAP    C$PNTF
022542  062706  000006                ADD     #6,SP
022546                                  EXIT    DU
022546  000167                        .WORD   J$JMP
022550  000030                        .WORD   L10032-2-.
```

ADD AND DROP UNITS SECTIONS

```
                045     116     045 1$:     .ASCIZ  /%N%A  UNIT %D%A DROPPED/
                                             .EVEN
                                             ENDDU
                                     L10032:
                                             TRAP    C$DU
                                     ;++
                                     ; AUTO-DROP CODE SECTION.
                                     ;--
                                             BGNAUTO
                                     L$AUTO::
                013705  002206               MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
                012703  000550               MOV     #360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
                004737  016250       10$:    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
                103420                       BCS     20$                     ;LEAVE WHEN SSR IS SET
                                             DELAY   250.                    ;WAIT FOR .25 SECONDS
                012727  000372               MOV     #250.,(PC)+
                000000                       .WORD   0
                013727  002116               MOV     L$DLY,(PC)+
                000000                       .WORD   0
                005367  177772               DEC     -6(PC)
                001375                       BNE     .-4
                005367  177756               DEC     -22(PC)
                001367                       BNE     .-20
                005303                       DEC     R3                      ;BUMP COUNTER DOWN
                001357                       BNE     10$                     ;KEEP GOING
                004737  017202               JSR     PC,CKDROP               ;TRY AND DROP UNIT
                                     20$:
                                             ENDAUTO                 ; UNUSED.
                                     L10033:
                                             TRAP    C$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.
                                ;--
                                        BGNCLN
022664                          L$CLEAN::
022664  013705  002206                  MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
022670  005737  003112                  TST     DUFLG           ;"DROPPED" FLAG IS SET ON...
022674  100405                          BMI     1$              ;...AND GROSS CONTROLLER FAULT...
                                                                ;...DON'T TRY TO XCT CLEANUP CODE.

022676  012765  000000  000002          MOV     #0,TSSR(R5)     ;DO SOFT INIT
022704  004737  016250                  JSR     PC,WAITF
022710                          1$:
022710                          2$:     ENDCLN
022710                          L10034:
022710  104412                          TRAP    C$CLEAN
                                ;++
                                ; THE REPORT CODING SECTION CONTAINS THE
                                ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;--
                                        BGNRPT
022712                          L$RPT::
022712                                  PRINTS  #DEVSUM
022712  012746  023154                  MOV     #DEVSUM,-(SP)
022716  012746  000001                  MOV     #1,-(SP)
022722  010600                          MOV     SP,R0
022724  104416                          TRAP    C$PNTS
022726  062706  000004                  ADD     #4,SP
022732  010246                          MOV     R2,-(SP)
022734  010346                          MOV     R3,-(SP)
022736  010446                          MOV     R4,-(SP)
022740  012704  003176                  MOV     #ERTABL,R4      ; GET START OF ERROR TABLE.
022744  005003                          CLR     R3              ; CLEAR UNIT NUMBER
022746  011402                  1$:     MOV     (R4),R2         ; GET ERROR TABLE ENTRY & TEST IT.
022750  001467                          BEQ     4$              ; ZERO IF UNIT NOT RUN
022752  100066                          BPL     4$
022754  032702  040000                  BIT     #BIT14,R2       ; WAS UNIT DROPPED?
022760  001015                          BNE     2$              ; BR IF YES
022762  042702  170000                  BIC     #^C7777,R2      ; GET ERROR COUNT FIELD
022766                                  PRINTS  #DEVONL,R3,R2   ; PRINT
022766  010246                          MOV     R2,-(SP)
022770  010346                          MOV     R3,-(SP)
022772  012746  023211                  MOV     #DEVONL,-(SP)
022776  012746  000003                  MOV     #3,-(SP)
023002  010600                          MOV     SP,R0
023004  104416                          TRAP    C$PNTS
023006  062706  000010                  ADD     #10,SP
023012  000446                          BR      4$
023014  020227  160000          2$:     CMP     R2,#160000      ; WAS UNIT NON-EXISTENT?
023020  001012                          BNE     3$              ; BR IF NO
023022                                  PRINTS  #DEVNXR,R3
023022  010346                          MOV     R3,-(SP)
023024  012746  023261                  MOV     #DEVNXR,-(SP)
```

CLEAN-UP AND REPORT CODING SECTIONS

```
                023030  012746  000002                  MOV     #2,-(SP)
                023034  010600                          MOV     SP,R0
                023036  104416                          TRAP    C$PNTS
                023040  062706  000006                  ADD     #6,SP
            257 023044  000431                          BR      4$
            258 023046  020227  160001          3$:     CMP     R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
            259 023052  001012                          BNE     30$             ; BR IF NO.
            260 023054                                  PRINTS  #DEVNRD,R3
                023054  010346                          MOV     R3,-(SP)
                023056  012746  023343                  MOV     #DEVNRD,-(SP)
                023062  012746  000002                  MOV     #2,-(SP)
                023066  010600                          MOV     SP,R0
                023070  104416                          TRAP    C$PNTS
                023072  062706  000006                  ADD     #6,SP
            261 023076  000414                          BR      4$
            262 023100  042702  170000          30$:    BIC     #^C7777,R2
            263 023104                                  PRINTS  #DEVDRO,R3,R2
                023104  010246                          MOV     R2,-(SP)
                023106  010346                          MOV     R3,-(SP)
                023110  012746  023424                  MOV     #DEVDRO,-(SP)
                023114  012746  000003                  MOV     #3,-(SP)
                023120  010600                          MOV     SP,R0
                023122  104416                          TRAP    C$PNTS
                023124  062706  000010                  ADD     #10,SP
            264 023130  062704  000002          4$:     ADD     #2,R4
            265 023134  005203                          INC     R3
            266 023136  020427  003376                  CMP     R4,#ERTABE
            267 023142  103701                          BLO     1$
            268 023144  012604                          MOV     (SP)+,R4
            269 023146  012603                          MOV     (SP)+,R3
            270 023150  012602                          MOV     (SP)+,R2
            271 023152                                  ENDRPT          ; UNUSED.
                023152                          L10035:
                023152  104425                          TRAP    C$RPT
            272
            273
            274 023154      045     116     045 DEVSUM: .ASCIZ  /%N%ADEVICE STATUS SUMMARY:%N/
            275 023211      045     101     040 DEVONL: .ASCIZ  /%A  UNIT %D3%A   ONLINE,  ERRORS = %D%N/
            276 023261      045     101     040 DEVNXR: .ASCIZ  /%A  UNIT %D3%A   DROPPED, NON-EXISTENT REGISTER%N/
            277 023343      045     101     040 DEVNRD: .ASCIZ  /%A  UNIT %D3%A   DROPPED, NOT READY AT STARTUP%N/
            278 023424      045     101     040 DEVDRO: .ASCIZ  /%A  UNIT %D3%A   DROPPED, ERRORS = %D%N/
            279                                         .EVEN
            280
            281 023474                                  ENDMOD
            282
            283
```

```
                                        .TITLE  TSV5 - HARDWARE TESTS
                                        BGNMOD  TSV5
023474                          TSV5::
023474
```

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
                                ;--


023474                                  BGNTST
023474                                                                          T1::
023474  012700  024132                  MOV     @TST13ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
023500  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
023504  012737  000012  002216          MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
023512                          T13LOOP:
023512  004737  024406                  JSR     PC,T13REST              ;SET PACKET TO START-UP VALUES

023516  012703  002764                  MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
023522  012704  024070                  MOV     #T13PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
023526  012764  000010  000006          MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
023534                          5$:
023534  004737  015774                  JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
023540  103405                          BCS     10$                     ;BR IF SOFT INIT OKAY
023542  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
023544                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
023544  104455                                                                  TRAP    C$ERDF
023546  000144                                                                  .WORD   100
023550  003652                                                                  .WORD   SFIERR
023552  012034                                                                  .WORD   SFIMSG

                                ;Do WRITE CHARACTERISTICS command.
023554  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
023560  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
023564  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
023570                                  FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
023604  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
023606  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
023610                                  NEXT.ERRNO
023610                          12$:    ERRDF   ERRNO,T13SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
023610  104455                                                                  TRAP    C$ERDF
023612  000145                                                                  .WORD   101
023614  024317                                                                  .WORD   T13SSR
023616  012046                                                                  .WORD   PKTSSR
023620  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
023624                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

```
   023624  104406                                                                              TRAP    C$CLP1
   023626  016501  000002                 MOV     TSSR(R5),R1            ;GET THE CONTENTS OF TSSR
   023632  012702  000200                 MOV     #SSR,R2                ;EXPECTED CONTENTS OF TSSR
   023636  032701  000100                 BIT     #OFL,R1                ;IS OFF-LINE BIT SET ?
   023642  001402                         BEQ     25$                    ;BRANCH IF NOT OFF-LINE
   023644  052702  000100                 BIS     #OFL,R2                ;SET OFF-LINE IN EXPECTED DATA

                                  ;If the NBA bit in the TSSR register is NOT=0 then Print Error.
   023650                         25$:
   023650                                 FORCERROR       27$                    ;88D
   023664  020201                         CMP     R2,R1                  ;DOES EXPECTED MATCH RECEIVED ?
   023666  001404                         BEQ     30$                    ;OKAY IF MATCH
   023670                                 NEXT.ERRNO
   023670                         27$:    ERRHRD  ERRNO,T13NBA,PKTSSR    ;NBA NOT ZERO
   023670  104456                                                                              TRAP    C$ERHRD
   023672  000146                                                                              .WORD   102
   023674  024244                                                                              .WORD   T13NBA
   023676  012046                                                                              .WORD   PKTSSR
   023700                         30$:    CKLOOP                         ;LOOP ON ERROR ?
   023700  104406                                                                              TRAP    C$CLP1


                                  ;Write to TSSR register to soft initialize the controller
   023702                         40$:
   023702  004737  015774                 JSR     PC,SOFINIT             ;WRITE TO TSSR TO SOFT INITIALIZE
   023706                                 FORCERROR       42$                    ;88D
   023722  103405                         BCS     50$                    ;BR IF SOFT INIT OKAY
   023724  010001                         MOV     R0,R1                  ;SAVE CONTENTS OF TSSR
   023726                                 NEXT.ERRNO
   023726                         42$:    ERRDF   ERRNO,SFIERR,SFIMSG    ;DEVICE FATAL DURING INIT
   023726  104455                                                                              TRAP    C$ERDF
   023730  000147                                                                              .WORD   103
   023732  003652                                                                              .WORD   SFIERR
   023734  012034                                                                              .WORD   SFIMSG


                                  ;If controller RAM 310-377 NOT=0 then Print Error
   023736  012704  000310         50$:    MOV     #310,R4                ;START WITH LOC 310
   023742  005002                         CLR     R2                     ;MEMORY EXPECTED SHOULD BE 000000
   023744  105065  000000                 CLRB    TSDB(R5)               ;SET MAINTENANCE MODE
   023750  004737  016336                 JSR     PC,CHKTSSR             ;WAIT FOR SSR READY
   023754  010465  000000         60$:    MOV     R4,TSDB(R5)            ;SELECT RAM ADDRESS
   023760  004737  016336                 JSR     PC,CHKTSSR             ;WAIT FOR SSR READY
   023764  116501  000000                 MOVB    TSBA(R5),R1            ;READ LOC CONTENTS
   023770                                 FORCERROR       62$,NOTSSR             ;88D
   024000  120102                         CMPB    R1,R2                  ;CHECK MEMORY FOR 000000
   024002  001406                         BEQ     70$                    ;BRANCH IF DATA OKAY
   024004                                 NEXT.ERRNO
   024004                         62$:    ERRDF   ERRNO,T13MEM,RAMEXP    ;MEMORY NOT ZERO AFTER INIT.
   024004  104455                                                                              TRAP    C$ERDF
   024006  000150                                                                              .WORD   104
   024010  024205                                                                              .WORD   T13MEM
   024012  015530                                                                              .WORD   RAMEXP
   024014  005237  002222                 INC     FATFLG                 ;SET THE FATAL ERROR FLAG
   024020                         70$:    CKLOOP
   024020  104406                                                                              TRAP    C$CLP1
   024022                                 ESCAPE  TST                    ;EXIT ON FATAL ERROR
   024022  104410                                                                              TRAP    C$ESCAPE
   024024  000430                                                                              .WORD   L10036-.
```

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

```
                                82$:    INC     R4                      ;LOOK AT NEXT RAM LOC.
                                        CMP     R4,#400                 ;AT TOP OF RAM ADDRESS SPACE
                                        BNE     60$                     ;BRANCH TILL ALL MEMORY TESTED

                                        TST     FATFLG                  ;ANY FATAL ERRORS ?
                                        BEQ     160$                    ;BRANCH IF NOT
                                        JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
                                160$:   JSR     PC,TSTLOOP              ;DONE ALL ITERATIONS?
                                        BCC     165$                    ;BR IF YES
                                        JMP     T13LOOP                 ;LOOP UNTIL ITERATION COUNT DONE
                                165$:
                                        EXIT    TST
024062  104432                                                                  TRAP    C$EXIT
024064  000370                                                                  .WORD   L10036-.

                                ;+
                                ;LOCAL STORAGE FOR THIS TEST
                                ;-

024070                                  .=<.+10>&177770
024070                          T13PACKET:
024070  100004                          .WORD   100004                  ;COMMAND PACKET FOR TEST
024072  024100                          .WORD   T13DATA                 ;WRITE CHARACTERISTICS COMMAND, WITH ACK
024074  000000                          .WORD   0                       ;ADDRESS OF CHARACTERISTICS BLOCK
024076  000010                          .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE

024100                          T13DATA:                                ;CHARACTERISTICS DATA BLOCK
024100  024112                          .WORD   T13BFR                  ;ADDRESS OF MESSAGE BUFFER
024102  000000                          .WORD   0
024104  000016                          .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
024106  000000  000000                  .WORD   0,0

024112                          T13BFR: .BLKW   8.                      ;MESSAGE BUFFER
                                ;LOCAL TEXT MESSAGES FOR TEST
                                ;-

024132      111     156     151 TST13ID:        .ASCIZ  'Initialization After WRITE CHARACTERISTICS'
024205      111     156     143 T13MEM: .ASCIZ  'Incorrect RAM Data After Init'
                                        .EVEN
024244      127     122     111 T13NBA: .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
024317      103     157     156 T13SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'

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

                                .EVEN

024406                          T13REST:
024406                                  SAVREG                          ;SAVE THE REGISTERS
024412  012701  024070                  MOV     #T13PACKET,R1           ;START OF THE PACKET
```

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

```
176 024416  012721  100004                  MOV     #100004,(R1)+   ;WRITE CHARACTERISTICS WITH ACK
177 024422  012721  024100                  MOV     #T13DATA,(R1)+  ;ADDRESS OF CHAR DATA BLOCK
178 024426  005021                          CLR     (R1)+           ;EXTENDED ADDRESS
179 024430  012721  000010                  MOV     #8.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
180 024434  012721  024112                  MOV     #T13BFR,(R1)+   ;ADDRESS OF MESSAGE BUFFER
181 024440  005021                          CLR     (R1)+
182 024442  012721  000016                  MOV     #14.,(R1)+      ;LENGTH OF MESSAGE BUFFER
183 024446  005021                          CLR     (R1)+
184 024450  005011                          CLR     (R1)
185 024452  000207                          RTS     PC              ;RETURN
186 024454                                  ENDTST
    024454                                                                      L10036:
    024454  104401                                                                     TRAP    C$ETST
187
188                                         .SBTTL  TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND
189                                 ;+
190                                 ;
191                                 ;THIS TEST VERIFIES THAT THE WRITE SUBSYSTEM MEMORY COMMAND WITH A
192                                 ;BSELO SELECT CODE OF 0 (NO-OP) EXECUTES CORRECTLY. IT ALSO
193                                 ;VERIFIES THAT A WRITE SUBSYSTEM MEMORY COMMAND WITH A NON-ZERO
194                                 ;MODE FIELD IS REJECTED. THE TEST FURTHER VERIFIES MICROPROGRAM
195                                 ;COMMAND DECODING AND HANDLING SEQUENCES.
196                                 ;
197                                 ;-
198 024456                                  BGNTST
    024456                                                                      T2::
203 024456  012700  026271                  MOV     #TST14ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
204 024462  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
205 024466  012737  000024  002216          MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
206 024474                          T14LOOP:
207 024474                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
    024474                                                                              T2.1:
    024474  104402                                                                     TRAP    C$BSUB
208 024476  004737  026336                  JSR     PC,T14REST              ;SET PACKET TO INITIAL VALUES
209 024502  004737  026374                  JSR     PC,T14RST               ;SET PACKET TO INITIAL VALUES
210 024506                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
    024506  012700  000000                                                          MOV     #PRI00,R0
    024512  104441                                                                  TRAP    C$SPRI
211 024514                          5$:
212 024514                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
    024514  104404                                                                 TRAP    C$BSEG
213 024516  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
214 024522  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
218 024524  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
219 024526                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
    024526  104455                                                                 TRAP    C$ERDF
    024530  000311                                                                 .WORD   201
    024532  003652                                                                 .WORD   SFIERR
    024534  012034                                                                 .WORD   SFIMSG
220 024536                          10$:
221 024536  012704  025630                  MOV     #T14PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
222 024542  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
223 024546  103405                          BCS     11$                     ;BR, IF COMMAND ISSUED OK
227 024550  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
228 024552                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
    024552  104456                                                                 TRAP    C$ERHRD
    024554  000312                                                                 .WORD   202
```

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
                024556  005056                                                          .WORD   WRTMSG
                024560  012034                                                          .WORD   SFIMSG
    229         024562                          11$:
    230         024562  005037  002222                  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
    231         024566  005037  002224                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
    232         024572  012704  025210                  MOV     #T14PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
    233         024576  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
    234         024602  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
    235         024606  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
    236         024610  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
    240         024612                                  ERRDF   ERRNO,T14SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                024612  104455                                                                  TRAP    C$ERDF
                024614  000313                                                                  .WORD   203
                024616  026023                                                                  .WORD   T14SSR
                024620  012046                                                                  .WORD   PKTSSR
    241         024622  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
    242         024626                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                024626  104406                                                                  TRAP    C$CLP1
    243         024630                                  ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
                024630  104410                                                                  TRAP    C$ESCAPE
                024632  000074                                                                  .WORD   10000$-.
    244         024634  005737  002224                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
    245         024640  001004                          BNE     22$                     ;BRANCH IF YES
    249         024642                                  ERRHRD  ERRNO,T14NINT,PKTSSR
                024642  104456                                                                  TRAP    C$ERHRD
                024644  000314                                                                  .WORD   204
                024646  026113                                                                  .WORD   T14NINT
                024650  012046                                                                  .WORD   PKTSSR
    250         024652  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
    251         024656  012702  000200                  MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
    252         024662  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
    253         024666  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
    254         024670  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
    255         024674  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
    256         024676  001404                          BEQ     30$                     ;OKAY IF MATCH
    260         024700                                  ERRHRD  ERRNO,T14NBA,PKTSSR     ;NBA NOT ZERO
                024700  104456                                                                  TRAP    C$ERHRD
                024702  000315                                                                  .WORD   205
                024704  025652                                                                  .WORD   T14NBA
                024706  012046                                                                  .WORD   PKTSSR
    261         024710                          30$:
    262         024710  004737  011114          35$:    JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
    263         024714  103405                          BCS     59$                     ;RAM OK GO ON
    267         024716                                  ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
                024716  104456                                                                  TRAP    C$ERHRD
                024720  000316                                                                  .WORD   206
                024722  004745                                                                  .WORD   PKTRAM
                024724  015510                                                                  .WORD   RAMERR
    268         024726                                  ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                024726                                                                  10000$:
                024726  104405                                                                  TRAP    C$ESEG
    269
    270
    271         024730                          59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                024730                                                                  L10040:
                024730  104403                                                                  TRAP    C$ESUB
    272
```

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
273 024732  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
274 024736  001402                          BEQ     60$                     ;BRANCH IF NOT
275 024740  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
276 024744                          60$:
277                                 ;+
278                                 ;
279                                 ;TEST 2, SUBTEST 2
280                                 ;
281                                 ;CHECK THAT NON-ZERO MODE BITS BEING SET CAUSES
282                                 ;WRITE SUBSYSTEM MEMORY COMMAND TO BE REJECTED
283                                 ;
284                                 ;-
285
286 024744                          BGNSUB                                  ;////////////// BEGIN SUBTEST ////////////
    024744                                                                                  T2.2:
    024744  104402                                                                         TRAP    C$BSUB
287
288 024746                          SETPRI  #PRI00                          ;LOWER PRIORITY TO ALLOW INTERRUPTS
    024746  012700  000000                                                                  MOV     #PRI00,R0
    024752  104441                                                                         TRAP    C$SPRI
289 024754  012704  025630          5$:     MOV     #T14PK2,R4              ;GET THE ADDRESS OF COMMAND PACKET
290 024760  004737  026336                  JSR     PC,T14REST              ;RESTORE PACKET TO STARTING VALUES
291 024764  004737  026374                  JSR     PC,T14RST               ;RESTORE PACKET TO STARTING VALUES
292 024770                          BGNSEG                                  ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
    024770  104404                                                                         TRAP    C$BSEG
293
294 024772  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
295 024776  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
299 025000  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
300 025002                          ERRDF   ERRNO,SFIERR,SFIMSG             ;DEVICE FATAL ERROR DURING INIT
    025002  104455                                                                         TRAP    C$ERDF
    025004  000317                                                                         .WORD   207
    025006  003652                                                                         .WORD   SFIERR
    025010  012034                                                                         .WORD   SFIMSG
301 025012                          10$:
302 025012  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
303 025016  103405                          BCS     11$                     ;BR, IF COMMAND ISSUED OK
307 025020  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
308 025022                          ERRHRD  ERRNO,WRTMSG,SFIMSG             ;WRITE SUBSYSTEM MEMORY FAILED
    025022  104456                                                                         TRAP    C$ERHRD
    025024  000320                                                                         .WORD   208
    025026  005056                                                                         .WORD   WRTMSG
    025030  012034                                                                         .WORD   SFIMSG
309 025032                          11$:
310 025032  005037  002224                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
311 025036  012704  025210                  MOV     #T14PACKET,R4           ;SET-UP WITH WRT SUBSYS MEM PACKET
312 025042  052714  007000                  BIS     #007000,(R4)            ;NON-ZERO COMMAND MODE BITS
313 025046  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
314 025052  004737  016250                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
315 025056  103405                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
316 025060  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
320 025062                          ERRDF   ERRNO,T14SSR,PKTSSR             ;DEVICE FATAL SSR FAILED TO SET
    025062  104455                                                                         TRAP    C$ERDF
    025064  000321                                                                         .WORD   209
    025066  026023                                                                         .WORD   T14SSR
    025070  012046                                                                         .WORD   PKTSSR
321 025072                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
     025072  104406                                                          TRAP    C$CLP1
322  025074                          ESCAPE  SEG             ;BY-PASS CHECKS IF FATAL ERROR
     025074  104410                                                          TRAP    C$ESCAPE
     025076  000074                                                          .WORD   10000$-.
323  025100  005737  002224          TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
324  025104  001004                  BNE     22$             ;BRANCH IF YES
328  025106                          ERRHRD  ERRNO,T14NINT,PKTSSR
     025106  104456                                                          TRAP    C$ERHRD
     025110  000322                                                          .WORD   210
     025112  026113                                                          .WORD   T14NINT
     025114  012046                                                          .WORD   PKTSSR
329  025116  016501  000002  22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
330  025122  012702  100206          MOV     #SC!SSR!TSREJ,R2 ;EXPECTED CONTENTS OF TSSR
331  025126  032701  000100          BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
332  025132  001402                  BEQ     25$             ;BRANCH IF NOT OFF-LINE
333  025134  052702  000100          BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
334  025140  020201          25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
335  025142  001404                  BEQ     30$             ;OKAY IF MATCH
339  025144                          ERRHRD  ERRNO,T142REJ,PKTSSR ;COMMAND NOT REJECTED
     025144  104456                                                          TRAP    C$ERHRD
     025146  000323                                                          .WORD   211
     025150  025726                                                          .WORD   T142REJ
     025152  012046                                                          .WORD   PKTSSR
340  025154                  30$:
341  025154  004737  011114  35$:    JSR     PC,CKRAM        ;CHECK RAM TO MEMORY
342  025160  103405                  BCS     59$             ;RAM OK GO ON
346  025162                          ERRHRD  ERRNO,PKTRAM,RAMERR ;THEY DON'T MATCH
     025162  104456                                                          TRAP    C$ERHRD
     025164  000324                                                          .WORD   212
     025166  004745                                                          .WORD   PKTRAM
     025170  015510                                                          .WORD   RAMERR
347  025172                          ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     025172                                                          10000$:
     025172  104405                                                          TRAP    C$ESEG
348
349  025174                  59$:    ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     025174                                                          L10041:
     025174  104403                                                          TRAP    C$ESUB
350  025176                          EXIT    TST             ;ALL DONE THIS TEST
     025176  104432                                                          TRAP    C$EXIT
     025200  001246                                                          .WORD   L10037-.
351
352                          ;+
353                          ;LOCAL STORAGE FOR THIS TEST
354                          ;-
355
357          025210                  .=<.+10>&177770
359  025210                  T14PACKET:
360  025210  100206                  .WORD   100206          ;COMMAND PACKET FOR TEST
361  025212  025220                  .WORD   T14DATA         ;WRITE SUBSYSTEM MEM. COMMAND, WITH IE, ACK
362  025214  000000                  .WORD   0               ;ADDRESS OF CHARACTERISTICS BLOCK
363  025216  000006                  .WORD   6.              ;STARTING VALUE OF BLOCK SIZE
364  025220                  T14DATA:                        ;CHARACTERISTICS DATA BLOCK
365  025220     000          T14BS0: .BYTE   0               ;BSEL0 BYTE
366  025221     000          T14BS1: .BYTE   0               ;BSEL1 BYTE
367  025222  000000          T14BS2: .WORD   0               ;BSEL1 WORD
368  025224  000000                  .WORD   0               ;DATA
```

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
                                T14BFR: .BLKW   128.                    ;MESSAGE BUFFER

                                        .=<.+10>&177770
                                T14PK2:                                 ;COMMAND PACKET FOR TEST
025630  100204                          .WORD   100204                  ;WRITE CHARA. MEM. CMND., WITH IE, ACK
025632  025640                          .WORD   T14DTA                  ;ADDRESS OF SELECT DATA BLOCK
025634  000000                          .WORD   0
025636  000010                          .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE

                                T14DTA:                                 ;SELECT DATA BLOCK
025640  025226                          .WORD   T14BFR                  ;ADDRESS OF MESSAGE BUFFER
025642  000000                          .WORD   0
025644  000400                          .WORD   256.                    ;LENGTH OF MESSAGE BUFFER
025646  000000  000000                  .WORD   0,0

                                ;+
                                ;LOCAL TEXT MESSAGES FOR TEST
                                ;-

025652  127  122  111  T14NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
025726  127  122  111  T142REJ:        .ASCIZ  'WRITE SUBSYSTEM MEMORY Not Rejected With Non-Zero Mode Field'
026023  103  157  156  T14SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
026113  105  170  160  T14NINT:        .ASCIZ  'Expected Interrupt Not Received On WRITE SUBSYSTEM MEMORY'
026205  111  156  143  T14TSBA:        .ASCIZ  'Incorrect TSBA Address After WRITE SUBSYSTEM MEMORY'
026271  102  141  163  TST14ID:        .ASCIZ  'Basic WRITE SUBSYSTEM MEMORY Command'
                                        .EVEN

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

                                T14REST:
                                        SAVREG                          ;SAVE THE REGISTERS
026342  012701  025210                  MOV     #T14PACKET,R1           ;START OF THE PACKET
026346  012721  100206                  MOV     #100206,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK, IE
026352  012721  025220                  MOV     #T14DATA,(R1)+          ;ADDRESS OF DATA BLOCK
026356  005021                          CLR     (R1)+                   ;EXTENDED ADDRESS
026360  012721  000006                  MOV     #6.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
026364  005021                          CLR     (R1)+                   ;CLEAR BSELO AND BSEL1
026366  005021                          CLR     (R1)+                   ;CLEAR SEL2
026370  005011                          CLR     (R1)                    ;CLEAR DATA AREA
026372  000207                          RTS     PC                      ;RETURN

                                T14RST:
026374                                  SAVREG                          ;SAVE THE REGISTERS
026400  012701  025630                  MOV     #T14PK2,R1              ;START OF THE PACKET
026404  012721  100204                  MOV     #100204,(R1)+           ;WRITE CHARA. WITH ACK, IE
026410  012721  025640                  MOV     #T14DTA,(R1)+           ;ADDRESS OF CHARAISTICS DATA BLOCK
026414  005021                          CLR     (R1)+                   ;EXTENDED ADDRESS
```

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
    428 026416  012721  000010                MOV     #8.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
    429 026422  012721  025226                MOV     #T14BFR,(R1)+   ;MESSAGE BUFFER ADDRESS
    430 026426  005021                        CLR     (R1)+
    431 026430  012721  000400                MOV     #256.,(R1)+     ;LENGTH OF MESSAGE BUFFER
    432 026434  005021                        CLR     (R1)+
    433 026436  005011                        CLR     (R1)
    434 026440  005037  025226                CLR     T14BFR          ;CLEAR 1ST LOC IN MESSAGE BUFFER
    435 026444  000207                        RTS     PC              ;RETURN
    436 026446                                ENDTST
        026446                                                    L10037:
        026446  104401                                                  TRAP    C$ETST
```

```
                                        .SBTTL  TEST  3: DMA MEMORY ADDRESSING

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

                                        BGNTST
                                                                                        T3::
026450                          MOV     #TST12ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
026454                          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
026460                          MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
026466                          INC     T38FLG                  ;SET TEST FLAG
026472                          JSR     PC,MEMCK                ;CHECK MEMORY

026476                  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
                                        ;--

026476                                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
026476                                                                                  T3.1:
026476  104402                                                                            TRAP    C$BSUB

                                        ;Write to TSSR to soft initialize
026500  004737  015774                          JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
026504  103405                                  BCS     15$                     ;BR IF SOFT INIT = OK
026506                                          NEXT.ERRNO
026506  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
026510                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
026510  104455                                                                          TRAP    C$ERDF
026512  000455                                                                          .WORD   301
026514  003652                                                                          .WORD   SFIERR
026516  012034                                                                          .WORD   SFIMSG

                                        ;Do a WRITE CHARACTERISTICS to setup a message buffer
026520                                  15$:
026520  012704  030310                          MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
026524  004737  031670                          JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
026530  005037  003134                          CLR     KTENABLE                ;TURN OFF KT-11
026534  010465  000000                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
026540  004737  016336                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
026544                                          FORCERROR       17$
026560  103405                                  BCS     20$                     ;BR IF SSR SET IN CHKTSSR
026562  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
026564                                          NEXT.ERRNO
026564                                  17$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
026564  104455                                                                          TRAP    C$ERDF
026566  000456                                                                          .WORD   302
026570  030622                                                                          .WORD   T12WRTSSR
026572  012046                                                                          .WORD   PKTSSR

                                        ;Verify a Get Status can be fetched from each address
                                        ;Get a valid modulo-4 test address
                                        ;Do a GET STATUS command from the test address
026574  005037  002222                  20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
026600  005037  030360                          CLR     T12KT                   ;TEST ABOVE 28K SWITCH
026604  012702  030364                          MOV     #T12BLK,R2              ;POINT TO TEST PATTERN TABLE
026610                                  T12LOOP:
026610  005037  003134                          CLR     KTENABLE                ;TURN OFF ABOVE 28K TEST FLAG
026614  012201                                  MOV     (R2)+,R1                ;GET TEST PATTERN ADDRESS
026616  005000                                  CLR     R0                      ;ASSUME NO TEST ABOVE 28K
026620  005737  030360                          TST     T12KT                   ;TEST ABOVE 28K THIS TIME?
026624  001407                                  BEQ     25$                     ;BR IF NO
026626  016200  177776                          MOV     -2(R2),R0               ;GET TEST PATTERN AGAIN
026632  042700  177774                          BIC     #^C<A1716>,R0           ;SAVE 18 BIT ADDRESS ONLY
026636  012737  000001  003134                  MOV     #1,KTENABLE             ;TURN ON ABOVE 28K TEST FLAG
026644  004737  031366                  25$:    JSR     PC,T12CONVERT           ;CONVERT TEST PATTERN TO TEST ADDRESS
```

TEST 3: SUBTEST 1: GET STATUS SELECTED LOCATIONS

```
                                        BCC     65$                     ;BR IF INVALID PACKET ADDRESS
                                        MOV     T12LOADD,R4             ;COPY CURRENT PACKET LOW ADDRESS
                                        MOV     T12HIADD,R3             ;COPY CURRENT PACKET HIGH ADDRESS
                                        JSR     PC,T12SETGET            ;SETUP CURRENT PACKET TO GET STATUS
                                        BIC     #^C<A1716>,R3           ;SAVE ADDRESS BITS 17-16
                                        BIS     R3,R4                   ;SETUP 18 BIT PACKET ADDRESS
                                        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
                                        NEXT.ERRNO
                                32$:    ERRDF   ERRNO,T12GETSSR,PKTGETS ;DEVICE FATAL SSR FAILED TO SET
026730  104455                                                                    TRAP    C$ERDF
026732  000457                                                                    .WORD   303
026734  030546                                                                    .WORD   T12GETSSR
026736  012064                                                                    .WORD   PKTGETS
                                40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
026740  104406                                                                    TRAP    C$CLP1
                                65$:
                                        FORCEXIT        80$
                                        CMP     R2,#T12TBE              ;DONE ALL TSTBLK TEST PATTERNS?
                                        BHIS    70$                     ;BR IF YES
                                        JMP     T121LOOP                ;DO ANOTHER MODULO- 4 ADDRESS
                                70$:    TST     T12KT                   ;DONE ABOVE 28K TESTING TOO?
                                        BGT     80$                     ;BR IF YES
                                        TST     KTFLG                   ;ANY MEMORY ABOVE 28K ON SYSTEM?
                                        BEQ     80$                     ;BR IF NO
                                        MOV     #1,T12KT                ;SET SWITCH
                                        MOV     #T12BLK,R2              ;RESET TEST PATTERN TABLE
                                        JMP     T121LOOP                ;DO ABOVE 28K TESTING
                                80$:    JSR     PC,KTOFF                ;TURN OFF KT11
                                        ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
027022                                                                          L10043:
027022  104403                                                                  TRAP    C$ESUB
                                        TST     FATFLG                  ;ANY FATAL ERRORS ?
                                        BEQ     100$                    ;BRANCH IF NOT
                                        JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
                                100$:

                                        .SBTTL  TEST 3: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS
                                ;**
                                ; TEST 3: SUBTEST 2:
                                ;
                                ; SUBTEST DESCRIPTION:
                                ;
                                ;       This subtest verifies the controller can deposit message packets
                                ;       to all available memory locations.
                                ;       Write Characteristics commands are executed with message
                                ;       buffer addresses set to various addresses in each available
                                ;       memory location.
                                ;       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 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
                                ;--

027036                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
027036                                                                                  T3.2:
027036  104402                                                                          TRAP    C$BSUB

                                ;Write to TSSR to soft initialize
027040  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
027044  103405                          BCS     15$                     ;BR IF SOFT INIT = OK
027046                                  NEXT.ERRNO
027046  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
027050                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
027050  104455                                                                          TRAP    C$ERDF
027052  000460                                                                          .WORD   304
027054  003652                                                                          .WORD   SFIERR
027056  012034                                                                          .WORD   SFIMSG

                                ;Do a WRITE CHARACTERISTICS to setup a message buffer to compare
027060                          15$:
027060  012704  030310                  MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
027064  004737  031670                  JSR     PC,T12SWRT              ;SET PACKET TO WRITE CHARACTERISTICS
027070  004737  017274                  JSR     PC,KTOFF                ;TURN OFF KT-11
027074  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
027100  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
027104                                  FORCERROR       17$
027120  103405                          BCS     20$                     ;BR IF SSR SET IN CHKTSSR
027122  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
027124                                  NEXT.ERRNO
027124                          17$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
027124  104455                                                                          TRAP    C$ERDF
027126  000461                                                                          .WORD   305
027130  030622                                                                          .WORD   T12WRTSSR
027132  012046                                                                          .WORD   PKTSSR

                                ;Get a valid modulo-4 test address
                                ;Set the packet message buffer to the test address
                                ;Do a WRITE CHARACTERISTICS
027134  005037  002222          20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
027140  012703  030364                  MOV     #T12BLK,R3              ;POINT TO TEST PATTERN TABLE
027144                          T122LOOP:
027144  012301                          MOV     (R3)+,R1                ;GET TEST PATTERN ADDRESS
027146  010100                          MOV     R1,R0                   ;GET ADDRESS ALL "18 BITS"
027150  042700  177774                  BIC     #177774,R0              ;LEAVE ONLY A17 AND A16
027154  042701  000003                  BIC     #3,R1                   ;GET RID OF A17 AND A16
```

```
642 027160  004737  031366                  JSR     PC,T12CONVERT           ;CONVERT TEST PATTERN TO TEST ADDRESS
643 027164  103402                          BCS     25$                     ;BR IF VALID MESSAGE BUFFER ADDRESS
644 027166  000137  027264                  JMP     150$                    ;GET ANOTHER TEST PATTERN TO TRY
645 027172  012704  030310          25$:    MOV     #T12PACKET,R4           ;SET THE COMMAND PACKET ADDRESS
646 027176  004737  031670                  JSR     PC,T12SWRT              ;SETUP T12PACKET TO WRITE CHAR.
647 027202  013737  030354  030320          MOV     T12LOADD,T12DATA        ;SETUP LOW ORDER MESSAGE BUFFER ADD.
648 027210  013737  030352  030322          MOV     T12HIADD,T12DATA+2      ;SETUP HIGH ORDER MESSAGE BUFFER ADD.
649 027216  004737  017274                  JSR     PC,KTOFF                ;TURN OFF KT-11
650 027222  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
651 027226  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
652 027232                                  FORCERROR       32$
653 027246  103405                          BCS     50$                     ;BR IF SSR SET IN CHKTSSR
654 027250  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
655 027252                                  NEXT.ERRNO
656 027252                          32$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
    027252  104455                                                                  TRAP    C$ERDF
    027254  000462                                                                  .WORD   306
    027256  030622                                                                  .WORD   T12WRTSSR
    027260  012046                                                                  .WORD   PKTSSR
657 027262                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
    027262  104406                                                                  TRAP    C$CLP1
658 027264                          150$:
659 027264                                  FORCEXIT        160$
660 027274  020327  030516                  CMP     R3,#T12TBE              ;DONE ALL TST12BLK TEST PATTERNS?
661 027300  103002                          BHIS    160$                    ;BR IF YES
662 027302  000137  027144                  JMP     T12LOOP                 ;DO ANOTHER MODULO- 4 ADDRESS
663 027306  004737  017274          160$:   JSR     PC,KTOFF                ;TURN OFF KT11
664 027312                                  ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
    027312                                                                          L10044:
    027312  104403                                                                  TRAP    C$ESUB
665 027314  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
666 027320  001402                          BEQ     180$                    ;BRANCH IF NOT
667 027322  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
668 027326                          180$:
669
670
671
672                                         .SBTTL  TEST 3: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS
673                                 ;**
674                                 ; TEST 3: SUBTEST 3:
675                                 ;
676                                 ; SUBTEST DESCRIPTION:
677                                 ;
678                                 ;       This subtest verifies the controller can fetch a
679                                 ;       Write Characteristics data block from all available
680                                 ;       memory locations.
681                                 ;       Write Characteristics commands are executed with
682                                 ;       characteristic data blocks at various memory addresses.
683                                 ;       The various memory addresses are determined by floating
684                                 ;       a 1 then a 0 through the address bits.
685                                 ;
686                                 ; TEST STEPS:
687                                 ;
688                                 ;    BEGIN
689                                 ;       Write to TSSR to soft initialize
690                                 ;
691                                 ;       REPEAT FOR SELECTED VALID ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
```

```
                                ;         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
                                ;--

027326                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
027326                                                                                  T3.3:
027326  104402                                                                          TRAP    C$BSUB

                                ;Write to TSSR to soft initialize
027330  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
027334  103405                          BCS     20$                     ;BR IF SOFT INIT = OK
027336                                  NEXT.ERRNO
027336  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
027340                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
027340  104455                                                                          TRAP    C$ERDF
027342  000463                                                                          .WORD   307
027344  003652                                                                          .WORD   SFIERR
027346  012034                                                                          .WORD   SFIMSG

                                ;Get a valid test address
027350  005037  002222          20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
027354  005037  030360                  CLR     T12KT                   ;TEST ABOVE 28K SWITCH
027360  012703  030364                  MOV     #T128LK,R3              ;POINT TO TEST PATTERN TABLE
027364                          T123LOOP:
027364  005037  003134                  CLR     KTENABLE                ;TURN OFF ABOVE 28K TEST FLAG
027370  012301                          MOV     (R3)+,R1                ;GET TEST PATTERN ADDRESS
027372  010100                          MOV     R1,R0                   ;GET ADDRESS ALL "18 BITS"
027374  042700  177774                BIC     #177774,R0              ;LEAVE ONLY A17 AND A16
027400  042701  000003                BIC     #3,R1                   ;GET RID OF A17 AND A16
027404  005737  030360                TST     T12KT                   ;TEST ABOVE 28K THIS TIME?
027410  001407                          BEQ     25$                     ;BR IF NO
027412  016300  177776                MOV     -2(R3),R0               ;GET TEST PATTERN AGAIN
027416  042700  177774                BIC     #^C<A1716>,R0           ;SAVE 18 BIT ADDRESS ONLY
027422  012737  000001  003134        MOV     #1,KTENABLE             ;TURN ON ABOVE 28K TEST FLAG
027430  004737  031366          25$:    JSR     PC,T12CONVERT           ;CONVERT TEST PATTERN TO TEST ADDRESS
027434  103402                          BCS     30$                     ;BR IF VALID TEST ADDRESS
027436  000137  027540                JMP     60$                     ;GET NEXT TEST PATTERN

                                ;Set the test packet characteristics data pointer to the test address
027442  012704  030310          30$:    MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
027446  004737  031670                  JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
027452  013764  030354  000002        MOV     T12LOADD,PKLOW(R4)      ;STORE CHAR. DATA PTR LOW ADDRESS
027460  013764  030352  000004        MOV     T12HIADD,PKHI(R4)       ;STORE CHAR. DATA PTR HIGH ADDRESS
027466  004737  032000                JSR     PC,T12CHAR              ;STORE EXPECTED DATA IN DATA BLOCK

                                ;Do a WRITE CHARACTERISTIC command
027472  004737  017274                JSR     PC,KTOFF                ;TURN OFF KT-11
027476  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
027502  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
027506                                  FORCERROR       32$
027522  103405                          BCS     40$                     ;BR IF SSR SET IN CHKTSSR
027524  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

TEST 3: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS

```
                                        NEXT.ERRNO
                              32$:      ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
027526 104455                                                                   TRAP    C$ERDF
027530 000464                                                                   .WORD   308
027532 030622                                                                   .WORD   T12WRTSSR
027534 012046                                                                   .WORD   PKTSSR
                              40$:      CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
027536 104406                                                                   TRAP    C$CLP1
                              60$:
027540 020327 030516                    CMP     R3,#T12TBE              ;DONE ALL TSTBLK TEST PATTERNS?
027544 103002                           BHIS    65$                     ;BR IF YES
027546 000137 027364                    JMP     T123LOOP                ;DO ANOTHER MODULO- 4 ADDRESS
027552 005737 030360          65$:      TST     T12KT                   ;DONE ABOVE 28K TESTING TOO?
027556 003012                           BGT     70$                     ;BR IF YES
027560 005737 003132                    TST     KTFLG                   ;ANY MEMORY ABOVE 28K ON SYSTEM?
027564 001407                           BEQ     70$                     ;BR IF NO
027566 012737 000001 030360             MOV     #1,T12KT                ;SET SWITCH
027574 012703 030364                    MOV     #T12BLK,R3              ;RESET TEST PATTERN TABLE
027600 000137 027364                    JMP     T123LOOP                ;DO ABOVE 28K TESTING
027604 004737 017274          70$:      JSR     PC,KTOFF                ;TURN OFF KT11
027610                                  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\
027610                                                                          L10045:
027610 104403                                                                   TRAP    C$ESUB
027612 005737 002222                    TST     FATFLG                  ;ANY FATAL ERRORS ?
027616 001402                           BEQ     75$                     ;BRANCH IF NOT
027620 004737 017202                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
027624                        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 11/23B system with more than
                              ;       128K words of memory!
                              ;       ********************************************************************
                              ;
                              ; TEST STEPS:
                              ;
                              ;    BEGIN
                              ;       Write to TSSR to soft initialize
                              ;       Do a write characteristic command
                              ;       Invert the extended features switch
                              ;
                              ;       REPEAT FOR SELECTED NON-EXISTENT MEMORY ADDRESSES
```

```
                                ;          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
                                ;--

027624                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
027624                                                                                      T3.4:
027624  104402                                                                                      TRAP    C$BSUB

027626  005737  003144                  TST     T23A                    ;26-APR-83 REV B - CHK FOR 23A CPU
027632  001406                          BEQ     5$                      ;26-APR-83 REV B - BR, IF NOT 23A
027634  023727  002120  007777          CMP     L$HIME,#7777            ;26-APR-83 REV B - CHK FOR > 256KB
027642  103402                          BLO     5$                      ;26-APR-83 REV B - BR, IF < 256KB
027644  000137  030236                  JMP     NOEXTF                  ;26-APR-83 REV B - JMP OVER 256KB
027650                          5$:
027650  005737  003136                  TST     NXMFLG                  ;GOT ENOUGH MEMORY?
027654  001002                          BNE     10$                     ;IF SET STAY
027656  000137  030236                  JMP     NOEXTF                  ;LEAVE IF NOT SET

                                ;Write to TSSR to soft initialize

027662  004737  015774          10$:    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
027666  103405                          BCS     11$                     ;BR IF SOFT INIT = OK
027670                                  NEXT.ERRNO
027670  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
027672                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
027672  104455                                                                              TRAP    C$ERDF
027674  000465                                                                              .WORD   309
027676  003652                                                                              .WORD   SFIERR
027700  012034                                                                              .WORD   SFIMSG

                                ;Do a WRITE CHARACTERISTIC command so to invert switch

027702                          11$:    CKLOOP                          ;LOOP IF SELECTED
027702  104406                                                                              TRAP    C$CLP1
027704  012704  030310                  MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
027710  004737  031670                  JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
027714  005037  003134                  CLR     KTENABLE                ;TURN OFF KT-11
027720  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
027724  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
027730                                  FORCERROR       15$
027744  103405                          BCS     17$                     ;BR IF SSR SET IN CHKTSSR
027746  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
027750                                  NEXT.ERRNO
027750                          15$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
027750  104455                                                                              TRAP    C$ERDF
027752  000466                                                                              .WORD   310
027754  030622                                                                              .WORD   T12WRTSSR
027756  012046                                                                              .WORD   PKTSSR
027760                          17$:    CKLOOP                          ;LOOP IF SELECTED
027760  104406                                                                              TRAP    C$CLP1
```

```
838 027762  004737  021206                  JSR     PC,INVERT               ;INVERT THE SWITCH
839
840                                 ;Get an invalid test address
841
842 027766  005037  002222          20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
843 027772                          25$:
844 027772  013737  003142  030352          MOV     NXMHI,T12HIADD          ;SAVE TEST ADDRESS HIGH
845 030000  013737  003140  030354          MOV     NXMLO,T12LOADD          ;SAVE TEST ADDRESS LOW
846 030006                          T124LOOP:
847
848                                 ;Set the test packet characteristics data pointer to the
849                                 ;  test address.
850
851 030006  012704  030310          30$:    MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
852 030012  004737  031670                  JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
853 030016  013764  030354  000002          MOV     T12LOADD,PKLOW(R4)      ;STORE CHAR. DATA PTR LOW ADDRESS
854 030024  013764  030352  000004          MOV     T12HIADD,PKHI(R4)       ;STORE CHAR. DATA PTR HIGH ADDRESS
855
856                                 ;Do a WRITE CHARACTERISTIC command
857 030032  004737  017274                  JSR     PC,KTOFF                ;TURN OFF KT-11
858 030036  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
859 030042  004737  016250                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
860 030046                                  FORCERROR       32$
861 030062  103407                          BCS     40$                     ;BR IF SSR SET IN CHKTSSR
862 030064  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
863 030066                                  NEXT.ERRNO
864 030066                          32$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
    030066  104455                                                                  TRAP    C$ERDF
    030070  000467                                                                  .WORD   311
    030072  030622                                                                  .WORD   T12WRTSSR
    030074  012046                                                                  .WORD   PKTSSR
865 030076  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
866 030102                          40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
    030102  104406                                                                  TRAP    C$CLP1
867 030104                                  FORCERROR       45$,NOTSSR
868 030114                                  ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
    030114  104410                                                                  TRAP    C$ESCAPE
    030116  000124                                                                  .WORD   L10046-.
869                                 ;If TSSR register NXM bit not set then print error message
870 030120                          45$:
871 030120  016501  000002                  MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
872 030124                                  FORCERROR       52$
873 030140  032701  004000                  BIT     #NXM,R1                 ;NXM SET?
874 030144  001012                          BNE     60$                     ;BR IF YES
875 030146                                  NEXT.ERRNO
876 030146  013737  030354  002240  52$:    MOV     T12LOADD,ERRLO          ;MEMORY TEST ADDRESS LOW
877 030154  013737  030352  002236          MOV     T12HIADD,ERRHI          ;MEMORY TEST ADDRESS HIGH
878 030162                                  ERRHRD  ERRNO,T12NXM,ADDSSR     ;REPORT ADDRESS AND TSSR ERROR
    030162  104456                                                                  TRAP    C$ERHRD
    030164  000470                                                                  .WORD   312
    030166  031257                                                                  .WORD   T12NXM
    030170  012126                                                                  .WORD   ADDSSR
879
880 030172                          60$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
    030172  104406                                                                  TRAP    C$CLP1
881 030174                                  FORCEXIT        90$
882 030204  005737  003144                  TST     T23A                    ;IS IT A 11/23A?
```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
883 030210  001012                        BNE     90$                 ;YES WERE DONE
884 030212  013700  030352                MOV     T12HIADD,R0         ;GET CURRENT HIGH ADDRESS
885 030216  005200                  65$:   INC     R0                  ;GET NEXT ADDRESS
886 030220  020027  000077                CMP     R0,#77              ;DONE A21-A16?
887 030224  101004                        BHI     90$                 ;BR IF YES
888 030226  010037  030352          75$:   MOV     R0,T12HIADD         ;SETUP NEW HIGH ORDER ADDRESS
889 030232  000137  030006                JMP     T124LOOP            ;DO ANOTHER NON-EXISTENT ADDRESS
890 030236                          90$:
891 030236                          NOEXTF:
892 030236  004737  017274                JSR     PC,KTOFF            ;TURN OFF KT11
893 030242                                ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\
    030242                                                                    L10046:
    030242  104403                                                                TRAP    C$ESUB
894 030244  005737  002222                TST     FATFLG              ;ANY FATAL ERRORS ?
895 030250  001402                        BEQ     100$                ;BRANCH IF NOT
896 030252  004737  017202                JSR     PC,CKDROP           ;TRY TO DROP THE UNIT
897 030256  004737  016456          100$: JSR     PC,TSTLOOP          ;SHOULD WE DO ITERATIONS?
898 030262  103002                        BCC     105$                ;BR IF NO
899 030264  000137  026476                JMP     T12LOOP             ;LOOP UNTIL ITERATION COUNT DONE
900 030270                          105$:
901 030270  004737  017274                JSR     PC,KTOFF            ;TURN OFF MEMORY MANAGEMENT
902 030274  005037  003150                CLR     T3BFLG              ;CLEAR TEST FLAG
903 030300                                EXIT    TST                 ;ALL DONE THIS TEST
    030300  104432                                                                TRAP    C$EXIT
    030302  001540                                                                .WORD   L10042-.
904
905
906                                 ;+
907                                 ;LOCAL STORAGE FOR THIS TEST
908                                 ;-
909
911         030310                          .=<.+10>&177770
913 030310                          T12PACKET:
914 030310  100004                          .WORD   100004              ;COMMAND PACKET FOR TEST
915 030312  030320                          .WORD   T12DATA             ;WRITE CHARACTERISTICS COMMAND, WITH ACK
916 030314  000000                          .WORD   0                   ;ADDRESS OF CHARACTERISTICS BLOCK
917 030316  000010                          .WORD   8.                  ;STARTING VALUE OF BLOCK SIZE
918
919 030320                          T12DATA:                            ;CHARACTERISTICS DATA BLOCK
920 030320  030332                          .WORD   T12BFR              ;LOW ADDRESS OF MESSAGE BUFFER
921 030322  000000                          .WORD   0                   ;HIGH ORDER OF MESSAGE BUFFER
922 030324  000016                          .WORD   14.                 ;LENGTH OF MESSAGE BUFFER
923 030326  000000  000000                  .WORD   0,0
924
925 030332                          T12BFR: .BLKW   8.                  ;MESSAGE BUFFER
926
927 030352  000000                T12HIADD:       .WORD   0             ;HIGH ADDRESS
928 030354  000000                T12LOADD:       .WORD   0             ;LOW ADDRESS
929 030356  000000                T12PAR6:        .WORD   0             ;ADDRESS IN PAR FORMAT
930 030360  000000                T12KT:          .WORD   0             ;TEST ABOVE 28K SWITCH
931 030362  000000                T124TST:        .WORD   0             ;ADDRESS TEST BIT
932                                 ;+
933                                 ;
934                                 ;TABLE OF ADDRESSES
935                                 ;
936                                 ;-
937 030364  000001                T12BLK: .WORD   000001
```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
    938 030366  000002                          .WORD   000002
    939 030370  000003                          .WORD   000003
    940 030372  000005                          .WORD   000005
    941 030374  000006                          .WORD   000006
    942 030376  000007                          .WORD   000007
    943 030400  000011                          .WORD   000011
    944 030402  000012                          .WORD   000012
    945 030404  000013                          .WORD   000013
    946 030406  000021                          .WORD   000021
    947 030410  000022                          .WORD   000022
    948 030412  000023                          .WORD   000023
    949 030414  000041                          .WORD   000041
    950 030416  000042                          .WORD   000042
    951 030420  000043                          .WORD   000043
    952 030422  000101                          .WORD   000101
    953 030424  000102                          .WORD   000102
    954 030426  000103                          .WORD   000103
    955 030430  000201                          .WORD   000201
    956 030432  000202                          .WORD   000202
    957 030434  000203                          .WORD   000203
    958 030436  000401                          .WORD   000401
    959 030440  000402                          .WORD   000402
    960 030442  000403                          .WORD   000403
    961 030444  001001                          .WORD   001001
    962 030446  001002                          .WORD   001002
    963 030450  001003                          .WORD   001003
    964 030452  002001                          .WORD   002001
    965 030454  002002                          .WORD   002002
    966 030456  002003                          .WORD   002003
    967 030460  004001                          .WORD   004001
    968 030462  004002                          .WORD   004002
    969 030464  004003                          .WORD   004003
    970 030466  010001                          .WORD   010001
    971 030470  010002                          .WORD   010002
    972 030472  010003                          .WORD   010003
    973 030474  020001                          .WORD   020001
    974 030476  020002                          .WORD   020002
    975 030500  020003                          .WORD   020003
    976 030502  040001                          .WORD   040001
    977 030504  040002                          .WORD   040002
    978 030506  040003                          .WORD   040003
    979 030510  100001                          .WORD   100001
    980 030512  100002                          .WORD   100002
    981 030514  100003                          .WORD   100003
    982 030516  177777                T12TBE:   .WORD           177777
    983                                ;+
    984                                ;LOCAL TEXT MESSAGES FOR TEST
    985                                ;-
    986
    987 030520      104     115     101  TST12ID:          .ASCIZ  'DMA Memory Addressing'
    988 030546      103     157     156  T12GETSSR:        .ASCIZ  'Contents of TSSR Incorrect After GET STATUS'
    989 030622      103     157     156  T12WRTSSR:        .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
    990 030711      115     145     163  T12MSGBUF:        .ASCIZ  'Message Buffer Contents Incorrect After WRITE CHARACTERISTICS'
    991 031007      102     141     143  T12BKGND:         .ASCIZ  'Background Pattern Disturbed By WRITE CHARACTERISTICS'
    992 031075      105     170     160  T12NINT:          .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
    993 031166      127     162     151  T12DPR:           .ASCIZ  'Write Characteristic data in ram does not match expected'
    994 031257      124     123     123  T12NXM:           .ASCIZ  'TSSR NXM bit failed to set when non-existent memory address specifi
ed'
```

```
                                        .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
                                ;-
031366                          T12CONVERT:
031366                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
031372  005037  030354                  CLR     T12LOADD        ;CLEAR LOW ADDRESS
031376  005037  030352                  CLR     T12HIADD        ;CLEAR HIGH ADDRESS
031402  005037  030356                  CLR     T12PAR6         ;CLEAR PAR6 BIASED ADDRESS
031406  042701  170000                  BIC     #^C<7777>,R1    ;FORCE TO LOWER 12 BITS OF ADDRESS
031412  010005                          MOV     R0,R5           ;SAVE HIGH ORDER ADDRESS BITS
031414  004737  017274                  JSR     PC,KTOFF        ;SHUTOFF MEMORY MANAGEMENT
031420  013702  003124                  MOV     FREE,R2         ;GET FIRST FREE ADDRESS
031424  062702  000020                  ADD     #16.,R2         ;IN CASE TEST PATTERN=0
031430  060102                          ADD     R1,R2           ;ADD IN TEST PATTERN
031432  042702  000003                  BIC     #3,R2           ;MAKE IT MODULO-4
031436  013703  003130          25$:    MOV     FREEHI,R3       ;GET LAST FREE ADDRESS
031442  162703  000020                  SUB     #16.,R3         ;SAVE AT LEAST 8 WORDS (IN CASE MESSAGE BUFFER)
031446  010237  030354                  MOV     R2,T12LOADD     ;SAVE POSSIBLE LOW ADDRESS
031452  010237  030356                  MOV     R2,T12PAR6      ;SAVE IT IN PAR6 BIASED TOO
031456  020203                          CMP     R2,R3           ;IS THIS ADDRESS ABOVE FREE SPACE?
031460  101007                          BHI     35$             ;BR IF YES
031462  020237  003124                  CMP     R2,FREE         ;IS IT IN FREE SPACE?
031466  103007                          BHIS    50$             ;BR IF YES- ITS GOOD
031470  005737  003134                  TST     KTENABLE        ;TESTING ABOVE 28K?
031474  001004                          BNE     50$             ;BR IF YES
031476  000424                          BR      90$             ;BR IF NOT IN FREE SPACE
031500  162702  000020          35$:    SUB     #16.,R2         ;FORCE FIT THE TEST PATTERN
031504  000754                          BR      25$             ;TRY THIS TEST PATTERN ADDRESS
031506                          50$:
031506  005737  003134                  TST     KTENABLE        ;TESTING ABOVE 28K?
031512  001420                          BEQ     100$            ;BR IF NO
031514  005737  003132                  TST     KTFLG           ;ANY MEMORY ABOVE 28K?
031520  001413                          BEQ     90$             ;BR IF NO
031522  004737  017256                  JSR     PC,KTON         ;TURN ON MEMORY MANAGEMENT
031526  010500                          MOV     R5,R0           ;GET HIGH ORDER ADDRESS
031530  010037  030352                  MOV     R0,T12HIADD     ;SAVE POSSIBLE HIGH ADDRESS
031534  010201                          MOV     R2,R1           ;GET COMPUTED LOW ORDER ADDRESS
031536  004737  017316                  JSR     PC,SETMAP       ;RETURN PAR6 BIASED ADDRESS IN R0
```

```
1052 031542  010037  030356                    MOV     R0,T12PAR6      ;COPY PAR6 BIASED ADDRESS
1053 031546  103403                            BCS     105$            ;BR IF VALID ADDRESS
1054 031550  000241                  90$:      CLC                     ;CLR C BIT FOR FAILURE
1055 031552  000401                            BR      105$            ;
1056 031554  000261                  100$:     SEC                     ;SET SUCCESS
1057 031556  000207                  105$:     RTS     PC              ;RETURN
1058
1059
1060                                 ;+
1061                                 ;
1062                                 ;ROUTINE TO READ THE FIRST 2 BYTES FROM RAM
1063                                 ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
1064                                 ;
1065                                 ;INPUT:
1066                                 ;
1067                                 ;       R4      ADDRESS OF THE COMMAND PACKET
1068                                 ;       R5      FIRST DEVICE UNIBUS ADDRESS
1069                                 ;
1070                                 ;OUTPUT:
1071                                 ;
1072                                 ;       CARRY   SET - RAM MATCHES PACKET
1073                                 ;               CLR - RAM DOES NOT MATCH PACKET
1074                                 ;
1075                                 ;IMPLICIT OUTPUT:
1076                                 ;
1077                                 ;       THE TABLE RAMDATA IS FILLED WITH THE
1078                                 ;       DATA HELD IN RAM.
1079                                 ;       RAMSIZ  SET TO 2 FOR PRAMPKT ROUTINE
1080                                 ;
1081                                 ;SIDE EFFECTS:
1082                                 ;
1083                                 ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1084                                 ;
1085                                 ;-
1086 031560                         T12CKRAM::
1087 031560                                 SAVREG                          ;SAVE THE GENERAL REGISTERS
1088 031564  012701  002242                MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
1089 031570  012702  000201                MOV     #RMPKTBEG,R2            ;BYTE ADDRESS OF FIRST RAM DATA
1090 031574  005003                        CLR     R3                      ;CLEAR THE ERROR FLAG
1091 031576  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR
1092 031602  112765  000000  000000        MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
1093 031610  004737  016336          10$:  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1094 031614  010265  000000                MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
1095 031620  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1096 031624  116511  000000                MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
1097 031630  122124                        CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
1098 031632  001401                        BEQ     20$                     ;BRANCH IF OK
1099 031634  005203                        INC     R3                      ;SET ERROR FLAG
1100 031636  005202                  20$:  INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
1101 031640  020227  000203                CMP     R2,#RMPKTBEG+2          ;DONE 2 BYTES?
1102 031644  002761                        BLT     10$                     ;BR IF NO
1103 031646  005703                        TST     R3                      ;WAS AN ERROR FOUND ?
1104 031650  001402                        BEQ     30$                     ;BRANCH IF NOT
1105 031652  000241                        CLC                             ;CLEAR CARRY TO SHOW ERROR
1106 031654  000401                        BR      50$                     ;AND EXIT
1107 031656  000261                  30$:  SEC                             ;SHOW GOOD COMPARE
1108 031660  012737  000002  002302  50$:  MOV     #2,RAMSIZ               ;SETUP RAMSIZ
```

```
1109 031666  000207                        RTS     PC                      ;RETURN
1110
1111                                    ;*
1112                                    ;
1113                                    ;ROUTINE TO SETUP PACKET TO WRITE CHARACTERISTICS
1114                                    ;-
1115
1116 031670                             T12SWRT:
1117 031670                                     SAVREG                          ;SAVE THE REGISTERS
1118 031674  012701  030310                      MOV     #T12PACKET,R1           ;START OF THE PACKET
1119 031700  012721  100004                      MOV     #100004,(R1)+           ;WRITE CHARACTERISTICS WITH ACK
1120 031704  012721  030320                      MOV     #T12DATA,(R1)+          ;ADDRESS OF CHAR DATA BLOCK
1121 031710  005021                              CLR     (R1)+                   ;EXTENDED ADDRESS
1122 031712  012721  000010                      MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
1123 031716  012721  030332                      MOV     #T12BFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
1124 031722  005021                              CLR     (R1)+
1125 031724  012721  000016                      MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
1126 031730  005021                              CLR     (R1)+
1127 031732  005011                              CLR     (R1)
1128 031734  000207                              RTS     PC                      ;RETURN
1129                                    ;*
1130                                    ;
1131                                    ;ROUTINE TO SETUP A GET STATUS COMMAND PACKET AT CURRENT PACKET ADDRESS
1132                                    ;
1133                                    ;       R3      HIGH ORDER PACKET ADDRESS
1134                                    ;       R4      LOW ORDER PACKET ADDRESS
1135                                    ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG CLEAR
1136                                    ;
1137                                    ;-
1138
1139 031736                             T12SETGET:
1140 031736                                     SAVREG                          ;SAVE THE REGISTERS
1141 031742  010401                              MOV     R4,R1                   ;GET LOW ORDER ADDRESS
1142 031744  005737  003134                      TST     KTENABLE                ;TESTING ABOVE 28K?
1143 031750  001404                              BEQ     10$                     ;BR IF NO
1144 031752  010300                              MOV     R3,R0                   ;GET HIGH ORDER ADDRESS
1145 031754  004737  017316                      JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
1146 031760  010001                              MOV     R0,R1                   ;GET ADDRESS
1147 031762  012700  000017             10$:     MOV     #P.GETSTATUS,R0         ;GET STATUS COMMAND CODE NO IE
1148 031766  052700  100000                      BIS     #P.ACK,R0               ;SET ACK
1149 031772  010021                              MOV     R0,(R1)+                ;STORE GET STATUS IN PACKET
1150 031774  005021                              CLR     (R1)+                   ;CLEAR UNUSED WORD
1151 031776  000207                              RTS     PC                      ;RETURN
1152
1153
1154                                    ;*
1155                                    ;
1156                                    ;ROUTINE TO SETUP A CHARACTERISTIC DATA BLOCK AT A TEST ADDRESS
1157                                    ;
1158                                    ;-
1159 032000                             T12CHAR:
1160 032000                                     SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
1161 032004  012700  030320                      MOV     #T12DATA,R0             ;GET T12PACKET DATA POINTER
1162 032010  013701  030354                      MOV     T12LOADD,R1             ;ASSUME NOT ABOVE 28K
1163 032014  005737  C03134                      TST     KTENABLE                ;TESTING ABOVE 28K?
1164 032020  001402                              BEQ     10$                     ;BR IF NO
1165 032022  013701  030356                      MOV     T12PAR6,R1              ;SET TEST ADDRESS ABOVE 28K
```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
1166 032026  012021                10$:    MOV     (R0)+,(R1)+     ;STORE DATA WORD 1
1167 032030  012021                        MOV     (R0)+,(R1)+     ;STORE DATA WORD 2
1168 032032  012021                        MOV     (R0)+,(R1)+     ;STORE DATA WORD 3
1169 032034  012021                        MOV     (R0)+,(R1)+     ;STORE DATA WORD 4
1170 032036  012021                        MOV     (R0)+,(R1)+     ;STORE DATA WORD 5
1171 032040  000207                        RTS     PC              ;RETURN
1172
1173 032042                                ENDTST
     032042                                                                L10042:
     032042  104401                                                            TRAP    C$ETST
```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
1175
1176                                            .SBTTL  TEST  4: RAM EXERCISER TEST
1177                                    ;*
1178                                    ;
1179                                    ;THIS TEST USES THE READ AND WRITE RAM (BOTH SINGLE AND 256
1180                                    ;LOCATIONS) SELECT CODES OF THE WRITE SUBSYSTEM MEMORY COMMAND
1181                                    ;TO EXERCISE THE CONTROLLER'S RAM MEMORY AND DMA LOGIC
1182                                    ;
1183                                    ;-
1184 032044                                     BGNTST
     032044                                                                     T4::
1185
1186
1191 032044  005737  002214                     TST     TSTCNT                  ;CHECK FOR RUN MODE
1192 032050  001402                             BEQ     10$                     ;BR, IF NOT ONLY PROGRAM RUN
1193 032052  005237  003400                     INC     SKIPT                   ;SET SKIP SW
1194 032056  012700  034503             10$:    MOV     #TST15ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
1195 032062  004737  016510                     JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
1196 032066  012737  000005  002216             MOV     #5,LOOPCNT              ;PERFORM 5 ITERATIONS
1197 032074                             T15LOOP:
1198                                    ;
1199                                    ;
1200                                    ;TEST 4, SUBTEST 1
1201                                    ;
1202                                    ;       THIS SUBTEST WRITES THE ADDRESS (8 BITS) INTO THE
1203                                    ;       RAM MEMORY SINGLE WORD (8 BITS) MODE
1204                                    ;
1205                                    ;-
1206 032074                                     BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     032074                                                                             T4.1:
     032074  104402                                                                     TRAP    C$BSUB
1207 032076                                     SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     032076  012700  000000                                                             MOV     #PRI00,R0
     032102  104441                                                                     TRAP    C$SPRI
1208 032104  005737  003400                     TST     SKIPT                   ;SHOULD WE SKIP THIS SUBTEST
1209 032110  001402                             BEQ     10$                     ;BR, IF NOW SKIP REQUIRED
1210 032112  000137  032374                     JMP     50$                     ;SKIP SUBTEST
1211 032116  004737  034522             10$:    JSR     PC,T15REST              ;SET COMMAND PACKET
1212 032122  004737  034574                     JSR     PC,T15RT2               ;SET UP OTHER COMMAND PACKET
1213 032126  004737  015774                     JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
1214 032132  103405                             BCS     20$                     ;BR IF INIT WAS OK
1218 032134  010001                             MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
1219 032136                                     ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
     032136  104455                                                                     TRAP    C$ERDF
     032140  000621                                                                     .WORD   401
     032142  003652                                                                     .WORD   SFIERR
     032144  012034                                                                     .WORD   SFIMSG
1220 032146                             20$:
1221 032146  012704  033420                     MOV     #T15PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
1222 032152  004737  010662                     JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
1223 032156  103405                             BCS     23$                     ;BR, IF COMMAND ISSUED OK
1227 032160  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1228 032162                                     ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     032162  104456                                                                     TRAP    C$ERHRD
     032164  000622                                                                     .WORD   402
     032166  005056                                                                     .WORD   WRTMSG
     032170  012034                                                                     .WORD   SFIMSG
```

TEST 4: RAM EXERCISER TEST

```
1229 032172  012703  000400          23$:   MOV     #256.,R3                ;STARTING ADDRESS FOR RAM WRITE
1230 032176  112737  000001  034131         MOVB    #1,T15BS1               ;SIZE OF TRANSFER
1231 032204  112737  000002  034130         MOVB    #2,T15BS0               ;WRITE RAM "COMMAND"
1232 032212                                 25$:
1233 032212  010337  034132                 MOV     R3,T15S2                ;ADDRESS FOR RAM
1234 032216  012704  034120                 MOV     #T15PK2,R4              ;WRITE SUBSYS MEM PACKET
1235 032222  110337  034134                 MOVB    R3,T15S3                ;DATA FOR WRITE (ADDRESS)
1236 032226  010465  000000                 MOV     R4,TSDB(R5)             ;ISSUE COMMAND
1237 032232  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR
1238 032236  103407                         BCS     30$                     ;BR, IF NO ERROR
1239 032240  010001                         MOV     R0,R1                   ;ERROR, SAVE TSSR
1243 032242                                 ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT AFTER WRITE SUB MEM
     032242  104456                                                         TRAP    C$ERHRD
     032244  000623                                                         .WORD   403
     032246  034136                                                         .WORD   T15SSR
     032250  012046                                                         .WORD   PKTSSR
1244 032252                                 ESCAPE  SUB                     ;DON'T CONTINUE IF ERROR ON WRITE
     032252  104410                                                         TRAP    C$ESCAPE
     032254  000122                                                         .WORD   L10050-.
1245 032256                                 30$:    CKLOOP                  ;SCOPE LOOP
     032256  104406                                                         TRAP    C$CLP1
1246
1247
1248 032260  005203                         INC     R3                      ;NEXT ADDRESS
1249 032262  020327  010000                 CMP     R3,#10000               ;END OF RAM MEMORY CHECK
1250 032266  001351                         BNE     25$                     ;LOOP TILL ALL RAM WRITTEN
1251 032270  005002                         CLR     R2                      ;CLEAR OUT R2 HIGH BITS
1252 032272  005303                         DEC     R3                      ;SET BACK TO 7777
1253 032274  110337  034134                 40$:    MOVB    R3,T15S3        ;GET DATA PATTERN BACK IN SHAPE
1254 032300  010337  034132                 MOV     R3,T15S2                ;ADDRESS FOR RAM READ
1255 032304  112737  000001  034130         MOVB    #1,T15BS0               ;READ RAM COMMAND
1256 032312  010465  000000                 MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS TO CONTR.
1257 032316  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1258 032322  103405                         BCS     43$                     ;BR, IF NO PROBLEM
1259 032324  010001                         MOV     R0,R1                   ;SAVE TSSR
1263 032326                                 ERRDF   ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032326  104455                                                         TRAP    C$ERDF
     032330  000624                                                         .WORD   404
     032332  034136                                                         .WORD   T15SSR
     032334  012046                                                         .WORD   PKTSSR
1264 032336                                 43$:    CKLOOP                  ;SCOPE LOOP
     032336  104406                                                         TRAP    C$CLP1
1265 032340  013701  033462                 MOV     T15BFR+20,R1            ;GET RAM READ DATA
1266 032344  010302                         MOV     R3,R2                   ;SET UP FOR COMPARE
1267 032346  120102                         CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
1268 032350  001404                         BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
1272 032352                                 ERRHRD  ERRNO,T15AM4,EXPBREC    ;WRITTEN DATA NOT = TO READ
     032352  104456                                                         TRAP    C$ERHRD
     032354  000625                                                         .WORD   405
     032356  034415                                                         .WORD   T15AM4
     032360  015502                                                         .WORD   EXPBREC
1273 032362                                 45$:    CKLOOP                  ;SCOPE LOOP
     032362  104406                                                         TRAP    C$CLP1
1274 032364  005303                         DEC     R3                      ;DROP DATA COUNTER (PATTERN)
1275 032366  020327  000377                 CMP     R3,#255.                ;AT BOTTOM YET
1276 032372  001340                         BNE     40$                     ;BR, IF MORE TO CHECK
1277 032374                                 50$:    CKLOOP                  ;SCOPE LOOP
```

TEST 4: RAM EXERCISER TEST

```
        032374  104406                                                          TRAP    C$CLP1
   1278 032376                          ENDSUB                  ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
        032376                                                                  L10050:
        032376  104403                                                          TRAP    C$ESUB
   1279
   1280
   1281 032400                          BGNSUB                  ;////////////// BEGIN SUBTEST //////////////
        032400                                                                  T4.2:
        032400  104402                                                          TRAP    C$BSUB
   1282
   1283                                 ;*
   1284                                 ;
   1285                                 ;TEST 4, SUBTEST 2
   1286                                 ;
   1287                                 ;
   1288                                 ;       THIS SUBTEST WRITES RAM WITH ALL ZEROS
   1289                                 ;       THEN WALKS AN ALL ONES WORD DOWN THROUGH MEMORY
   1290                                 ;
   1291 032402  004737  034522          JSR     PC,T15REST      ;RESTORE PACKET FOR WRITE CHARA
   1292 032406  004737  034574          JSR     PC,T15RT2       ;RESTORE PACKET FOR WRT SUB SYS MEM
   1293 032412  004737  015774          JSR     PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
   1294 032416  103405                  BCS     20$             ;BR IF INIT WAS OK
   1298 032420  010001                  MOV     R0,R1           ;CONTENTS OF TSSR REGISTER
   1299 032422                          ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
        032422  104455                                                          TRAP    C$ERDF
        032424  000626                                                          .WORD   406
        032426  003652                                                          .WORD   SFIERR
        032430  012034                                                          .WORD   SFIMSG
   1300 032432                  20$:
   1301 032432  012704  033420          MOV     #T15PACKET,R4   ;SUBROUTINE NEEDS PACKET ADDRESS
   1302 032436  004737  010662          JSR     PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
   1303 032442  103405                  BCS     25$             ;BR, IF COMMAND ISSUED OK
   1307 032444  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
   1308 032446                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
        032446  104456                                                          TRAP    C$ERHRD
        032450  000627                                                          .WORD   407
        032452  005056                                                          .WORD   WRTMSG
        032454  012034                                                          .WORD   SFIMSG
   1309 032456                  25$:
   1310 032456  112737  000001  034131      MOVB    #1,T15BS1       ;SET SIZE OF TRANSFER 1 BYTE
   1311 032464  012704  034120          MOV     #T15PK2,R4      ;SET NEW PACKET ADDRESS
   1312 032470  012703  000400          MOV     #256.,R3        ;STARTING ADDRESS IN RAM
   1313 032474  112737  000002  034130      MOVB    #2,T15BS0       ;WRITE RAM COMMAND
   1314 032502  105037  034134          CLRB    T15S3           ;SET DATA TO 000
   1315 032506  010337  034132  30$:    MOV     R3,T15S2        ;ADDRESS TO PACKET DATA AREA
   1316 032512  010465  000000          MOV     R4,TSDB(R5)     ;SEND OUT PACKET ADDRESS
   1317 032516  004737  016336          JSR     PC,CHKTSSR      ;WAIT FOR SSR
   1318 032522  103405                  BCS     33$             ;BR, IF NO PROBLEM
   1319 032524  010001                  MOV     R0,R1           ;SAVE TSSR
   1323 032526                          ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
        032526  104456                                                          TRAP    C$ERHRD
        032530  000630                                                          .WORD   408
        032532  034136                                                          .WORD   T15SSR
        032534  012046                                                          .WORD   PKTSSR
   1324 032536                  33$:    CKLOOP                  ;SCOPE LOOP
        032536  104406                                                          TRAP    C$CLP1
   1325
```

TEST 4: RAM EXERCISER TEST

```
1326
1327 032540  005203                        INC     R3                      ;NEXT ADDRESS
1328 032542  020327  010000                CMP     R3,#10000               ;END OF RAM MEMORY CHECK
1329 032546  001357                        BNE     30$                     ;BR, MORE RAM TO GO
1330 032550  005303                 35$:   DEC     R3                      ;SET BACK TO 7777
1331 032552  005002                 40$:   CLR     R2                      ;SET TO ALL ZEROS
1332 032554  112737  000001  034130        MOVB    #1,T15BS0               ;READ RAM COMMAND
1333 032562  010337  034132                MOV     R3,T15S2                ;ADDRESS TO BE READ TO PACKET DATA
1334 032566  010465  000000                MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
1335 032572  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1336 032576  103405                        BCS     41$                     ;BR, IF ALL IS WELL
1337 032600  010001                        MOV     R0,R1                   ;SAVE TSSR
1341 032602                                ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032602  104456                                                        TRAP    C$ERHRD
     032604  000631                                                        .WORD   409
     032606  034136                                                        .WORD   T15SSR
     032610  012046                                                        .WORD   PKTSSR
1342 032612                         41$:   CKLOOP                          ;SCOPE LOOP
     032612  104406                                                        TRAP    C$CLP1
1343 032614  013701  033462                MOV     T15BFR+20,R1            ;PICK UP READ DATA
1344 032620  120102                        CMPB    R1,R2                   ;BOTH SHOULD BE 00000000 BINARY
1345 032622  001404                        BEQ     42$                     ;BR, IF DATA IS GOOD
1349 032624                                ERRHRD  ERRNO,T15AM3,EXPBREC    ;CHARACTERISTICS DATA NOT CORRECT
     032624  104456                                                        TRAP    C$ERHRD
     032626  000632                                                        .WORD   410
     032630  034313                                                        .WORD   T15AM3
     032632  015502                                                        .WORD   EXPBREC
1350 032634                         42$:   CKLOOP                          ;SCOPE LOOPER
     032634  104406                                                        TRAP    C$CLP1
1351 032636  012702  000377                MOV     #000377,R2              ;SET ALL ONES WORD
1352 032642  112737  000002  034130        MOVB    #2,T15BS0               ;WRITE RAM COMMAND
1353 032650  112737  000377  034134        MOVB    #000377,T15S3           ;ALL ONES PATTERN
1354 032656  010465  000000                MOV     R4,TSDB(R5)             ;PASS PACKET ADDRESS TO CONTR.
1355 032662  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR
1356 032666  103405                        BCS     43$                     ;BR, IF OK (NO ERROR)
1357 032670  010001                        MOV     R0,R1                   ;SAVE TSSR
1361 032672                                ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032672  104456                                                        TRAP    C$ERHRD
     032674  000633                                                        .WORD   411
     032676  034136                                                        .WORD   T15SSR
     032700  012046                                                        .WORD   PKTSSR
1362 032702                         43$:   CKLOOP                          ;SCOPE LOOP
     032702  104406                                                        TRAP    C$CLP1
1363 032704  112737  000001  034130        MOVB    #1,T15BS0               ;SET UP FOR RAM READ
1364 032712  010465  000000                MOV     R4,TSDB(R5)             ;ISSUE RAM READ
1365 032716  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1366 032722  103405                        BCS     44$                     ;BR, IF OK (NO ERROR)
1367 032724  010001                        MOV     R0,R1                   ;SAVE TSSR
1371 032726                                ERRDF   ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032726  104455                                                        TRAP    C$ERDF
     032730  000634                                                        .WORD   412
     032732  034136                                                        .WORD   T15SSR
     032734  012046                                                        .WORD   PKTSSR
1372 032736  013701  033462                44$:   MOV     T15BFR+20,R1            ;PICK UP REC'D DATA
1373 032742  120102                        CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
1374 032744  001404                        BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
1378 032746                                ERRHRD  ERRNO,T15AM2,EXPBREC    ;WRITTEN DATA NOT = TO READ
```

TEST 4: RAM EXERCISER TEST

```
        032746  104456                                                          TRAP    C$ERHRD
        032750  000635                                                          .WORD   413
        032752  034212                                                          .WORD   T15AM2
        032754  015502                                                          .WORD   EXPBREC
  1379  032756                          45$:    CKLOOP                  ;SCOPE LOOP
        032756  104406                                                          TRAP    C$CLP1
  1380  032760  005303                          DEC     R3              ;DROP RAM ADDRESS POINTER
  1381  032762  020327  000377                CMP     R3,#255.        ;AT START YET
  1382  032766  001271                        BNE     40$             ;BR, IF MORE RAM TO CHECK
  1383
  1384  032770                          ENDSUB                  ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
        032770                                                                  L10051:
        032770  104403                                                          TRAP    C$ESUB
  1385
  1386  032772                          BGNSUB                  ;////////////// BEGIN SUBTEST //////////////
        032772                                                                  T4.3:
        032772  104402                                                          TRAP    C$BSUB
  1387                                  ;*
  1388                                  ;
  1389                                  ;TEST 4, SUBTEST 3
  1390                                  ;
  1391                                  ;
  1392                                  ;       THIS SUBTEST WRITES RAM WITH ALL ONES
  1393                                  ;       THEN WALKS AN ALL ZEROS WORD DOWN THROUGH MEMORY
  1394                                  ;
  1395  032774  005737  003400                  TST     SKIPT                   ;CHECK RUN MODE
  1396  033000  001402                          BEQ     10$                     ;BR, IF NO SKIP
  1397  033002  000137  033376                  JMP     50$                     ;SKIP SUBTEST
  1398  033006  004737  034522          10$:    JSR     PC,T15REST              ;RESTORE PACKET FOR WRITE CHARA
  1399  033012  004737  034574                  JSR     PC,T15RT2               ;RESTORE PACKET FOR WRT SUB SYS MEM
  1400  033016  004737  015774                  JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
  1401  033022  103405                          BCS     20$                     ;BR IF INIT WAS OK
  1405  033024  010001                          MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
  1406  033026                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
        033026  104455                                                          TRAP    C$ERDF
        033030  000636                                                          .WORD   414
        033032  003652                                                          .WORD   SFIERR
        033034  012034                                                          .WORD   SFIMSG
  1407  033036                          20$:
  1408  033036  012704  033420                  MOV     #T15PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
  1409  033042  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
  1410  033046  103405                          BCS     25$                     ;BR, IF COMMAND ISSUED OK
  1414  033050  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  1415  033052                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
        033052  104456                                                          TRAP    C$ERHRD
        033054  000637                                                          .WORD   415
        033056  005056                                                          .WORD   WRTMSG
        033060  012034                                                          .WORD   SFIMSG
  1416  033062                          25$:
  1417  033062  112737  000001  034131          MOVB    #1,T15BC1               ;SET SIZE TO 1 BYTE
  1418  033070  012704  034120                  MOV     #T15PK2,R4              ;SET NEW PACKET ADDRESS
  1419  033074  012703  000400                  MOV     #256.,R3                ;STARTING ADDRESS IN RAM
  1420  033100  112737  000002  034130          MOVB    #2,T15BS0               ;WRITE RAM COMMAND
  1421  033106  112737  000377  034134          MOVB    #377,T15S3              ;SET DATA TO 377
  1422  033114  010337  034132          30$:    MOV     R3,T15S2                ;ADDRESS TO PACKET DATA AREA
  1423  033120  010465  000000                  MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
  1424  033124  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
```

TEST 4: RAM EXERCISER TEST

```
1425 033130  103405                        BCS     33$                     ;BR, IF NO PROBLEM
1426 033132  010001                        MOV     R0,R1                   ;SAVE TSSR
1430 033134                                ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     033134  104456                                                                TRAP    C$ERHRD
     033136  000640                                                                .WORD   416
     033140  034136                                                                .WORD   T15SSR
     033142  012046                                                                .WORD   PKTSSR
1431 033144                         33$:    CKLOOP                          ;SCOPE LOOP
     033144  104406                                                                TRAP    C$CLP1
1432
1433
1434 033146  005203                        INC     R3                      ;NEXT ADDRESS
1435 033150  020327  010000                CMP     R3,#10000               ;END OF RAM MEMORY CHECK
1436 033154  001357                        BNE     30$                     ;BR, MORE RAM TO GO
1437 033156  005303                 35$:   DEC     R3                      ;SET BACK TO 7777
1438 033160  112702  000377         40$:   MOVB    #377,R2                 ;SET TO ALL ONES
1439 033164  112737  000001  034130        MOVB    #1,T15BS0               ;READ RAM COMMAND
1440 033172  010337  034132                MOV     R3,T15S2                ;ADDRESS TO BE READ TO PACKET DATA
1441 033176  010465  000000                MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
1442 033202  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1443 033206  103405                        BCS     41$                     ;BR, IF ALL IS WELL
1444 033210  010001                        MOV     R0,R1                   ;SAVE TSSR
1448 033212                                ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     033212  104456                                                                TRAP    C$ERHRD
     033214  000641                                                                .WORD   417
     033216  034136                                                                .WORD   T15SSR
     033220  012046                                                                .WORD   PKTSSR
1449 033222                         41$:    CKLOOP                          ;SCOPE LOOP
     033222  104406                                                                TRAP    C$CLP1
1450 033224  013701  033462                MOV     T15BFR+20,R1            ;PICK UP READ DATA
1451 033230  120102                        CMPB    R1,R2                   ;BOTH SHOULD BE 11111111 BINARY
1452 033232  001404                        BEQ     42$                     ;BR, IF DATA IS GOOD
1456 033234                                ERRHRD  ERRNO,T15AM3,EXPBREC    ;CHARACTERISTICS DATA NOT CORRECT
     033234  104456                                                                TRAP    C$ERHRD
     033236  000642                                                                .WORD   418
     033240  034313                                                                .WORD   T15AM3
     033242  015502                                                                .WORD   EXPBREC
1457 033244  012702  000377         42$:   MOV     #000377,R2              ;SET ALL ONES WORD
1458 033250  012737  000002  034130        MOV     #2,T15BS0               ;WRITE RAM COMMAND
1459 033256  112737  000377  034134        MOVB    #000377,T15S3           ;ALL ONES PATTERN
1460 033264  010465  000000                MOV     R4,TSDB(R5)             ;PASS PACKET ADDRESS TO CONTR.
1461 033270  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR
1462 033274  103405                        BCS     43$                     ;BR, IF OK (NO ERROR)
1463 033276  010001                        MOV     R0,R1                   ;SAVE TSSR
1467 033300                                ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     033300  104456                                                                TRAP    C$ERHRD
     033302  000643                                                                .WORD   419
     033304  034136                                                                .WORD   T15SSR
     033306  012046                                                                .WORD   PKTSSR
1468 033310                         43$:    CKLOOP                          ;SCOPE LOOP
     033310  104406                                                                TRAP    C$CLP1
1469 033312  112737  000001  034130        MOVB    #1,T15BS0               ;SET UP FOR RAM READ
1470 033320  010465  000000                MOV     R4,TSDB(R5)             ;ISSUE RAM READ
1471 033324  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1472 033330  103405                        BCS     44$                     ;BR, IF OK (NO ERROR)
1473 033332  010001                        MOV     R0,R1                   ;SAVE TSSR
1477 033334                                ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
```

TEST 4: RAM EXERCISER TEST

```
        033334  104456                                                          TRAP    C$ERHRD
        033336  000644                                                          .WORD   420
        033340  034136                                                          .WORD   T15SSR
        033342  012046                                                          .WORD   PKTSSR
1478    033344  013701  033462          44$:    MOV     T15BFR+20,R1            ;PICK UP REC'D DATA
1479    033350  120102                  CMPB    R1,R2                           ;CHECK WITH DATA WRITTEN
1480    033352  001404                  BEQ     45$                             ;BR IF OK, DATA IN = DATA OUT
1484    033354                          ERRHRD  ERRNO,T15AM2,EXPBREC            ;WRITTEN DATA NOT = TO READ
        033354  104456                                                          TRAP    C$ERHRD
        033356  000645                                                          .WORD   421
        033360  034212                                                          .WORD   T15AM2
        033362  015502                                                          .WORD   EXPBREC
1485    033364                          45$:    CKLOOP                          ;SCOPE LOOP
        033364  104406                                                          TRAP    C$CLP1
1486    033366  005303                  DEC     R3                              ;DROP RAM ADDRESS POINTER
1487    033370  020327  000377          CMP     R3,#255.                        ;AT START YET
1488    033374  001271                  BNE     40$                             ;BR, IF MORE RAM TO CHECK
1489
1490    033376                          50$:
1491    033376                          ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
                                                                                L10052:
        033376
        033376  104403                                                          TRAP    C$ESUB
1492
1493    033400  004737  016456                  JSR     PC,TSTLOOP              ;DO WE NEED TO ITERATE TEST ?
1494    033404  103002                  BCC     63$                             ;BRANCH IF NOT
1495    033406  000137  032074          JMP     T15LOOP                         ;EXECUTE AGAIN
1496    033412                          63$:    EXIT    TST                     ;ALL DONE THIS TEST
        033412  104432                                                          TRAP    C$EXIT
        033414  001216                                                          .WORD   L10047-.
1497                                    ;+
1498                                    ;LOCAL STORAGE FOR THIS TEST
1499                                    ;-
1500
1502            033420                          .=<.+10>&177770
1504    033420                          T15PACKET:                              ;COMMAND PACKET FOR TEST
1505    033420  100204                          .WORD   100204                  ;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
1506    033422  033430                          .WORD   T15DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
1507    033424  000000                          .WORD   0
1508    033426  000010                          .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
1509    033430                          T15DATA:                                ;CHARACTERISTICS DATA BLOCK
1510    033430  033442                          .WORD   T15BFR                  ;ADDRESS OF MESSAGE BUFFER
1511    033432  000000                          .WORD   0
1512    033434  000400                          .WORD   256.                    ;LENGTH OF MESSAGE BUFFER
1513    033436  000000  000000                  .WORD   0,0
1514    033442                          T15BFR: .BLKW   150.                    ;MESSAGE BUFFER
1515                                    ;
1516                                    ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
1517                                    ;
1519            034120                          .=<.+10>&177770
1521    034120                          T15PK2:
1522    034120  100206                          .WORD   100206                  ;WRITE SUB SYS MEM COMMAND, IE AND ACK
1523    034122  034130                          .WORD   T15BF2                  ;ADDRESS OF SELECT BLOCK DATA
1524    034124  000000                          .WORD   0
1525    034126  000006                          .WORD   6.                      ;SIZE OF DATA PACKET
1526
1527                                            .EVEN
1528    034130                          T15BF2:
```

TEST 4: RAM EXERCISER TEST

```
1529 034130     000                 T15BS0: .BYTE   0                       ;BSEL0 AREA
1530 034131     000                 T15BS1: .BYTE   0                       ;BSEL1 AREA
1531 034132  000000                 T15S2:  .WORD   0                       ;SEL 2 AREA
1532 034134  000000                 T15S3:  .WORD   0                       ;DATA AREA
1533
1534
1535
1536
1537                                ;+
1538                                ;LOCAL TEXT MESSAGES FOR TEST
1539                                ;-
1540
1541 034136     127     122     111 T15SSR: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
1542 034212     127     122     111 T15AM2: .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Ones Word Read Back'
1543 034313     127     122     111 T15AM3: .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Zeros Word Read Back'
1544 034415     127     122     111 T15AM4: .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On Address Test'
1545 034503     122     101     115 TST15ID:        .ASCIZ  'RAM Exerciser'
1546                                        .EVEN
1547                                ;+
1548                                ;
1549                                ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
1550                                ;WRITE SUBSYSTEM MEMORY COMMAND
1551                                ;
1552                                ;-
1553
1554 034522                         T15REST:
1555 034522                                 SAVREG                          ;SAVE THE REGISTERS
1556 034526  012701  033420                 MOV     #T15PACKET,R1           ;START OF THE PACKET
1557 034532  012721  100204                 MOV     #100204,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK, IE
1558 034536  012721  033430                 MOV     #T15DATA,(R1)+          ;ADDRESS OF CHARAISTICS DATA BLOCK
1559 034542  005021                         CLR     (R1)+                   ;EXTENDED ADDRESS
1560 034544  012721  000010                 MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
1561 034550  012721  033442                 MOV     #T15BFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
1562 034554  005021                         CLR     (R1)+
1563 034556  012721  000400                 MOV     #256.,(R1)+             ;LENGTH OF MESSAGE BUFFER
1564 034562  005021                         CLR     (R1)+
1565 034564  005011                         CLR     (R1)
1566 034566  005037  033442                 CLR     T15BFR                  ;CLEAR 1ST LOC IN MESSAGE BUFFER
1567 034572  000207                         RTS     PC                      ;RETURN
1568
1569
1570 034574                         T15RT2:
1571 034574                                 SAVREG                          ;SAVE THE REGISTERS
1572 034600  012701  034120                 MOV     #T15PK2,R1              ;START OF THE PACKET
1573 034604  012721  100206                 MOV     #100206,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK, IE
1574 034610  012721  034130                 MOV     #T15BF2,(R1)+           ;ADDRESS OF DATA BLOCK
1575 034614  005021                         CLR     (R1)+                   ;EXTENDED ADDRESS
1576 034616  012721  000006                 MOV     #6.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
1577 034622  005021                         CLR     (R1)+
1578 034624  005021                         CLR     (R1)+
1579 034626  005011                         CLR     (R1)
1580 034630  000207                         RTS     PC                      ;RETURN
1581 034632                                 ENDTST
     034632                                                     L10047:
     034632  104401                                                   TRAP    C$ETST
```

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


034634                                  BGNTST
034634                                                                          T5::
034634  012700  036712                    MOV     #TST16ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
034640  004737  016510                    JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
034644  012737  000012  002216            MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
034652                          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
```

```
                                ;         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
                                ;--
034652                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
034652                                                                                    T5.1:
034652  104402                                                                            TRAP    C$BSUB

034654                          5$:
                                ;       Write to TSSR register to soft initialize the controller
034654  004737  015774                  JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
034660  103405                          BCS     10$                     ;BR IF SOFT INIT OKAY
034662  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
034664                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
034664  104455                                                                            TRAP    C$ERDF
034666  000764                                                                            .WORD   500
034670  003652                                                                            .WORD   SFIERR
034672  012034                                                                            .WORD   SFIMSG
                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
034674  004737  040060          10$:    JSR     PC,T16REST              ;RESTORE PACKET DEFAULTS
034700  005037  002222                  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
034704  012704  040240                  MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
034710  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
034714                                  FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
034730  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
034732  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
034734                                  NEXT.ERRNO
034734                          12$:    ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
034734  104455                                                                            TRAP    C$ERDF
034736  000765                                                                            .WORD   501
034740  036762                                                                            .WORD   T16SSR
034742  012046                                                                            .WORD   PKTSSR
034744  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
034750                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
034750  104406                                                                            TRAP    C$CLP1

                                ;       If Extended Features Hardware Switch Clear then:
                                ;         (* Verify Extended Features switch can be Inverted to SET *)
                                ;         REPEAT FOR TEST PATTERNS IN TSTBLK TABLE
034752  012701  040262                  MOV     #T16BFR,R1              ;MESSAGE BUFFER ADDRESS
034756  032761  000200  000012          BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH CLEAR?
034764  001402                          BEQ     20$                     ;BR IF YES
034766  000137  035336                  JMP     200$                    ;
034772  012703  002764          20$:    MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
```

```
1689                                    ;       Do Write Subsystem Write Miscellaneous to SET Extended Features.
1690
1691 034776  004737  040220                      JSR     PC,T16SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
1692 035002  012704  040310                      MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
1693 035006  010465  000000                      MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
1694 035012  004737  016336                      JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1695 035016                                      FORCERROR       32$                     ;@@DFORCE ERROR IF FORCER=1
1696 035032  103407                              BCS     40$                     ;BR IF CARRY SET (GOOD RETURN)
1697 035034  010001                              MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1698 035036                                      NEXT.ERRNO
1699 035036                              32$:    ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FAT/  SSR FAILED TO SET
     035036  104455                                                                      TRAP    C$ERDF
     035040  000766                                                                      .WORD   502
     035042  037017                                                                      .WORD   T162SSR
     035044  012046                                                                      .WORD   PKTSSR
1700 035046  005237  002222                      INC     FATFLG                  ;SET FATAL ERROR FLAG
1701 035052                              40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035052  104406                                                                      TRAP    C$CLP1
1702
1703                                    ;         DO a WRITE CHARACTERISTICS with an extended characteristic word
1704 035054  012737  125252  002312              MOV     #125252,DATA            ;SETUP TEST DATA FOR EXTENDED WORD
1705 035062  012704  040240                      MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
1706 035066  012764  000020  000006              MOV     #16.,PKBCNT(R4)         ;STORE MESSAGE PACKET SIZE
1707 035074  013737  002312  040260              MOV     DATA,T16DATA+10         ;STORE TEST DATA IN EXTENDED WORD
1708 035102  004737  010662                      JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
1709 035106                                      FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
1710 035122  103407                              BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
1711 035124  010001                              MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1712 035126                                      NEXT.ERRNO
1713 035126                              42$:    ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     035126  104455                                                                      TRAP    C$ERDF
     035130  000767                                                                      .WORD   503
     035132  036762                                                                      .WORD   T16SSR
     035134  012046                                                                      .WORD   PKTSSR
1714 035136  005237  002222                      INC     FATFLG                  ;SET FATAL ERROR FLAG
1715 035142                              50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035142  104406                                                                      TRAP    C$CLP1
1716                                    ;         If the TSBA Address Register NOT= Expected  Then Print Error
1717 035144  016501  000000                      MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
1718 035150  012702  040262                      MOV     #T16BFR,R2              ;START OF THE DATA BUFFER
1719 035154  062702  000020              62$:    ADD     #16.,R2                 ;EXPECTED CONTENTS OF TSBA
1720 035160                                      FORCERROR       72$,NOTSSR              ;@@DFORCE ERROR IF FORCER=1
1721 035170  020102                              CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1722 035172  001404                              BEQ     80$                     ;ERROR IF NOT EQUAL
1723 035174                                      NEXT.ERRNO
1724 035174                              72$:    ERRHRD  ERRNO,T16TSBA,EXPREC    ;PRINT THE ERROR & EXPD/RECV
     035174  104456                                                                      TRAP    C$ERHRD
     035176  000770                                                                      .WORD   504
     035200  037130                                                                      .WORD   T16TSBA
     035202  015474                                                                      .WORD   EXPREC
1725 035204                              80$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035204  104406                                                                      TRAP    C$CLP1
1726                                    ;         Compare the controller ram to the extended characteristic word
1727                                    ;         If Data word in controller ram NOT= to word sent  Then Print Error
1728 035206  012704  040250                      MOV     #T16DATA,R4             ;GET CHARACTERISTIC DATA ADDRESS
1729 035212  004737  011224                      JSR     PC,CKRAM2               ;DOES RAM DATA EQUAL DATA SENT?
1730 035216                                      FORCERROR       92$                     ;@@DFORCE ERROR IF FORCER=1
```

```
1731 035232  103404                        BCS     100$                    ;BR IF YES
1732 035234                                NEXT.ERRNO
1733 035234                          92$:  ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
     035234  104456                                                        TRAP    C$ERHRD
     035236  000771                                                        .WORD   505
     035240  004745                                                        .WORD   PKTRAM
     035242  015510                                                        .WORD   RAMERR
1734 035244                          100$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035244  104406                                                        TRAP    C$CLP1
1735                                 ;       If Message Buffer Data Length NOT= 12.  Then Print Error
1736 035246  012702  040262                MOV     #T16BFR,R2              ;GET MESSAGE BUFFER ADDRESS
1737 035252  016201  000002                MOV     2(R2),R1                ;GET RECV DATA FIELD LENGTH
1738 035256  012702  000014                MOV     #12.,R2                 ;GET EXPD DATA FIELD LENGTH
1739 035262                                FORCERROR       112$,NOTSSR             ;@@DFORCE ERROR IF FORCER=1
1740 035272  020102                        CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1741 035274  001404                        BEQ     120$                    ;ERROR IF NOT EQUAL
1742 035276                                NEXT.ERRNO
1743 035276                          112$: ERRHRD  ERRNO,T16LEN,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     035276  104456                                                        TRAP    C$ERHRD
     035300  000772                                                        .WORD   506
     035302  037232                                                        .WORD   T16LEN
     035304  015474                                                        .WORD   EXPREC
1744 035306                          120$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035306  104406                                                        TRAP    C$CLP1
1745
1746 035310  004737  015774                JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
1747 035314  103405                        BCS     125$                    ;BR IF SOFT INIT OKAY
1748 035316  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1749 035320                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     035320  104455                                                        TRAP    C$ERDF
     035322  000772                                                        .WORD   506
     035324  003652                                                        .WORD   SFIERR
     035326  012034                                                        .WORD   SFIMSG
1750 035330                          125$: CKLOOP                          ;LOOP IF SELECTED
     035330  104406                                                        TRAP    C$CLP1
1751 035332  000137  035516                JMP     300$                    ;
1752
1753                                 ;    (* Verify Extended Features switch can be Inverted to CLEAR *)
1754 035336                          200$:
1755                                 ;       Do Write Subsystem Write Miscellaneous to CLEAR Extended Features.
1756 035336  004737  040220                JSR     PC,T16SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
1757 035342  012704  040310                MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
1758 035346  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
1759 035352  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1760 035356                                FORCERROR       232$                    ;@@DFORCE ERROR IF FORCER=1
1761 035372  103407                        BCS     240$                    ;BR IF CARRY SET (GOOD RETURN)
1762 035374  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1763 035376                                NEXT.ERRNO
1764 035376                          232$: ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     035376  104455                                                        TRAP    C$ERDF
     035400  000773                                                        .WORD   507
     035402  037017                                                        .WORD   T162SSR
     035404  012046                                                        .WORD   PKTSSR
1765 035406  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
1766 035412                          240$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035412  104406                                                        TRAP    C$CLP1
1767
```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```
1768                                    ;       DO a WRITE CHARACTERISTICS without an extended charact.ristic word
1769 035414  012704  040240                       MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
1770 035420  012764  000016  000006               MOV     #14.,PKBCNT(R4)         ;STORE MESSAGE PACKET SIZE
1771 035426  004737  010662                       JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
1772 035432                                       FORCERROR       242$                    ;@@DFORCE ERROR IF FORCER=1
1773 035446  103407                               BCS     250$                    ;BR IF CARRY SET (GOOD RETURN)
1774 035450  010001                               MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1775 035452                                       NEXT.ERRNO
1776 035452                             242$:     ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     035452  104455                                                                       TRAP    C$ERDF
     035454  000774                                                                       .WORD   508
     035456  036762                                                                       .WORD   T16SSR
     035460  012046                                                                       .WORD   PKTSSR
1777 035462  005237  002222                       INC     FATFLG                  ;SET FATAL ERROR FLAG
1778 035466                             250$:     CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035466  104406                                                                       TRAP    C$CLP1
1779                                    ;       If Message Buffer Data Length NOT= 10.  Then Print Error
1780 035470  013701  040264                       MOV     T16BFR+2,R1             ;GET RECV DATA FIELD LENGTH
1781 035474  012702  000012                       MOV     #10.,R2                 ;GET EXPD DATA FIELD LENGTH
1782 035500  020102                               CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1783 035502  001404                               BEQ     270$                    ;ERROR IF NOT EQUAL
1784 035504                                       NEXT.ERRNO
1785 035504                             262$:     ERRHRD  ERRNO,T16LEN,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     035504  104456                                                                       TRAP    C$ERHRD
     035506  000775                                                                       .WORD   509
     035510  037232                                                                       .WORD   T16LEN
     035512  015474                                                                       .WORD   EXPREC
1786 035514                             270$:     CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035514  104406                                                                       TRAP    C$CLP1
1787
1788
1789                                    ;       (* Verify Function Reject when Message Buffer 21-19 are non-zero *)
1790                                    ;       Write to TSSR register to soft initialize the controller
1791 035516                             300$:
1792                                    ;       REPEAT FOR MESSAGE BUFFER ADDRESS bits <21:19> FROM 0 TO 7
1793 035516  012737  000001  002312     320$:     MOV     #1,DATA                 ;START AT BITS<21:19>=001
1794                                    ;       DO a WRITE CHARACTERISTICS with a message address bit<21:19> non-zero
1795 035524                             325$:
1796 035524  012704  040240                       MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
1797 035530  012764  000016  000006               MOV     #14.,PKBCNT(R4)         ;STORE MESSAGE PACKET SIZE
1798 035536  013700  002312                       MOV     DATA,R0                 ;GET TEST DATA
1799         000003                               .REPT   3
1800                                              ASL     R0                      ;SHIFT INTO BITS 21:19
1801                                              .ENDR
1802 035550  010037  040252                       MOV     R0,T16DATA+2            ;STORE BUFFER ADDRESS BITS 21:19
1803 035554  010465  000000                       MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
1804 035560  004737  016250                       JSR     PC,WAITF                ;WAIT FOR SSR
1805 035564                                       FORCERROR       342$                    ;@@DFORCE ERROR IF FORCER=1
1806 035600  103407                               BCS     350$                    ;BR IF CARRY SET (GOOD RETURN)
1807 035602  010001                               MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1808 035604                                       NEXT.ERRNO
1809 035604                             342$:     ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     035604  104455                                                                       TRAP    C$ERDF
     035606  000776                                                                       .WORD   510
     035610  036762                                                                       .WORD   T16SSR
     035612  012046                                                                       .WORD   PKTSSR
1810 035614  005237  002222                       INC     FATFLG                  ;SET FATAL ERROR FLAG
```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```
1811 035620                            350$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035620  104406                                                                    TRAP    C$CLP1
1812
1813                                    ;       If TSSR termination code NOT= Function Reject  Then Print Error
1814 035622  016501  000002                    MOV     TSSR(R5),R1             ;GET RECV TSSR
1815 035626  010102                            MOV     R1,R2                   ;COPY RECV TSSR
1816 035630  042702  000016                    BIC     #TERCLS,R2              ;CLEAR TC<2:0> EXPD
1817 035634  052702  000006                    BIS     #TSREJ,R2               ;SET EXPD TC<2:0>= FUNCTION REJECT
1818 035640                                    FORCERROR       352$,NOTSSR             ;@@FORCE ERROR IF FORCER=1
1819 035650  020102                            CMP     R1,R2                   ;EXPD EQUAL RECV?
1820 035652  001404                            BEQ     360$                    ;BR IF YES
1821 035654                                    NEXT.ERRNO
1822 035654                            352$:   ERRHRD  ERRNO,T16REJ,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     035654  104456                                                                    TRAP    C$ERHRD
     035656  000777                                                                    .WORD   511
     035660  037344                                                                    .WORD   T16REJ
     035662  012046                                                                    .WORD   PKTSSR
1823 035664                            360$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035664  104406                                                                    TRAP    C$CLP1
1824 035666                                    FORCLXIT        370$
1825 035676  005237  002312                    INC     DATA                    ;GET NEXT TST PATTERN
1826 035702  023727  002312  000007            CMP     DATA,#7                 ;DONE ALL DATA?
1827 035710  101002                            BHI     370$                    ;BR IF YES
1828 035712  000137  035524                    JMP     325$                    ;DO ANOTHER TEST PATTERN
1829                                    ;       END-REPEAT
1830 035716                            370$:
1831 035716                                    ENDSUB                          ;////////////// END SUBTEST //////////
     035716                                                                            L10054:
     035716  104403                                                                    TRAP    C$ESUB
1832
1833 035720  005737  002222                    TST     FATFLG                  ;ANY FATAL ERRORS ?
1834 035724  001402                            BEQ     460$                    ;BRANCH IF NOT
1835 035726  004737  017202                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
1836 035732                            460$:
1837
1838
1839
1840
1841
1842                                            .SBTTL  TEST  5: SUBTEST 2: VERIFY TIMERS A,B
1843
1844                                    ;**
1845                                    ; TEST 5: SUBTEST 2:
1846                                    ;
1847                                    ; SUBTEST DESCRIPTION:
1848                                    ;
1849                                    ;       This subtest verifies that timers A,B can be reset
1850                                    ;       and that Timer A is twice the frequency of Timer B.
1851                                    ;       Timer A has a period of 25 microseconds and Timer B
1852                                    ;        as a period of 50 microseconds. The timers are
1853                                    ;       checked at 1, 28, 53, and 78 micoseconds.
1854                                    ;
1855                                    ; TEST STEPS:
1856                                    ;
1857                                    ;
1858                                    ;       Write to TSSR register to soft initialize the controller
1859                                    ;       Do WRITE CHARACTERISTICS to setup a Message Buffer
```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```
1860                                    ;       (* Verify Timers A,B after RESET TIMER with 0 microsecond delay *)
1861                                    ;       Do a Write Control RESET TIMER with 1 microsecond delay
1862                                    ;       Do a Write Subsystem READ STATUS
1863                                    ;       If Timer A NOT= 0  Then Print Error
1864                                    ;       If Timer B NOT= 0  Then Print Error
1865                                    ;       (* Verify Timers A,B after RESET TIMER with 28 microsecond delay *)
1866                                    ;       Do a Write Control RESET TIMER with 28 microsecond delay
1867                                    ;       If Timer A NOT= 1  Then Print Error
1868                                    ;       If Timer B NOT= 1  Then Print Error
1869                                    ;       Do a Write Control RESET TIMER with 53 microsecond delay
1870                                    ;       If Timer A NOT= 0  Then Print Error
1871                                    ;       If Timer B NOT= 1  Then Print Error
1872                                    ;       Do a Write Control RESET TIMER with 78 microsecond delay
1873                                    ;       If Timer A NOT= 1  Then Print Error
1874                                    ;       If Timer B NOT= 0  Then Print Error
1875                                    ;--
1876 035732                                     BGNSUB                          ;////////////// BEGIN SUBTEST //////////
      035732                                                                                      T5.2:
      035732  104402                                                                                    TRAP    C$BSUB
1877                                    ;       Write to TSSR register to soft initialize the controller
1878 035734                             5$:
1879 035734  004737  015774                     JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
1880 035740  103405                             BCS     10$                     ;BR IF SOFT INIT OKAY
1881 035742  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1882 035744                                     ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      035744  104455                                                                                    TRAP    C$ERDF
      035746  000777                                                                                    .WORD   511
      035750  003652                                                                                    .WORD   SFIERR
      035752  012034                                                                                    .WORD   SFIMSG
1883                                    ;       Do WRITE CHARACTERISTICS to setup a Message Buffer
1884 035754  004737  040060             10$:    JSR     PC,T16REST              ;RESTORE PACKET DEFAULTS
1885 035760  005037  002222                     CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1886 035764  012704  040240                     MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
1887 035770  012764  000010  000006             MOV     #8.,PKBCNT(R4)          ;MESSAGE PACKET SIZE NO EXTEND
1888 035776  004737  010662                     JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
1889 036002                                     FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
1890 036016  103407                             BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1891 036020  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1892 036022                                     NEXT.ERRNO
1893 036022                             12$:    ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      036022  104455                                                                                    TRAP    C$ERDF
      036024  001000                                                                                    .WORD   512
      036026  036762                                                                                    .WORD   T16SSR
      036030  012046                                                                                    .WORD   PKTSSR
1894 036032  005237  002222                     INC     FATFLG                  ;SET FATAL ERROR FLAG
1895 036036                             15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      036036  104406                                                                                    TRAP    C$CLP1
1896
1897                                    ;       (* Verify Timers A,B after RESET TIMER with 1 microsecond delay *)
1898                                    ;       Do a Write Control RESET TIMER with 1 microsecond delay
1899 036040  012700  000001                     MOV     #MS.RSD,R0              ;RESET TIMER COMMAND
1900 036044  013701  036702                     MOV     T16D01,R1               ;1 MICROSECOND DELAY
1901 036050  004737  040172                     JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
1902 036054  012704  040310                     MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
1903 036060  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
1904 036064  004737  016336                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1905 036070                                     FORCERROR       32$                     ;@@DFORCE ERROR IF FORCER=1
```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```
1906 036104  103407                    BCS     40$                     ;BR IF CARRY SET (GOOD RETURN)
1907 036106  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1908 036110                            NEXT.ERRNO
1909 036110                    32$:    ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     036110  104455                                                            TRAP    C$ERDF
     036112  001001                                                            .WORD   513
     036114  037017                                                            .WORD   T162SSR
     036116  012046                                                            .WORD   PKTSSR
1910 036120  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
1911 036124                    40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036124  104406                                                            TRAP    C$CLP1
1912                           ;       If Timer A NOT= 0  Then Print Error
1913                           ;       If Timer B NOT= 0  Then Print Error
1914 036126  005002                    CLR     R2                      ;INIT EXPD
1915 036130  042702  000010            BIC     #S2.ATIM,R2             ;TIMER A EXPD=0
1916 036134  042702  000004            BIC     #S2.BTIM,R2             ;TIMER B EXPD=0
1917 036140  012700  040302            MOV     #T16BFSTA,R0            ;GET RECV READ STATUS
1918 036144  016001  000002            MOV     2(R0),R1                ;GET RECV BYTE 2
1919 036150  042701  177763            BIC     #^C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
1920 036154                            FORCERROR       72$,NOTSSR      ;@@D
1921 036164  020201                    CMP     R2,R1                   ;EXPD EQUAL RECV?
1922 036166  001404                    BEQ     80$                     ;BR IF YES
1923 036170                            NEXT.ERRNO
1924 036170                    72$:    ERRHRD  ERRNO,T16T01,TIMEXP     ;REPORT ERROR
     036170  104456                                                            TRAP    C$ERHRD
     036172  001002                                                            .WORD   514
     036174  037461                                                            .WORD   T16T01
     036176  015552                                                            .WORD   TIMEXP
1925 036200                    80$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036200  104406                                                            TRAP    C$CLP1
1926
1927                           ;       Do a Write Control RESET TIMER with 28 microsecond delay
1928 036202  012700  000001            MOV     #MS.RSD,R0              ;RESET TIMER COMMAND
1929 036206  013701  036704            MOV     T16D28,R1               ;28 MICROSECOND DELAY
1930 036212  004737  040172            JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
1931 036216  012704  040310            MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
1932 036222  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
1933 036226  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1934 036232                            FORCERROR       112$            ;@@DFORCE ERROR IF FORCER=1
1935 036246  103407                    BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
1936 036250  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1937 036252                            NEXT.ERRNO
1938 036252                    112$:   ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     036252  104455                                                            TRAP    C$ERDF
     036254  001003                                                            .WORD   515
     036256  037017                                                            .WORD   T162SSR
     036260  012046                                                            .WORD   PKTSSR
1939 036262  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
1940 036266                    120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036266  104406                                                            TRAP    C$CLP1
1941                           ;       If Timer A NOT= 1  Then Print Error
1942                           ;       If Timer B NOT= 1  Then Print Error
1943 036270  005002                    CLR     R2                      ;INIT EXPD
1944 036272  052702  000010            BIS     #S2.ATIM,R2             ;TIMER A EXPD=1
1945 036276  052702  000004            BIS     #S2.BTIM,R2             ;TIMER B EXPD=1
1946 036302  012700  040302            MOV     #T16BFSTA,R0            ;GET RECV READ STATUS
1947 036306  016001  000002            MOV     2(R0),R1                ;GET RECV BYTE 2
```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```
1948 036312  042701  177763              BIC     @#C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
1949 036316                              FORCERROR       172$,NOTSSR     ;@@D
1950 036326  020201                      CMP     R2,R1                   ;EXPD EQUAL RECV?
1951 036330  001404                      BEQ     180$                    ;BR IF YES
1952 036332                              NEXT.ERRNO
1953 036332                  172$:       ERRHRD  ERRNO,T16T28,TIMEXP     ;REPORT ERROR
     036332  104456                                                              TRAP    C$ERHRD
     036334  001004                                                              .WORD   516
     036336  037560                                                              .WORD   T16T28
     036340  015552                                                              .WORD   TIMEXP
1954 036342                  180$:       CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036342  104406                                                              TRAP    C$CLP1
1955
1956                         ;           Do a Write Control RESET TIMER with 53 microsecond delay
1957 036344  012700  000001              MOV     @#MS.RSD,R0             ;RESET TIMER COMMAND
1958 036350  013701  036706              MOV     T16D53,R1               ;53 MICROSECOND DELAY
1959 036354  004737  040172              JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
1960 036360  012704  040310              MOV     @#T16PK2,R4             ;GET WRITE SUBSYSTEM COMMAND PACKET
1961 036364  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
1962 036370  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1963 036374                              FORCERROR       212$            ;@@DFORCE ERROR IF FORCER=1
1964 036410  103407                      BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
1965 036412  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1966 036414                              NEXT.ERRNO
1967 036414                  212$:       ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     036414  104455                                                              TRAP    C$ERDF
     036416  001005                                                              .WORD   517
     036420  037017                                                              .WORD   T162SSR
     036422  012046                                                              .WORD   PKTSSR
1968 036424  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
1969 036430                  220$:       CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036430  104406                                                              TRAP    C$CLP1
1970                         ;           If Timer A NOT= 0  Then Print Error
1971                         ;           If Timer B NOT= 1  Then Print Error
1972 036432  005002                      CLR     R2                      ;INIT EXPD
1973 036434  042702  000010              BIC     @#S2.ATIM,R2            ;TIMER A EXPD=0
1974 036440  052702  000004              BIS     @#S2.BTIM,R2            ;TIMER B EXPD=1
1975 036444  012700  040302              MOV     @#T16BFSTA,R0           ;GET RECV READ STATUS
1976 036450  016001  000002              MOV     2(R0),R1                ;GET RECV BYTE 2
1977 036454  042701  177763              BIC     @#C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
1978 036460                              FORCERROR       272$,NOTSSR     ;@@D
1979 036470  020201                      CMP     R2,R1                   ;EXPD EQUAL RECV?
1980 036472  001404                      BEQ     280$                    ;BR IF YES
1981 036474                              NEXT.ERRNO
1982 036474                  272$:       ERRHRD  ERRNO,T16T53,TIMEXP     ;REPORT ERROR
     036474  104456                                                              TRAP    C$ERHRD
     036476  001006                                                              .WORD   518
     036500  037660                                                              .WORD   T16T53
     036502  015552                                                              .WORD   TIMEXP
1983 036504                  280$:       CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036504  104406                                                              TRAP    C$CLP1
1984                         ;           Do a Write Control RESET TIMER with 78 microsecond delay
1985 036506  012700  000001              MOV     @#MS.RSD,R0             ;RESET TIMER COMMAND
1986 036512  013701  036710              MOV     T16D78,R1               ;78 MICROSECOND DELAY
1987 036516  004737  040172              JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
1988 036522  012704  040310              MOV     @#T16PK2,R4             ;GET WRITE SUBSYSTEM COMMAND PACKET
1989 036526  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```
1990 036532  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1991 036536                                FORCERROR       312$                    ;@@DFORCE ERROR IF FORCER=1
1992 036552  103407                        BCS     320$                    ;BR IF CARRY SET (GOOD RETURN)
1993 036554  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1994 036556                                NEXT.ERRNO
1995 036556                        312$:   ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     036556  104455                                                                TRAP    C$ERDF
     036560  001007                                                                .WORD   519
     036562  037017                                                                .WORD   T162SSR
     036564  012046                                                                .WORD   PKTSSR
1996 036566  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
1997 036572                        320$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036572  104406                                                                TRAP    C$CLP1
1998                               ;       If Timer A NOT= 1  Then Print Error
1999                               ;       If Timer B NOT= 0  Then Print Error
2000 036574  005002                        CLR     R2                      ;INIT EXPD
2001 036576  052702  000010                BIS     #S2.ATIM,R2             ;TIMER A EXPD=1
2002 036602  042702  000004                BIC     #S2.BTIM,R2             ;TIMER B EXPD=0
2003 036606  012700  040302                MOV     #T16BFSTA,R0            ;GET RECV READ STATUS
2004 036612  016001  000002                MOV     2(R0),R1                ;GET RECV BYTE 2
2005 036616  042701  177763                BIC     #^C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
2006 036622                                FORCERROR       372$,NOTSSR     ;@@D
2007 036632  020201                        CMP     R2,R1                   ;EXPD EQUAL RECV?
2008 036634  001404                        BEQ     380$                    ;BR IF YES
2009 036636                                NEXT.ERRNO
2010 036636                        372$:   ERRHRD  ERRNO,T16T78,TIMEXP     ;REPORT ERROR
     036636  104456                                                                TRAP    C$ERHRD
     036640  001010                                                                .WORD   520
     036642  037760                                                                .WORD   T16T78
     036644  015552                                                                .WORD   TIMEXP
2011 036646                        380$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036646  104406                                                                TRAP    C$CLP1
2012
2013 036650                                ENDSUB                          ;////////////// END SUBTEST //////////
     036650                                                                     L10055:
     036650  104403                                                                TRAP    C$ESUB
2014
2015 036652  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
2016 036656  001402                        BEQ     460$                    ;BRANCH IF NOT
2017 036660  004737  017202                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2018 036664  004737  016456        460$:   JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS?
2019 036670  103002                        BCC     465$                    ;BR IF NO
2020 036672  000137  034652                JMP     T16LOOP                 ;LOOP UNTIL ITERATIONS DONE
2021 036676                        465$:
2022
2023
2024 036676                                EXIT    TST                     ;////////// EXIT TEST //////////
     036676  104432                                                                TRAP    C$EXIT
     036700  001524                                                                .WORD   L10053-.
2025
2026
2027                                       ;+
2028                               ;LOCAL STORAGE FOR THIS TEST
2029                               ;-
2030 036702  000001                T16D01:         .WORD   1               ;1 MICROSECOND DELAY (ACTUALLY .8 MIC)
2031 036704  000040                T16D28:         .WORD   40              ;28 MICROSECOND DELAY (.8 MICROS PER)
2032 036706  000076                T16D53:         .WORD   76              ;53 MICROSECOND
```

```
2033 036710  000142                  T16D78:         .WORD   142             ;78 MICROSECOND
2034                                 ;+
2035                                 ;LOCAL TEXT MESSAGES FOR TEST
2036                                 ;-
2037
2038 036712      105     170     164 TST16ID:        .ASCIZ  'Extended Features Switch and Timers A,B'
2039 036762      127     122     111 T16SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
2040 037017      127     122     111 T162SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
2041 037063      127     122     111 T163SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
2042 037130      102     165     163 T16TSBA:.ASCIZ  'Bus Address Register (TSBA) Incorrect after Write Characteristics'
2043 037232      104     141     164 T16LEN:.ASCIZ   'Data Field Length in Message Buffer Incorrect after Write Characteristics'
2044 037344      124     123     123 T16REJ:.ASCIZ   'TSSR Function Reject Not Returned When Non-Existent Buffer Address Specifie
d'
2045 037461      124     151     155 T16T01:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 1 microsecond Delay'
2046 037560      124     151     155 T16T28:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 28 microsecond Delay'
2047 037660      124     151     155 T16T53:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 53 microsecond Delay'
2048 037760      124     151     155 T16T78:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 78 microsecond Delay'
2049                                                 .EVEN
2050
2051                                 ;+
2052                                 ; SET DEFAULT PACKET
2053                                 ;-
2054 040060                          T16REST:
2055 040060  012700  040240                 MOV     #T16PACKET,R0           ;PACKET ADDRESS
2056 040064  012720  100004                 MOV     #100004,(R0)+           ;WRITE CHARACTERISTICS WITH ACK
2057 040070  012720  040250                 MOV     #T16DATA,(R0)+          ;ADDRESS OF CHAR DATA BLOCK
2058 040074  005020                         CLR     (R0)+                   ;EXTENDED ADDRESS
2059 040076  012720  000012                 MOV     #10.,(R0)+              ;SIZE OF MESSAGE PACKET
2060 040102  012720  040262                 MOV     #T16BFR,(R0)+           ;MESSAGE BUFFER ADDRESS
2061 040106  005020                         CLR     (R0)+                   ;CLEAR EXTENDED BUFFER ADDRESS
2062 040110  012720  000024                 MOV     #20.,(R0)+              ;LENGTH OF MESSAGE BUFFER
2063 040114  005020                         CLR     (R0)+                   ;CLEAR ESS,ENB,EAI,ERI
2064 040116  005010                         CLR     (R0)                    ;CLEAR EXTENDED FEATURES WORD
2065 040120  005037  040262                 CLR     T16BFR                  ;CLEAR 1ST LOCATION IN MESSAGE BUFFER
2066 040124  000207                         RTS     PC                      ;
2067
2068                                 ;+
2069                                 ; CLEAR MESSAGE BUFFER
2070                                 ;-
2071 040126                          T16CLRBUF:                              ;
2072 040126                                 SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
2073 040132  012701  040262                 MOV     #T16BFR,R1              ;GET MESSAGE BUFFER ADDRESS
2074 040136  012702  000026                 MOV     #T16BEND-T16BFR,R2      ;SIZE OF MESSAGE BUFFER IN BYTES
2075 040142  105021                  10$:    CLRB    (R1)+                   ;CLEAR A BYTE
2076 040144  005302                         DEC     R2                      ;DONE?
2077 040146  003375                         BGT     10$                     ;BR IF NO
2078 040150  000207                         RTS     PC                      ;RETURN
2079
2080                                 ;+
2081                                 ; SETUP T16PK2 PACKET FOR READ STATUS
2082                                 ;-
2083 040152                          T16SRD:
2084 040152  004737  040126                 JSR     PC,T16CLRBUF            ;CLEAR MESSAGE BUFFER
2085 040156  012700  040320                 MOV     #T16DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
2086 040162  112720  000005                 MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
2087 040166  105010                         CLRB    (R0)                    ;CLEAR BSEL1
2088 040170  000207                         RTS     PC                      ;RETURN
2089
```

```
2090
2091                                    ;+
2092                                    ; SETUP T16PK2 PACKET FOR WRITE MISC.
2093                                    ;
2094                                    ; INPUT:
2095                                    ;       R0      CONTAINS WRITE MISC FUNCTION CODE (BSEL1)
2096                                    ;       R1      CONTAINS DELAY (TIMES 800 NS) FOR BSEL2
2097                                    ;-
2098 040172                            T16WMISC:
2099 040172                                    SAVREG                          ;SAVE R1-R5 UNITL NEXT RETURN
2100 040176  004737  040126                    JSR     PC,T16CLRBUF            ;CLEAR MESSAGE BUFFER
2101 040202  012702  040320                    MOV     #T16DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
2102 040206  112722  000010                    MOVB    #PW.WMISC,(R2)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
2103 040212  110022                            MOVB    R0,(R2)+                ;STORE WRITE MISC CODE IN BSEL1
2104 040214  110112                            MOVB    R1,(R2)                 ;STORE DELAY (RESET TIMER) IN BSEL2
2105 040216  000207                            RTS     PC                      ;RETURN
2106                                    ;+
2107                                    ; SETUP T16PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
2108                                    ;-
2109 040220                            T16SEXT:
2110 040220  012700  040320                    MOV     #T16DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
2111 040224  112720  000010                    MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
2112 040230  112710  000200                    MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
2113 040234  000207                            RTS     PC                      ;RETURN
2114
2115
2116
2117
2119         040240                                .=<.+10>&177770
2121                                    ;
2122                                    ;WRITE CHARACTERISTICS COMMAND PACKET
2123                                    ;
2124 040240                            T16PACKET:                              ;COMMAND PACKET FOR TEST
2125 040240  100004                            .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
2126 040242  040250                            .WORD   T16DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
2127 040244  000000                            .WORD   0
2128 040246  000012                            .WORD   10.                     ;MESSAGE PACKET SIZE
2129
2130 040250                            T16DATA:                                ;CHARACTERISTICS DATA BLOCK
2131 040250  040262                            .WORD   T16BFR                  ;ADDRESS OF MESSAGE BUFFER
2132 040252  000000                            .WORD   0
2133 040254  000024                            .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
2134 040256  000000                            .WORD   0                       ;ESS,ENB,EAI,ERI
2135 040260  000000                            .WORD   0                       ;EXTENDED FEATURES WORD
2136
2137
2138                                    ;MESSAGE BUFFER
2139
2140 040262                            T16BFR:                                 ;BEGIN MESSAGE BUFFER
2141 040262  000000                            .WORD   0                       ;MESSAGE TYPE
2142 040264  000000                            .WORD   0                       ;DATA FIELD LENGTH
2143 040266  000000                            .WORD   0                       ;RBPCR
2144 040270  000000                            .WORD   0                       ;XST0
2145 040272  000000                            .WORD   0                       ;XST1
2146 010274  000000                            .WORD   0                       ;XST2
2147 040276  000000                            .WORD   0                       ;XST3
2148 040300  000000                            .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```
2149 040302                        T16BFSTA: .BLKB 6.                      ;READ STATUS AND WRITE FIFO BUFFER
2150 040310                        T16BEND:                                ;END OF MESSAGE BUFFER
2151                               ;
2152                               ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
2153                               ;
2157 040310                        T16PK2:
2158 040310  100006                       .WORD   P.WRTSUB!P.ACK           ;WRITE SUBSYSTEM WITH ACK
2159 040312  040320                       .WORD   T16DT2                   ;LOW ADDRESS OF DATA BLOCK
2160 040314  000000                       .WORD   0                        ;HIGH ADDRESS OF DATA BLOCK
2161 040316  000012                       .WORD   10.                      ;MINIMUM MESSAGE PACKET SIZE
2162
2163 040320                        T16DT2:                                 ;DATA BLOCK
2164 040320     000                       .BYTE   0                        ;BSEL0
2165 040321     000                       .BYTE   0                        ;BSEL1
2166 040322  000000                       .WORD   0                        ;SEL2
2167 040324                               .BLKB   64.                      ;WRITE FIFO DATA OUTPUT BUFFER
2168
2169
2170 040424                               ENDTST
     040424                                                          L10053:
     040424  104401                                                        TRAP    C$ETST
```

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


040426                                  BGNTST
040426                                                                  T6::
040426  012700  046656                  MOV     #TST17ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
040432  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
040436  012737  000012  002216          MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
040444  004737  017274                  JSR     PC,KTOFF                ;SHUT OFF MEMORY MANAGEMENT
040450  005037  003134                  CLR     KTENABLE                ;REALLY SHUT DOWN KT-11
040454                          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
```

```
2232                                    ;--
2233 040454                             BGNSUB                          ;////////////// BEGIN SUBTEST //////////
     040454                                                                     T6.1:
     040454  104402                                                                TRAP    C$BSUB
2234
2235                                    ;       Write to TSSR register to soft initialize the controller
2236 040456                             5$:
2237 040456  004737  015774            JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
2238 040462  103405                    BCS     10$                     ;BR IF SOFT INIT OKAY
2239 040464  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2240 040466                            ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     040466  104455                                                                TRAP    C$ERDF
     040470  001130                                                                .WORD   600
     040472  003652                                                                .WORD   SFIERR
     040474  012034                                                                .WORD   SFIMSG
2241                                    ;       Do a WRITE CHARACTERISTICS to setup a message buffer
2242 040476  005037  002222            10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
2243 040502  012704  050250            MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
2244 040506  004737  010662            JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
2245 040512                            FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
2246 040526  103407                    BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
2247 040530  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2248 040532                            NEXT.ERRNO
2249 040532                            42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     040532  104455                                                                TRAP    C$ERDF
     040534  001131                                                                .WORD   601
     040536  046675                                                                .WORD   T17SSR
     040540  012046                                                                .WORD   PKTSSR
2250 040542  005237  002222            50$:    INC     FATFLG                  ;SET FATAL ERROR FLAG
2251 040546                            CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     040546  104406                                                                TRAP    C$CLP1
2252
2253                                    ;       Do a Write Subsystem READ STATUS
2254 040550  004737  050034            JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2255 040554  012704  050420            MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2256 040560  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2257 040564  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2258 040570                            FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
2259 040604  103407                    BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
2260 040606  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2261 040610                            NEXT.ERRNO
2262 040610                            62$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     040610  104455                                                                TRAP    C$ERDF
     040612  001132                                                                .WORD   602
     040614  046776                                                                .WORD   T173SSR
     040616  012046                                                                .WORD   PKTSSR
2263 040620  005237  002222            70$:    INC     FATFLG                  ;SET FATAL ERROR FLAG
2264 040624                            CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     040624  104406                                                                TRAP    C$CLP1
2265                                    ;       Set WORDS 0-7  of expd message buffer = to recv since not testing
2266 040626  004737  050216            JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2267 040632  012701  046452            MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2268 040636  012702  050312            MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2269 040642  012221                    MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2270 040644  011211                    MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2271 040646  052711  000020            BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= TRUE
2272 040652  042711  000040            BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= FALSE
```

TEST 6: SUBTEST 1: FIFO INITIALIZE STATUS TEST

```
2273 040656  042711  000200                    BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = FALSE
2274                                    ;       If Input Ready NOT=1 then Print Error
2275                                    ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2276 040662  005000                             CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2277 040664  012701  050272                     MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2278 040670  012702  046432                     MOV     #T17EXP,R2              ;EXPD ADDRESS
2279 040674  012703  000024                     MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2280 040700  004737  011500                     JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2281 040704                                     FORCERROR       82$,NOTSSR      ;ØØD
2282 040714  103404                             BCS     90$                     ;BR IF YES
2283 040716                                     NEXT.ERRNO
2284 040716                             82$:    ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
     040716  104456                                                                     TRAP    C$ERHRD
     040720  001133                                                                     .WORD   603
     040722  047215                                                                     .WORD   T171CMP
     040724  012350                                                                     .WORD   MSGSTAT
2285 040726                             90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     040726  104406                                                                     TRAP    C$CLP1
2286
2287 040730                                     ENDSUB                          ;////////////// END SUBTEST //////////
     040730                                                                   L10057:
     040730  104403                                                                     TRAP    C$ESUB
2288
2289 040732  005737  002222                     TST     FATFLG                  ;ANY FATAL ERRORS ?
2290 040736  001402                             BEQ     160$                    ;BRANCH IF NOT
2291 040740  004737  017202                     JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2292 040744                             160$:
2293
2294                                             .SBTTL  TEST  6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST
2295
2296                                    ;++
2297                                    ; TEST 6: SUBTEST 2:
2298                                    ;
2299                                    ; SUBTEST DESCRIPTION:
2300                                    ;
2301                                    ;       This subtest verifies the ability of the FIFO to correctly
2302                                    ;       pass a single data byte from input to output. For each
2303                                    ;       of 256 data values (0-377 octal) the following is done:
2304                                    ;               1. Initial FIFO status is checked
2305                                    ;               2. The Write FIFO function, specifying a count of
2306                                    ;                  one byte to be written is executed.
2307                                    ;               3. Read Status is executed and FIFO status is checked.
2308                                    ;               4. Read FIFO is executed and the data and final status
2309                                    ;                  is checked.
2310                                    ;
2311                                    ; TEST STEPS:
2312                                    ;
2313                                    ; BEGIN
2314                                    ;       Write to TSSR to soft initialize
2315                                    ;       Do a WRITE CHARACTERISTICS to setup a message buffer
2316                                    ;       Do a Write Subsystem READ STATUS
2317                                    ;       If Input Ready  NOT=1   Then Print Error
2318                                    ;       If Output Ready NOT=0   Then Print Error
2319                                    ;       If Data In Miss NOT=0   Then Print Error
2320                                    ;
2321                                    ;  REPEAT FOR DATA FROM 0 TO 377 OCTAL
2322                                    ;    BEGIN
```

```
                                ;       Do a Write Subsystem WRITE NPR to set tape direction out
                                ;       Do a Write Subsystem WRITE FIFO with byte count equal to 1
                                ;       Do a Write Subsystem READ STATUS
                                ;       If Input Ready  NOT=1   Then Print Error
                                ;       If Output Ready NOT=1   Then Print Error
                                ;       If Data In Miss NOT=0   Then Print Error
                                ;       Do Write Subsystem READ FIFO with byte count equal to 1
                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
                                ;       Do a Write Subsystem READ STATUS
                                ;       If Input Ready  NOT=1   Then Print Error
                                ;       If Output Ready NOT=0   Then Print Error
                                ;       If Data In Miss NOT=0   Then Print Error
                                ;   END
                                ; END
                                ;--
040744                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
040744                                                                                  T6.2:
040744  104402                                                                          TRAP    C$BSUB

                                ;       Write to TSSR register to soft initialize the controller
040746                          5$:
004737  015774                          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
103405                                  BCS     10$                     ;BR IF SOFT INIT OKAY
010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040756                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
040756  104455                                                                          TRAP    C$ERDF
040760  001133                                                                          .WORD   603
040762  003652                                                                          .WORD   SFIERR
040764  012034                                                                          .WORD   SFIMSG

                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
005037  002222              10$:        CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
012704  050250                          MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
004737  010662                          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
                                        FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
103407                                  BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        NEXT,ERRNO
041022                          42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
041022  104455                                                                          TRAP    C$ERDF
041024  001134                                                                          .WORD   604
041026  046675                                                                          .WORD   T17SSR
041030  012046                                                                          .WORD   PKTSSR
005237  002222                          INC     FATFLG                  ;SET FATAL ERROR FLAG
041036                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
041036  104406                                                                          TRAP    C$CLP1

                                ;       Do a Write Subsystem READ STATUS
004737  050034                          JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
012704  050420                          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
010465  000000                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
004737  016336                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
                                        FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
103407                                  BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        NEXT,ERRNO
041100                          62$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
041100  104455                                                                          TRAP    C$ERDF
041102  001135                                                                          .WORD   605
```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```
        041104  046776                                                          .WORD   T173SSR
        041106  012046                                                          .WORD   PKTSSR
 2367   041110  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
 2368   041114                  70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        041114  104406                                                          TRAP    C$CLP1
 2369                           ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
 2370   041116  004737  050216          JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
 2371   041122  012701  046452          MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
 2372   041126  012702  050312          MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
 2373   041132  012221                  MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
 2374   041134  011211                  MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
 2375   041136  052711  000020          BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= TRUE
 2376   041142  042711  000040          BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= FALSE
 2377   041146  042711  000200          BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = FALSE
 2378                           ;       If Input Ready NOT=1 then Print Error
 2379                           ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
 2380   041152  005000                  CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 2381   041154  012701  050272          MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 2382   041160  012702  046432          MOV     #T17EXP,R2              ;EXPD ADDRESS
 2383   041164  012703  000024          MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 2384   041170  004737  011500          JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 2385   041174                          FORCERROR       82$,NOTSSR      ;@@D
 2386   041204  103404                  BCS     90$                     ;BR IF YES
 2387   041206                          NEXT.ERRNO
 2388   041206                  82$:    ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
        041206  104456                                                          TRAP    C$ERHRD
        041210  001136                                                          .WORD   606
        041212  047215                                                          .WORD   T171CMP
        041214  012350                                                          .WORD   MSGSTAT
 2389   041216                  90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        041216  104406                                                          TRAP    C$CLP1
 2390
 2391                           ;  Repeat for DATA from 0 to 377
 2392   041220  012737  000000  002312          MOV     #0,DATA                 ;GET FIRST DATA
 2393   041226                  100$:                                   ;REPEAT LABEL
 2394                           ;       Do a Write Subsystem WRITE NPR to set tape direction out
 2395   041226  012700  000100          MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
 2396   041232  004737  050076          JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
 2397   041236  012704  050420          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 2398   041242  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 2399   041246  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 2400   041252                          FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
 2401   041266  103407                  BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
 2402   041270  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2403   041272                          NEXT.ERRNO
 2404   041272                  102$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        041272  104455                                                                  TRAP    C$ERDF
        041274  001137                                                                  .WORD   607
        041276  047043                                                                  .WORD   T174SSR
        041300  012046                                                                  .WORD   PKTSSR
 2405   041302  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
 2406   041306                  105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        041306  104406                                                          TRAP    C$CLP1
 2407                           ;       Do a Write Subsystem WRITE FIFO with byte count equal to 1
 2408   041310  012700  000001          MOV     #1,R0                   ;WRITE 1 BYTE
 2409   041314  012701  002312          MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
 2410   041320  004737  050122          JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```
2411 041324  012704  050420                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2412 041330  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2413 041334  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2414 041340                                FORCERROR       107$                    ;@@DFORCE ERROR IF FORCER=1
2415 041354  103407                        BCS     110$                    ;BR IF CARRY SET (GOOD RETURN)
2416 041356  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2417 041360                                NEXT.ERRNO
2418 041360                        107$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041360  104455                                                                  TRAP    C$ERDF
     041362  001140                                                                  .WORD   608
     041364  047.06                                                                  .WORD   T175SSR
     041366  012046                                                                  .WORD   PKTSSR
2419 041370  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
2420 041374                        110$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041374  104406                                                                  TRAP    C$CLP1
2421
2422                               ;       Do a Write Subsystem READ STATUS
2423 041376  004737  050034                JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2424 041402  012704  050420                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2425 041406  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2426 041412  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2427 041416                                FORCERROR       112$                    ;@@DFORCE ERROR IF FORCER=1
2428 041432  103407                        BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
2429 041434  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2430 041436                                NEXT.ERRNO
2431 041436                        112$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041436  104455                                                                  TRAP    C$ERDF
     041440  001141                                                                  .WORD   609
     041442  046776                                                                  .WORD   T173SSR
     041444  012046                                                                  .WORD   PKTSSR
2432 041446  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
2433 041452                        120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041452  104406                                                                  TRAP    C$CLP1
2434                               ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2435 041454  004737  050216                JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2436 041460  012701  046452                MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2437 041464  012702  050312                MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2438 041470  012221                        MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2439 041472  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2440 041474  052711  000020                BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
2441 041500  052711  000040                BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
2442 041504  042711  000200                BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
2443                               ;       If Input Ready NOT=1 then Print Error
2444                               ;       If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
2445 041510  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2446 041512  012701  050272                MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2447 041516  012702  046432                MOV     #T17EXP,R2              ;EXPD ADDRESS
2448 041522  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2449 041526  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2450 041532                                FORCERROR       132$,NOTSSR             ;@@D
2451 041542  103404                        BCS     140$                    ;BR IF YES
2452 041544                                NEXT.ERRNO
2453 041544                        132$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
     041544  104456                                                                  TRAP    C$ERHRD
     041546  001142                                                                  .WORD   610
     041550  047373                                                                  .WORD   T173CMP
     041552  012350                                                                  .WORD   MSGSTAT
```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```
2454 041554                          140$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041554  104406                                                                  TRAP    C$CLP1
2455
2456                                 ;       Do Write Subsystem READ FIFO with byte count equal to 1
2457 041556  012700  000001                  MOV     #1,R0                   ;SET READ BYTE COUNT
2458 041562  004737  050156                  JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
2459 041566  012704  050420                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2460 041572  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2461 041576  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2462 041602                                  FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
2463 041616  103407                          BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
2464 041620  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2465 041622                                  NEXT.ERRNO
2466 041622                          142$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041622  104455                                                                  TRAP    C$ERDF
     041624  001143                                                                  .WORD   611
     041626  047152                                                                  .WORD   T176SSR
     041630  012046                                                                  .WORD   PKTSSR
2467 041632  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
2468 041636                          150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041636  104406                                                                  TRAP    C$CLP1
2469                                 ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2470 041640  004737  050216                  JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2471 041644  012701  046452                  MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2472 041650  012702  050312                  MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2473 041654  013721  002312                  MOV     DATA,(R1)+              ;SET EXPD WORD #8 = COUNT DATA
2474 041660  011211                          MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV (NOT TESTING)
2475                                 ;       If Data read from FIFO NOT= to Data sent Then Print Error
2476                                 ;       The data is in WORD #8 of the message buffer
2477 041662  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2478 041664  012701  050272                  MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2479 041670  012702  046432                  MOV     #T17EXP,R2              ;EXPD ADDRESS
2480 041674  012703  000022                  MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
2481 041700  004737  011500                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2482 041704                                  FORCERROR       152$,NOTSSR     ;@@D
2483 041714  103404                          BCS     160$                    ;BR IF YES
2484 041716                                  NEXT.ERRNO
2485 041716                          152$:   ERRHRD  ERRNO,T172CMP,MSGSUB    ;REPORT ERROR
     041716  104456                                                                  TRAP    C$ERHRD
     041720  001144                                                                  .WORD   612
     041722  047277                                                                  .WORD   T172CMP
     041724  013742                                                                  .WORD   MSGSUB
2486 041726                          160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041726  104406                                                                  TRAP    C$CLP1
2487
2488                                 ;       Do a Write Subsystem READ STATUS
2489 041730  004737  050034                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2490 041734  012704  050420                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2491 041740  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2492 041744  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2493 041750                                  FORCERROR       162$                    ;@@DFORCE ERROR IF FORCER=1
2494 041764  103407                          BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
2495 041766  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2496 041770                                  NEXT.ERRNO
2497 041770                          162$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041770  104455                                                                  TRAP    C$ERDF
     041772  001145                                                                  .WORD   613
```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```
        041774  046776                                                          .WORD   T173SSR
        041776  012046                                                          .WORD   PKTSSR
2498    042000  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
2499    042004                  170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        042004  104406                                                          TRAP    C$CLP1
2500                            ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2501    042006  004737  050216          JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2502    042012  012701  046452          MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2503    042016  012702  050312          MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2504    042022  012221                  MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2505    042024  011211                  MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2506    042026  052711  000020          BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
2507    042032  042711  000040          BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
2508    042036  042711  000200          BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
2509                            ;       If Input Ready NOT=1 then Print Error
2510                            ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2511    042042  005000                  CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2512    042044  012701  050272          MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2513    042050  012702  046432          MOV     #T17EXP,R2              ;EXPD ADDRESS
2514    042054  012703  000024          MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2515    042060  004737  011500          JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2516    042064                          FORCERROR       172$,NOTSSR     ;@@D
2517    042074  103404                  BCS     180$                    ;BR IF YES
2518    042076                          NEXT.ERRNO
2519    042076                  172$:   ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
        042076  104456                                                          TRAP    C$ERHRD
        042100  001146                                                          .WORD   614
        042102  047457                                                          .WORD   T174CMP
        042104  012350                                                          .WORD   MSGSTAT
2520    042106                  180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        042106  104406                                                          TRAP    C$CLP1
2521    042110                          FORCEXIT        205$            ;@@D
2522    042120  005237  002312          INC     DATA                    ;GET NEXT TEST DATA
2523    042124  023727  002312  000377  CMP     DATA,#377               ;DONE 0 TO 377?
2524    042132  101002                  BHI     205$                    ;BR IF YES
2525    042134  000137  041226          JMP     100$                    ;DO ANOTHER TEST PATTERN
2526    042140                  205$:
2527
2528    042140                          ENDSUB                          ;////////////// END SUBTEST //////////
        042140                                                                  L10060:
        042140  104403                                                                  TRAP    C$ESUB
2529
2530    042142  005737  002222          TST     FATFLG                  ;ANY FATAL ERRORS ?
2531    042146  001402                  BEQ     260$                    ;BRANCH IF NOT
2532    042150  004737  017202          JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2533    042154                  260$:
2534
2535                                    .SBTTL  TEST  6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST
2536
2537                            ;++
2538                            ; TEST 6: SUBTEST 3:
2539                            ;
2540                            ; SUBTEST DESCRIPTION:
2541                            ;
2542                            ;       This subtest verifies the ability of the FIFO to correctly
2543                            ;       pass a multiple data bytes from input to output.
2544                            ;       The following sequence is done with various data patterns
```

```
2545                                    ;       and byte counts from 2 to 64.
2546                                    ;               1. Initial FIFO status is checked
2547                                    ;               2. The Write FIFO function.
2548                                    ;               3. Read Status is executed and FIFO status is checked.
2549                                    ;               4. Read FIFO is executed and the data and final status
2550                                    ;                  is checked.
2551                                    ;
2552                                    ; TEST STEPS:
2553                                    ;
2554                                    ; BEGIN
2555                                    ;       Write to TSSR to soft initialize
2556                                    ;       Do a WRITE CHARACTERISTICS to setup a message buffer
2557                                    ;       Do a Write Subsystem READ STATUS
2558                                    ;       If Input Ready  NOT=1   Then Print Error
2559                                    ;       If Output Ready NOT=0   Then Print Error
2560                                    ;       If Data In Miss NOT=0   Then Print Error
2561                                    ;       If Last Word    NOT=0   Then Print Error
2562                                    ;  REPEAT FOR DATA 0 TO 377, 377 TO 0, FLOATING 1'S,0'S AND ALL 1'S/0'S
2563                                    ;  REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
2564                                    ;    BEGIN
2565                                    ;       Do a Write Subsystem WRITE NPR to set tape direction out
2566                                    ;       Do a Write Subsystem WRITE FIFO
2567                                    ;       Do a Write Subsystem READ STATUS
2568                                    ;       If Input Ready  NOT=1   Then Print Error
2569                                    ;       If Output Ready NOT=1   Then Print Error
2570                                    ;       If Data In Miss NOT=0   Then Print Error
2571                                    ;       If Last Word    NOT=0   Then Print Error
2572                                    ;       Do Write Subsystem READ FIFO
2573                                    ;       If Data read from FIFO NOT= to Data sent Then Print Error
2574                                    ;       Do a Write Subsystem READ STATUS
2575                                    ;       If Input Ready  NOT=1   Then Print Error
2576                                    ;       If Output Ready NOT=0   Then Print Error
2577                                    ;       If Data In Miss NOT=0   Then Print Error
2578                                    ;       If Last Word    NOT=0   Then Print Error
2579                                    ;    END
2580                                    ;  END
2581                                    ;--
2582 042154                                     BGNSUB                          ;////////////// BEGIN SUBTEST //////////
      042154                                                                                      T6.3:
      042154  104402                                                                                     TRAP    C$BSUB
2583
2584                                    ;       Write to TSSR register to soft initialize the controller
2585 042156                             5$:
2586 042156  004737  015774                        JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
2587 042162  103405                                BCS     10$                     ;BR IF SOFT INIT OKAY
2588 042164  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2589 042166                                        ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      042166  104455                                                                                     TRAP    C$ERDF
      042170  001146                                                                                     .WORD   614
      042172  003652                                                                                     .WORD   SFIERR
      042174  012034                                                                                     .WORD   SFIMSG
2590                                    ;       Do a WRITE CHARACTERISTICS to setup a message buffer
2591 042176  005037  002222             10$:       CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
2592 042202  012704  050250                        MOV     @T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
2593 042206  004737  010662                        JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
2594 042212                                        FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
2595 042226  103407                                BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```
 2596 042230  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2597 042232                                 NEXT.ERRNO
 2598 042232                          42$:   ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      042232  104455                                                                TRAP    C$ERDF
      042234  001147                                                                .WORD   615
      042236  046675                                                                .WORD   T17SSR
      042240  012046                                                                .WORD   PKTSSR
 2599 042242  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 2600 042246                          50$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      042246  104406                                                                TRAP    C$CLP1
 2601                                 ;      Do a Write Subsystem READ STATUS
 2602 042250  004737  050034                 JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
 2603 042254  012704  050420                 MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 2604 042260  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 2605 042264  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 2606 042270                                 FORCERROR       62$             ;@@DFORCE ERROR IF FORCER=1
 2607 042304  103407                         BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
 2608 042306  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2609 042310                                 NEXT.ERRNO
 2610 042310                          62$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      042310  104455                                                                TRAP    C$ERDF
      042312  001150                                                                .WORD   616
      042314  046776                                                                .WORD   T173SSR
      042316  012046                                                                .WORD   PKTSSR
 2611 042320  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 2612 042324                          70$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      042324  104406                                                                TRAP    C$CLP1
 2613                                 ;      Set WORDS 0-7 of expd message buffer = to recv since not testing
 2614 042326  004737  050216                 JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
 2615 042332  012701  046452                 MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
 2616 042336  012702  050312                 MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
 2617 042342  012221                         MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
 2618 042344  011211                         MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
 2619 042346  052711  000020                 BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY=  1
 2620 042352  042711  000040                 BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY=  0
 2621 042356  042711  000200                 BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
 2622 042362  042711  000100                 BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
 2623                                 ;      If Input Ready NOT=1 then Print Error
 2624                                 ;      If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
 2625                                 ;      If Last Word    NOT=0   Then Print Error
 2626 042366  005000                         CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 2627 042370  012701  050272                 MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 2628 042374  012702  046432                 MOV     #T17EXP,R2              ;EXPD ADDRESS
 2629 042400  012703  000024                 MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 2630 042404  004737  011500                 JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 2631 042410                                 FORCERROR       82$,NOTSSR      ;@@D
 2632 042420  103404                         BCS     90$                     ;BR IF YES
 2633 042422                                 NEXT.ERRNO
 2634 042422                          82$:   ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
      042422  104456                                                                TRAP    C$ERHRD
      042424  001151                                                                .WORD   617
      042426  047215                                                                .WORD   T171CMP
      042430  012350                                                                .WORD   MSGSTAT
 2635 042432                          90$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      042432  104406                                                                TRAP    C$CLP1
 2636
 2637
```

```
                                        ;  REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
                                        ;  TSTFLAG =1 FOR INCREMENT TEST PATTERN
                                        ;           =2 FOR DECREMENT TEST PATTERN
                                        ;           =3 FOR TSTBLK TABLE PATTERN
042434  012737  000001  002314          MOV     #1,TSTFLAG              ;TEST PATTERN FLAG
042442                                  95$:
042442  012737  000002  002310          MOV     #2,COUNT                ;GET FIRST BYTE COUNT
042450                                  100$:
                                        ;       Do a Write Subsystem WRITE NPR to set tape direction out
042450  012700  000100                  MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
042454  004737  050076                  JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
042460  012704  050420                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
042464  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
042470  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
042474                                  FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
042510  103407                          BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
042512  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
042514                                  NEXT.ERRNO
042514                                  102$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
042514  104455                                                                          TRAP    C$ERDF
042516  001152                                                                          .WORD   618
042520  047043                                                                          .WORD   T174SSR
042522  012046                                                                          .WORD   PKTSSR
042524  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
042530                                  105$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
042530  104406                                                                          TRAP    C$CLP1
                                        ;       Do a Write Subsystem WRITE FIFO
042532  004737  050176                  JSR     PC,T17CLEXP             ;CLEAR EXPD BUFFER
042536  012701  046554                  MOV     #T17WFDATA,R1           ;EXPD WRITE FIFO DATA BUFFER
042542  013702  002310                  MOV     COUNT,R2                ;TEST PATTERN SIZE
042546  022737  000001  002314          CMP     #1,TSTFLAG              ;INCREMENT PATTERN THIS TIME THRU?
042554  001005                          BNE     115$                    ;BR IF NO
042556  005000                          CLR     R0                      ;INCREMENT TEST PATTERN
042560  110021                  110$:   MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
042562  005200                          INC     R0                      ;SET NEXT PATTERN
042564  005302                          DEC     R2                      ;DONE?
042566  003374                          BGT     110$                    ;BR IF NO
042570  022737  000002  002314  115$:   CMP     #2,TSTFLAG              ;DECREMENT PATTERN THIS TIME THRU?
042576  001006                          BNE     125$                    ;BR IF NO
042600  012700  000377                  MOV     #377,R0                 ;DECREMENT TEST PATTERN
042604  110021                  120$:   MOVB    R0,(R1)+                ;STORE DECREMENT TEST BYTE
042606  005300                          DEC     R0                      ;SET NEXT PATTERN
042610  005302                          DEC     R2                      ;DONE?
042612  003374                          BGT     120$                    ;BR IF NO
042614  022737  000003  002314  125$:   CMP     #3,TSTFLAG              ;TSTBLK PATTERNS THIS TIME THRU?
042622  001005                          BNE     135$                    ;BR IF NO
042624  012700  002752                  MOV     #TSTBLK,R0              ;FLOAT 1'S/0'S ETC. TEST TABLE
042630  112021                  130$:   MOVB    (R0)+,(R1)+             ;STORE A TSTBLK BYTE
042632  005302                          DEC     R2                      ;DONE?
042634  003375                          BGT     130$                    ;BR IF NO
042636                                  135$:
042636  013700  002310                  MOV     COUNT,R0                ;FIFO BYTE COUNT
042642  012701  046554                  MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
042646  004737  050122                  JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
042652  012704  050420                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
042656  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```
2690 042662  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2691 042666                                  FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
2692 042702  103407                          BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
2693 042704  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2694 042706                                  NEXT.ERRNO
2695 042706                          142$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042706  104455                                                                  TRAP    C$ERDF
     042710  001153                                                                  .WORD   619
     042712  047106                                                                  .WORD   T175SSR
     042714  012046                                                                  .WORD   PKTSSR
2696 042716  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
2697 042722                          150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042722  104406                                                                  TRAP    C$CLP1
2698
2699                                 ;       Do a Write Subsystem READ STATUS
2700 042724  004737  050034                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2701 042730  012704  050420                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2702 042734  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2703 042740  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2704 042744                                  FORCERROR       157$                    ;@@DFORCE ERROR IF FORCER=1
2705 042760  103407                          BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
2706 042762  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2707 042764                                  NEXT.ERRNO
2708 042764                          157$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042764  104455                                                                  TRAP    C$ERDF
     042766  001154                                                                  .WORD   620
     042770  046776                                                                  .WORD   T173SSR
     042772  012046                                                                  .WORD   PKTSSR
2709 042774  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
2710 043000                          160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043000  104406                                                                  TRAP    C$CLP1
2711
2712                                 ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2713 043002  004737  050216                  JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2714 043006  012701  046452                  MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2715 043012  012702  050312                  MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2716 043016  012221                          MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2717 043020  011211                          MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2718 043022  052711  000020                  BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
2719 043026  052711  000040                  BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
2720 043032  042711  000200                  BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
2721 043036  042711  000100                  BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
2722                                 ;       If Input Ready NOT=1 then Print Error
2723                                 ;       If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
2724 043042  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2725 043044  012701  050272                  MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2726 043050  012702  046432                  MOV     #T17EXP,R2              ;EXPD ADDRESS
2727 043054  012703  000024                  MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2728 043060  004737  011500                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2729 043064                                  FORCERROR       162$,NOTSSR             ;@@D
2730 043074  103404                          BCS     170$                    ;BR IF YES
2731 043076                                  NEXT.ERRNO
2732 043076                          162$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
     043076  104456                                                                  TRAP    C$ERHRD
     043100  001155                                                                  .WORD   621
     043102  047373                                                                  .WORD   T173CMP
     043104  012350                                                                  .WORD   MSGSTAT
```

```
2733 043106                          170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043106  104406                                                                  TRAP    C$CLP1
2734
2735                                 ;       Do Write Subsystem READ FIFO
2736 043110  013700  002310                  MOV     COUNT,R0                ;SET READ BYTE COUNT
2737 043114  004737  050156                  JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
2738 043120  012704  050420                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2739 043124  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2740 043130  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2741 043134                                  FORCERROR       172$                    ;@@DFORCE ERROR IF FORCER=1
2742 043150  103407                          BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
2743 043152  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2744 043154                                  NEXT.ERRNO
2745 043154                          172$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     043154  104455                                                                  TRAP    C$ERDF
     043156  001156                                                                  .WORD   622
     043160  047152                                                                  .WORD   T176SSR
     043162  012046                                                                  .WORD   PKTSSR
2746 043164  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
2747 043170                          180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043170  104406                                                                  TRAP    C$CLP1
2748
2749                                 ;       If Data read from FIFO NOT= to Data sent Then Print Error
2750 043172  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2751 043174  012702  046554                  MOV     #T17WFDATA,R2           ;GET EXPECTED ADDRESS FOR CKMSG2
2752 043200  012701  050312                  MOV     #T17BFSTA,R1            ;GET RECEIVED ADDRESS FOR CKMSG2
2753 043204  013703  002310                  MOV     COUNT,R3                ;NUMBER OF BYTES TO COMPARE
2754 043210  004737  011500                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2755 043214                                  FORCERROR       192$,NOTSSR             ;@@D
2756 043224  103406                          BCS     200$                    ;BR IF YES
2757 043226                                  NEXT.ERRNO
2758 043226  013701  002310          192$:   MOV     COUNT,R1                ;GET BYTE COUNT
2759 043232                                  ERRHRD  ERRNO,T175CMP,FIFEXP    ;REPORT ERROR
     043232  104456                                                                  TRAP    C$ERHRD
     043234  001157                                                                  .WORD   623
     043236  047542                                                                  .WORD   T175CMP
     043240  012170                                                                  .WORD   FIFEXP
2760 043242                          200$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043242  104406                                                                  TRAP    C$CLP1
2761
2762                                 ;       Do a Write Subsystem READ STATUS
2763 043244  004737  050034                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2764 043250  012704  050420                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2765 043254  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2766 043260  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2767 043264                                  FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
2768 043300  103407                          BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
2769 043302  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2770 043304                                  NEXT.ERRNO
2771 043304                          212$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     043304  104455                                                                  TRAP    C$ERDF
     043306  001160                                                                  .WORD   624
     043310  046776                                                                  .WORD   T173SSR
     043312  012046                                                                  .WORD   PKTSSR
2772 043314  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
2773 043320                          220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043320  104406                                                                  TRAP    C$CLP1
```

```
2774                                    ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2775 043322  004737  050216                     JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2776 043326  012701  046452                     MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2777 043332  012702  050312                     MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2778 043336  012221                             MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2779 043340  011211                             MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2780 043342  052711  000020                     BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
2781 043346  042711  000040                     BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
2782 043352  042711  000200                     BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
2783 043356  042711  000100                     BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
2784                                    ;       If Input Ready NOT=1 then Print Error
2785                                    ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2786 043362  005000                             CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2787 043364  012701  050272                     MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2788 043370  012702  046432                     MOV     #T17EXP,R2              ;EXPD ADDRESS
2789 043374  012703  000024                     MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2790 043400  004737  011500                     JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2791 043404                                     FORCERROR       232$,NOTSSR     ;@@D
2792 043414  103404                             BCS     240$                    ;BR IF YES
2793 043416                                     NEXT.ERRNO
2794 043416                             232$:   ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
     043416  104456                                                                     TRAP    C$ERHRD
     043420  001161                                                                     .WORD   625
     043422  047457                                                                     .WORD   T174CMP
     043424  012350                                                                     .WORD   MSGSTAT
2795 043426                             240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043426  104406                                                                     TRAP    C$CLP1
2796 043430                                     FORCEXIT        250$            ;@@D
2797 043440  005237  002310                     INC     COUNT                   ;GET NEXT BYTE COUNT
2798 043444  023727  002310  000077             CMP     COUNT,#77               ;DONE 0 TO 77
2799 043452  101002                             BHI     250$                    ;BR IF YES
2800 043454  000137  042450                     JMP     100$                    ;DO ANOTHER BYTE COUNT
2801 043460  005237  002314             250$:   INC     TSTFLAG                 ;GET NEXT TEST PATTERN CODE
2802 043464  023727  002314  000003             CMP     TSTFLAG,#3              ;DONE INC,DEC,TSTBLK PATTERNS?
2803 043472  101002                             BHI     255$                    ;BR IF YES
2804 043474  000137  042442                     JMP     95$                     ;DO ANOTHER TEST PATTERN
2805 043500                             255$:
2806 043500                                     ENDSUB                          ;////////////// END SUBTEST //////////
     043500                                                                             L10061:
     043500  104403                                                                             TRAP    C$ESUB
2807
2808 043502  005737  002222                     TST     FATFLG                  ;ANY FATAL ERRORS ?
2809 043506  001402                             BEQ     260$                    ;BRANCH IF NOT
2810 043510  004737  017202                     JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2811 043514                             260$:
2812
2813
2814
2815                                            .SBTTL  TEST  6: SUBTEST 4: FIFO Verify ILW Status
2816
2817                                    ;++
2818                                    ; TEST 6: SUBTEST 4:
2819                                    ;
2820                                    ; SUBTEST DESCRIPTION:
2821                                    ;
2822                                    ;       This subtest verifies that reading the FIFO when it is
2823                                    ;       empty causes the Last Word (ILW) status to assert.
```

```
                                ;
                                ; TEST STEPS:
                                ;
                                ; BEGIN
                                ;       Write to TSSR to soft initialize
                                ;       Do Write Subsystem READ FIFO with byte count equal to 1
                                ;       Do a Write Subsystem READ STATUS
                                ;       If Input Ready  NOT=1   Then Print Error
                                ;       If Output Ready NOT=0   Then Print Error
                                ;       If Data In Miss NOT=0   Then Print Error
                                ;       If Last Word (ILW) NOT=1 Then Print Error
                                ;    END
                                ;--
043514                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
043514                                                                                  T6.4:
043514  104402                                                                          TRAP    C$BSUB

                                ;       Write to TSSR register to soft initialize the controller
043516                          5$:
043516  004737  015774                  JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
043522  103405                          BCS     10$                     ;BR IF SOFT INIT OKAY
043524  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043526                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
043526  104455                                                                          TRAP    C$ERDF
043530  001161                                                                          .WORD   625
043532  003652                                                                          .WORD   SFIERR
043534  012034                                                                          .WORD   SFIMSG

                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
043536  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
043542  012704  050250                  MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
043546  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
043552                                  FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
043566  103407                          BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
043570  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043572                                  NEXT.ERRNO
043572                          42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
043572  104455                                                                          TRAP    C$ERDF
043574  001162                                                                          .WORD   626
043576  046675                                                                          .WORD   T17SSR
043600  012046                                                                          .WORD   PKTSSR
043602  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
043606                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
043606  104406                                                                          TRAP    C$CLP1

                                ;       Do Write Subsystem READ FIFO with byte count equal to 1
043610  012700  000001                  MOV     #1,R0                   ;SET READ BYTE COUNT
043614  004737  050156                  JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
043620  012704  050420                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
043624  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
043630  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
043634                                  FORCERROR       142$                    ;@@FORCE ERROR IF FORCER=1
043650  103407                          BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
043652  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043654                                  NEXT.ERRNO
043654                          142$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
043654  104455                                                                          TRAP    C$ERDF
043656  001163                                                                          .WORD   627
```

TEST 6: SUBTEST 4: FIFO VERIFY ILW STATUS

```
                043660  047152                                                          .WORD   T176SSR
                043662  012046                                                          .WORD   PKTSSR
2868            043664  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
2869            043670                  150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                043670  104406                                                          TRAP    C$CLP1
2870
2871                            ;       Do a Write Subsystem READ STATUS
2872            043672  004737  050034          JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2873            043676  012704  050420          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2874            043702  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2875            043706  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2876            043712                          FORCERROR       162$                    ;@@DFORCE ERROR IF FORCER=1
2877            043726  103407                  BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
2878            043730  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2879            043732                          NEXT.ERRNO
2880            043732                  162$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                043732  104455                                                          TRAP    C$ERDF
                043734  001164                                                          .WORD   628
                043736  046776                                                          .WORD   T173SSR
                043740  012046                                                          .WORD   PKTSSR
2881            043742  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
2882            043746                  170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                043746  104406                                                          TRAP    C$CLP1
2883                            ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2884            043750  004737  050216          JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2885            043754  012701  046452          MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2886            043760  012702  050312          MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2887            043764  012221                  MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2888            043766  011211                  MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2889            043770  052711  000020          BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
2890            043774  042711  000040          BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
2891            044000  042711  000200          BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
2892            044004  052711  000100          BIS     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=1
2893                            ;       If Input Ready NOT=1 then Print Error
2894                            ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2895                            ;       If Last Word (ILW) NOT=1 Then Print Error
2896            044010  005000          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2897            044012  012701  050272          MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2898            044016  012702  046432          MOV     #T17EXP,R2              ;EXPD ADDRESS
2899            044022  012703  000024          MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2900            044026  004737  011500          JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2901            044032                          FORCERROR       172$,NOTSSR             ;@@D
2902            044042  103404                  BCS     180$                    ;BR IF YES
2903            044044                          NEXT.ERRNO
2904            044044                  172$:   ERRHRD  ERRNO,T176CMP,MSGSTAT   ;REPORT ERROR
                044044  104456                                                          TRAP    C$ERHRD
                044046  001165                                                          .WORD   629
                044050  047616                                                          .WORD   T176CMP
                044052  012350                                                          .WORD   MSGSTAT
2905            044054                  180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                044054  104406                                                          TRAP    C$CLP1
2906
2907            044056                          ENDSUB                          ;////////////// END SUBTEST //////////
                044056                                                                  L10062:
                044056  104403                                                          TRAP    C$ESUB
2908
2909            044060  005737  002222          TST     FATFLG                  ;ANY FATAL ERRORS ?
```

```
2910 044064  001402                        BEQ     260$                    ;BRANCH IF NOT
2911 044066  004737  017202                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2912 044072                          260$:
2913
2914
2915                                        .SBTTL  TEST  6: SUBTEST 5: FIFO Verify Input Ready
2916
2917                                ;**
2918                                ; TEST 6: SUBTEST 5:
2919                                ;
2920                                ; SUBTEST DESCRIPTION:
2921                                ;
2922                                ;       This subtest verifies that writing 64. bytes into the FIFO
2923                                ;       without reading any out causes the Input Ready status to
2924                                ;       negate. The Subtest then verfiies that writing a 65th byte
2925                                ;       into the FIFO causes the Data In Miss status to assert.
2926                                ;       Next it is verified that the original 64 bytes can be read
2927                                ;       out correctly and that the data has not been corrupted.
2928                                ;
2929                                ; TEST STEPS:
2930                                ;
2931                                ; BEGIN
2932                                ;       Write to TSSR to soft initialize
2933                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
2934                                ;       Do a Write Subsystem WRITE NPR to set tape direction out
2935                                ;       Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
2936                                ;       Do a Write Subsystem READ STATUS
2937                                ;       If Input Ready  NOT=0   Then Print Error
2938                                ;       If Output Ready NOT=1   Then Print Error
2939                                ;       If Data In Miss NOT=0   Then Print Error
2940                                ;       Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
2941                                ;       Do a Write Subsystem READ STATUS
2942                                ;       If Input Ready  NOT=0   Then Print Error
2943                                ;       If Output Ready NOT=1   Then Print Error
2944                                ;       If Data In Miss NOT=1   Then Print Error
2945                                ;       Do Write Subsystem READ FIFO
2946                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
2947                                ;       Do a Write Subsystem READ STATUS
2948                                ;       If Input Ready  NOT=1   Then Print Error
2949                                ;       If Output Ready NOT=0   Then Print Error
2950                                ;       If Data In Miss NOT=1   Then Print Error
2951                                ; END
2952                                ;--
2953 044072                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////
      044072                                                                      T6.5:
      044072  104402                                                                TRAP    C$BSUB
2954
2955                                ;       Write to TSSR register to soft initialize the controller
2956 044074                          5$:
2957 044074  004737  015774                JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
2958 044100  103405                        BCS     10$                     ;BR IF SOFT INIT OKAY
2959 044102  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2960 044104                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      044104  104455                                                                TRAP    C$ERDF
      044106  001165                                                                .WORD   629
      044110  003652                                                                .WORD   SFIERR
      044112  012034                                                                .WORD   SFIMSG
```

TEST 6: SUBTEST 5: FIFO VERIFY INPUT READY

```
2961                                    ;       Do a WRITE CHARACTERISTICS to setup a message buffer
2962 044114  005037  002222             10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
2963 044120  012704  050250                     MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
2964 044124  004737  010662                     JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
2965 044130                                     FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
2966 044144  103407                             BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
2967 044146  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2968 044150                                     NEXT.ERRNO
2969 044150                             42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     044150  104455                                                                     TRAP    C$ERDF
     044152  001166                                                                     .WORD   630
     044154  046675                                                                     .WORD   T17SSR
     044156  012046                                                                     .WORD   PKTSSR
2970 044160  005237  002222                     INC     FATFLG                  ;SET FATAL ERROR FLAG
2971 044164                             50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044164  104406                                                                     TRAP    C$CLP1
2972
2973                                    ;       Do a Write Subsystem WRITE NPR to set tape direction out
2974 044166  012700  000100             100$:   MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
2975 044172  004737  050076                     JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
2976 044176  012704  050420                     MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2977 044202  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2978 044206  004737  016336                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2979 044212                                     FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
2980 044226  103407                             BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
2981 044230  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2982 044232                                     NEXT.ERRNO
2983 044232                             102$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044232  104455                                                                     TRAP    C$ERDF
     044234  001167                                                                     .WORD   631
     044236  047043                                                                     .WORD   T174SSR
     044240  012046                                                                     .WORD   PKTSSR
2984 044242  005237  002222                     INC     FATFLG                  ;SET FATAL ERROR FLAG
2985 044246                             105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044246  104406                                                                     TRAP    C$CLP1
2986
2987                                    ;       Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
2988 044250  012737  000100  002310             MOV     #64.,COUNT              ;WRITE 64 BYTES
2989 044256  012701  046554                     MOV     #T17WFDATA,R1           ;EXPD WRITE FIFO DATA BUFFER
2990 044262  012702  000100                     MOV     #64.,R2                 ;TEST PATTERN SIZE
2991 044266  005000                             CLR     R0                      ;INCREMENT TEST PATTERN
2992 044270  110021                     110$:   MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
2993 044272  005200                             INC     R0                      ;SET NEXT PATTERN
2994 044274  005302                             DEC     R2                      ;DONE?
2995 044276  003374                             BGT     110$                    ;BR IF NO
2996 044300  013700  002310                     MOV     COUNT,R0                ;FIFO BYTE COUNT
2997 044304  012701  046554                     MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
2998 044310  004737  050122                     JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
2999 044314  012704  050420                     MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3000 044320  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3001 044324  004737  016336                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3002 044330                                     FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
3003 044344  103407                             BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
3004 044346  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3005 044350                                     NEXT.ERRNO
3006 044350                             142$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044350  104455                                                                     TRAP    C$ERDF
```

TEST 6: SUBTEST 5: FIFO VERIFY INPUT READY

```
                                                                                .WORD   632
                                                                                .WORD   T175SSR
                                                                                .WORD   PKTSSR
3007 044360  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
3008 044364                  150$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044364  104406                                                             TRAP    C$CLP1
3009
3010                                ;       Do a Write Subsystem READ STATUS
3011                                ;       If Input Ready  NOT=0   Then Print Error
3012                                ;       If Output Ready NOT=1   Then Print Error
3013                                ;       If Data In Miss NOT=0   Then Print Error
3014 044366  004737  050034          JSR     PC,T17SRD              ;SETUP PACKET FOR READ STATUS
3015 044372  012704  050420          MOV     #T17PK2,R4             ;GET WRITE SUBSYSTEM COMMAND PACKET
3016 044376  010465  000000          MOV     R4,TSDB(R5)            ;SET THE PACKET ADDRESS TO EXECUTE
3017 044402  004737  016336          JSR     PC,CHKTSSR             ;WAIT FOR SSR TO SET
3018 044406                          FORCERROR       157$                    ;@@DFORCE ERROR IF FORCER=1
3019 044422  103407                  BCS     160$                   ;BR IF CARRY SET (GOOD RETURN)
3020 044424  010001                  MOV     R0,R1                  ;SAVE CONTENTS OF TSSR
3021 044426                          NEXT.ERRNO
3022 044426                  157$:  ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044426  104455                                                             TRAP    C$ERDF
     044430  001171                                                             .WORD   633
     044432  046776                                                             .WORD   T173SSR
     044434  012046                                                             .WORD   PKTSSR
3023 044436  005237  002222          INC     FATFLG                 ;SET FATAL ERROR FLAG
3024 044442                  160$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044442  104406                                                             TRAP    C$CLP1
3025                                ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
3026 044444  004737  050216          JSR     PC,T17SETEXP           ;SET WORDS 0-7 EXPD=RECV
3027 044450  012701  046452          MOV     #T17EXSTA,R1           ;GET EXPECTED READ STATUS
3028 044454  012702  050312          MOV     #T17BFSTA,R2           ;GET RECV READ STATUS
3029 044460  012221                  MOV     (R2)+,(R1)+            ;SET EXPD WORD #8 = RECV TEMP
3030 044462  011211                  MOV     (R2),(R1)              ;SET EXPD WORD #9 = RECV TEMP
3031 044464  042711  000020          BIC     #S2.INRDY,(R1)         ;SET EXP INPUT READY= 0
3032 044470  052711  000040          BIS     #S2.OUTRDY,(R1)        ;SET EXP OUTPUT READY= 1
3033 044474  042711  000200          BIC     #S2.DIM,(R1)           ;SET EXP DATA IN MISS = 0
3034 044500  005000                  CLR     R0                     ;HIGH RECV ADDRESS FOR CKMSG2
3035 044502  012701  050272          MOV     #T17BFR,R1             ;LOW RECV ADDRESS FOR CKMSG2
3036 044506  012702  046432          MOV     #T17EXP,R2             ;EXPD ADDRESS
3037 044512  012703  000024          MOV     #20.,R3                ;NUMBER OF BYTES TO COMPARE
3038 044516  004737  011500          JSR     PC,CKMSG2              ;EXPD EQUAL RECV?
3039 044522                          FORCERROR       162$,NOTSSR            ;@@D
3040 044532  103404                  BCS     170$                   ;BR IF YES
3041 044534                          NEXT.ERRNO
3042 044534                  162$:  ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
     044534  104456                                                             TRAP    C$ERHRD
     044536  001172                                                             .WORD   634
     044540  047373                                                             .WORD   T173CMP
     044542  012350                                                             .WORD   MSGSTAT
3043 044544                  170$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044544  104406                                                             TRAP    C$CLP1
3044
3045
3046                                ;       Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
3047 044546  012700  000001          MOV     #1,R0                  ;FIFO BYTE COUNT
3048 044552  012701  046554          MOV     #T17WFDATA,R1          ;FIFO WRITE DATA ADDRESS
3049 044556  004737  050122          JSR     PC,T17WFIF             ;SETUP T17PK2 FOR WRITE FIFO
```

TEST 6: SUBTEST 5: FIFO VERIFY INPUT READY

```
3050 044562  012704  050420                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3051 044566  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3052 044572  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3053 044576                                    FORCERROR       172$                    ;@@DFORCE ERROR IF FORCER=1
3054 044612  103407                            BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
3055 044614  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3056 044616                                    NEXT.ERRNO
3057 044616                            172$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044616  104455                                                                    TRAP    C$ERDF
     044620  001173                                                                    .WORD   635
     044622  047106                                                                    .WORD   T175SSR
     044624  012046                                                                    .WORD   PKTSSR
3058 044626  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
3059 044632                            180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044632  104406                                                                    TRAP    C$CLP1
3060
3061                                   ;       Do a Write Subsystem READ STATUS
3062                                   ;       If Input Ready  NOT=0   Then Print Error
3063                                   ;       If Output Ready NOT=1   Then Print Error
3064                                   ;       If Data In Miss NOT=1   Then Print Error
3065 044634  004737  050034                    JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
3066 044640  012704  050420                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3067 044644  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3068 044650  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3069 044654                                    FORCERROR       187$                    ;@@DFORCE ERROR IF FORCER=1
3070 044670  103407                            BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
3071 044672  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3072 044674                                    NEXT.ERRNO
3073 044674                            187$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044674  104455                                                                    TRAP    C$ERDF
     044676  001174                                                                    .WORD   636
     044700  046776                                                                    .WORD   T173SSR
     044702  012046                                                                    .WORD   PKTSSR
3074 044704  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
3075 044710                            190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044710  104406                                                                    TRAP    C$CLP1
3076                                   ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
3077 044712  004737  050216                    JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
3078 044716  012701  046452                    MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
3079 044722  012702  050312                    MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
3080 044726  012221                            MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
3081 044730  011211                            MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
3082 044732  042711  000020                    BIC     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 0
3083 044736  052711  000040                    BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
3084 044742  052711  000200                    BIS     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 1
3085 044746  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
3086 044750  012701  050272                    MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
3087 044754  012702  046432                    MOV     #T17EXP,R2              ;EXPD ADDRESS
3088 044760  012703  000024                    MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
3089 044764  004737  011500                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3090 044770                                    FORCERROR       192$,NOTSSR             ;@@D
3091 045000  103404                            BCS     200$                    ;BR IF YES
3092 045002                                    NEXT.ERRNO
3093 045002                            192$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
     045002  104456                                                                    TRAP    C$ERHRD
     045004  001175                                                                    .WORD   637
     045006  047373                                                                    .WORD   T173CMP
```

TEST 6: SUBTEST 5: FIFO VERIFY INPUT READY

```
                                                                                .WORD   MSGSTAT
3094 045012                     200$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        045012  104406                                                         TRAP    C$CLP1
3095                            ;       Do Write Subsystem READ FIFO
3096 045014  013700  002310             MOV     COUNT,R0                ;SET READ BYTE COUNT
3097 045020  004737  050156             JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
3098 045024  012704  050420             MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3099 045030  010465  000000             MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3100 045034  004737  016336             JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3101 045040                             FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
3102 045054  103407                     BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
3103 045056  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3104 045060                             NEXT.ERRNO
3105 045060                     212$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        045060  104455                                                          TRAP    C$ERDF
        045062  001176                                                          .WORD   638
        045064  047152                                                          .WORD   T176SSR
        045066  012046                                                          .WORD   PKTSSR
3106 045070  005237  002222             INC     FATFLG                  ;SET FATAL ERROR FLAG
3107 045074                     220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        045074  104406                                                         TRAP    C$CLP1
3108
3109                            ;       If Data read from FIFO NOT= to Data sent Then Print Error
3110 045076  005000                     CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
3111 045100  012702  046554             MOV     #T17WFDATA,R2           ;GET EXPECTED ADDRESS FOR CKMSG2
3112 045104  012701  050312             MOV     #T17BFSTA,R1            ;GET RECEIVED ADDRESS FOR CKMSG2
3113 045110  013703  002310             MOV     COUNT,R3                ;NUMBER OF BYTES TO COMPARE
3114 045114  004737  011500             JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3115 045120                             FORCERROR       232$,NOTSSR             ;@@D
3116 045130  103406                     BCS     240$                    ;BR IF YES
3117 045132                             NEXT.ERRNO
3118 045132  013701  002310     232$:   MOV     COUNT,R1                ;GET BYTE COUNT
3119 045136                             ERRHRD  ERRNO,T175CMP,FIFEXP    ;REPORT ERROR
        045136  104456                                                         TRAP    C$ERHRD
        045140  001177                                                         .WORD   639
        045142  047542                                                         .WORD   T175CMP
        045144  012170                                                         .WORD   FIFEXP
3120 045146                     240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        045146  104406                                                         TRAP    C$CLP1
3121
3122                            ;       Do a Write Subsystem READ STATUS
3123                            ;       If Input Ready  NOT=1   Then Print Error
3124                            ;       If Output Ready NOT=0   Then Print Error
3125                            ;       If Data In Miss NOT=1   Then Print Error
3126 045150  004737  050034             JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
3127 045154  012704  050420             MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3128 045160  010465  000000             MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3129 045164  004737  016336             JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3130 045170                             FORCERROR       252$                    ;@@DFORCE ERROR IF FORCER=1
3131 045204  103407                     BCS     260$                    ;BR IF CARRY SET (GOOD RETURN)
3132 045206  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3133 045210                             NEXT.ERRNO
3134 045210                     252$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        045210  104455                                                          TRAP    C$ERDF
        045212  001200                                                          .WORD   640
        045214  046776                                                          .WORD   T173SSR
        045216  012046                                                          .WORD   PKTSSR
```

```
3135 045220  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
3136 045224                              260$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045224  104406                                                                    TRAP    C$CLP1
3137                                     ;      Set WORDS 0-7 of expd message buffer = to recv since not testing
3138 045226  004737  050216                    JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
3139 045232  012701  046452                    MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
3140 045236  012702  050312                    MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
3141 045242  012221                            MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
3142 045244  011211                            MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
3143 045246  052711  000020                    BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
3144 045252  042711  000040                    BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
3145 045256  052711  000200                    BIS     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 1
3146 045262  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
3147 045264  012701  050272                    MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
3148 045270  012702  046432                    MOV     #T17EXP,R2              ;EXPD ADDRESS
3149 045274  012703  000024                    MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
3150 045300  004737  011500                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3151 045304                                    FORCERROR       272$,NOTSSR     ;@@D
3152 045314  103404                            BCS     280$                    ;BR IF YES
3153 045316                                    NEXT.ERRNO
3154 045316                              272$: ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
     045316  104456                                                                    TRAP    C$ERHRD
     045320  001201                                                                    .WORD   641
     045322  047457                                                                    .WORD   T174CMP
     045324  012350                                                                    .WORD   MSGSTAT
3155 045326                              280$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045326  104406                                                                    TRAP    C$CLP1
3156
3157 045330                                    ENDSUB                          ;////////////// END SUBTEST //////////
     045330                                                                  L10063:
     045330  104403                                                                    TRAP    C$ESUB
3158
3159 045332  005737  002222                    TST     FATFLG                  ;ANY FATAL ERRORS ?
3160 045336  001402                            BEQ     300$                    ;BRANCH IF NOT
3161 045340  004737  017202                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
3162 045344                              300$:
3163
3164
3165
3166                                           .SBTTL  TEST  6: SUBTEST 6: FIFO Verify Reset FIFO Test
3167
3168                                     ;**
3169                                     ; TEST 6: SUBTEST 6:
3170                                     ;
3171                                     ; SUBTEST DESCRIPTION:
3172                                     ;
3173                                     ;       This subtest verifies that the Reset FIFO function within
3174                                     ;       the Write Miscellaneous Control 1 function initializes
3175                                     ;       the FIFO to correct intial status. The following steps
3176                                     ;       are performed:
3177                                     ;               1. Reset an already initialized FIFO and check for
3178                                     ;                  proper status.
3179                                     ;               2. Write a varying number of bytes (1-65.) into the
3180                                     ;                  FIFO and verify that after each block of bytes is
3181                                     ;                  written the FIFO can be be reset to it's initial
3182                                     ;                  state.
3183                                     ;
```

```
                                ; 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
                                ;--
045344                          ;       BGNSUB                  ;////////////// BEGIN SUBTEST //////////
045344                                                                          T6.6:
045344  104402                                                                  TRAP    C$BSUB

                                ;       Write to TSSR register to soft initialize the controller
                                5$:
045346  004737  015774          ;       JSR     PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
045352  103405                  ;       BCS     10$             ;BR IF SOFT INIT OKAY
045354  010001                  ;       MOV     R0,R1           ;SAVE CONTENTS OF TSSR
045356                          ;       ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
045356  104455                                                                  TRAP    C$ERDF
045360  001201                                                                  .WORD   641
045362  003652                                                                  .WORD   SFIERR
045364  012034                                                                  .WORD   SFIMSG

                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
045366  005037  002222          10$:    CLR     FATFLG          ;CLEAR FATAL ERROR FLAG
045372  012704  050250          ;       MOV     @T17PACKET,R4   ;GET THE ADDRESS OF COMMAND PACKET
045376  004737  010662          ;       JSR     PC,WRTCHR       ;DO WRITE CHARACTERISTICS COMMAND
045402                          ;       FORCERROR       42$             ;@@DFORCE ERROR IF FORCER=1
045416  103407                  ;       BCS     50$             ;BR IF CARRY SET (GOOD RETURN)
045420  010001                  ;       MOV     R0,R1           ;SAVE CONTENTS OF TSSR
045422                          ;       NEXT.ERRNO
045422                          42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
045422  104455                                                                  TRAP    C$ERDF
045424  001202                                                                  .WORD   642
045426  046675                                                                  .WORD   T17SSR
045430  012046                                                                  .WORD   PKTSSR
045432  005237  002222          ;       INC     FATFLG          ;SET FATAL ERROR FLAG
045436                          50$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
045436  104406                                                                  TRAP    C$CLP1

                                ;       Do a Write Subsystem Write Misc to Reset FIFO
045440  004737  050054          ;       JSR     PC,T17RSFIF     ;SETUP PKT FOR WRITE MISC RESET FIFO
045444  012704  050420          ;       MOV     @T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
045450  010465  000000          ;       MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS TO EXECUTE
045454  004737  016336          ;       JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
045460                          ;       FORCERROR       62$             ;@@DFORCE ERROR IF FORCER=1
045474  103407                  ;       BCS     70$             ;BR IF CARRY SET (GOOD RETURN)
```

TEST 6: SUBTEST 6: FIFO VERIFY RESET FIFO TEST

```
3230 045476  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3231 045500                                 NEXT.ERRNO
3232 045500                          62$:   ERRDF   ERRNO,T172SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045500  104455                                                            TRAP    C$ERDF
     045502  001203                                                            .WORD   643
     045504  046732                                                            .WORD   T172SSR
     045506  012046                                                            .WORD   PKTSSR
3233 045510  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
3234 045514                          70$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045514  104406                                                            TRAP    C$CLP1
3235
3236                                 ;      Do a Write Subsystem READ STATUS
3237                                 ;      If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
3238                                 ;        signals NOT=0 Then Print Error
3239 045516  004737  050034                JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
3240 045522  012704  050420                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3241 045526  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3242 045532  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3243 045536                                FORCERROR       77$                     ;@@DFORCE ERROR IF FORCER=1
3244 045552  103407                        BCS     80$                     ;BR IF CARRY SET (GOOD RETURN)
3245 045554  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3246 045556                                 NEXT.ERRNO
3247 045556                          77$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045556  104455                                                            TRAP    C$ERDF
     045560  001204                                                            .WORD   644
     045562  046776                                                            .WORD   T173SSR
     045564  012046                                                            .WORD   PKTSSR
3248 045566  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
3249 045572                          80$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045572  104406                                                            TRAP    C$CLP1
3250 045574  004737  050216                JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
3251 045600  012701  046452                MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
3252 045604  012702  050312                MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
3253 045610  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
3254 045612  042711  002000                BIC     #S1.ICER,(R1)           ;SET EXPD ICER  =0
3255 045616  042711  001000                BIC     #S1.IFMK,(R1)           ;SET EXPD IFMK  =0
3256 045622  042711  000400                BIC     #S1.IHER,(R1)           ;SET EXPD IHER  =0
3257 045626  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
3258 045634  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
3259 045636  012701  050272                MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
3260 045642  012702  046432                MOV     #T17EXP,R2              ;EXPD ADDRESS
3261 045646  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
3262 045652  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3263 045656                                FORCERROR       92$,NOTSSR              ;@@D
3264 045666  103404                        BCS     100$                    ;BR IF YES
3265 045670                                NEXT.ERRNO
3266 045670                          92$:   ERRHRD  ERRNO,T177CMP,MSGSTAT   ;REPORT ERROR
     045670  104456                                                            TRAP    C$ERHRD
     045672  001205                                                            .WORD   645
     045674  047724                                                            .WORD   T177CMP
     045676  012350                                                            .WORD   MSGSTAT
3267 045700                          100$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045700  104406                                                            TRAP    C$CLP1
3268
3269                                 ;      Do a Write Subsystem WRITE NPR to set tape direction out
3270 045702  012700  000100                MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
3271 045706  004737  050076                JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
```

TEST 6: SUBTEST 6: FIFO VERIFY RESET FIFO TEST

```
3272 045712  012704  050420                MOV     @T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3273 045716  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3274 045722  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3275 045726                                FORCERROR       112$                    ;@@DFORCE ERROR IF FORCER=1
3276 045742  103407                        BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
3277 045744  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3278 045746                                NEXT.ERRNO
3279 045746                        112$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045746  104455                                                                TRAP    C$ERDF
     045750  001206                                                                .WORD   646
     045752  047043                                                                .WORD   T174SSR
     045754  012046                                                                .WORD   PKTSSR
3280 045756  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
3281 045762                        120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045762  104406                                                                TRAP    C$CLP1
3282
3283                               ;       Setup incrementing pattern in FIFO data buffer
3284 045764  012701  046452                MOV     @T17EXSTA,R1            ;EXPD WRITE FIFO DATA BUFFER
3285 045770  012702  000100                MOV     @64.,R2                 ;TEST PATTERN SIZE
3286 045774  005000                        CLR     R0                      ;INCREMENT TEST PATTERN
3287 045776  110021                130$:   MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
3288 046000  005200                        INC     R0                      ;SET NEXT PATTERN
3289 046002  005302                        DEC     R2                      ;DONE?
3290 046004  003374                        BGT     130$                    ;BR IF NO
3291
3292                               ;  REPEAT FOR BYTE COUNT 1 TO 65.
3293 046006  012737  000001  002310        MOV     @1,COUNT                ;GET FIRST BYTE COUNT
3294                               ;       Do a Write Subsystem WRITE FIFO with the current byte count
3295 046014                        150$:                                   ;REPEAT LOOP LABEL
3296 046014  013700  002310                MOV     COUNT,R0                ;FIFO BYTE COUNT
3297 046020  012701  046452                MOV     @T17EXSTA,R1            ;FIFO WRITE DATA ADDRESS
3298 046024  004737  050122                JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
3299 046030  012704  050420                MOV     @T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3300 046034  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3301 046040  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3302 046044                                FORCERROR       152$                    ;@@DFORCE ERROR IF FORCER=1
3303 046060  103407                        BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
3304 046062  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3305 046064                                NEXT.ERRNO
3306 046064                        152$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     046064  104455                                                                TRAP    C$ERDF
     046066  001207                                                                .WORD   647
     046070  047106                                                                .WORD   T175SSR
     046072  012046                                                                .WORD   PKTSSR
3307 046074  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
3308 046100                        160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     046100  104406                                                                TRAP    C$CLP1
3309
3310                               ;       Do a Write Subsystem Write Misc to Reset FIFO
3311 046102  004737  050054                JSR     PC,T17RSFIF             ;SETUP PKT FOR WRITE MISC RESET FIFO
3312 046106  012704  050420                MOV     @T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3313 046112  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3314 046116  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3315 046122                                FORCERROR       162$                    ;@@DFORCE ERROR IF FORCER=1
3316 046136  103407                        BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
3317 046140  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3318 046142                                NEXT.ERRNO
```

```
3319 046142                          162$:  ERRDF   ERRNO,T172SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     046142  104455                                                                TRAP    C$ERDF
     046144  001210                                                                .WORD   648
     046146  046732                                                                .WORD   T172SSR
     046150  012046                                                                .WORD   PKTSSR
3320 046152  005237  002222                      INC     FATFLG                  ;SET FATAL ERROR FLAG
3321 046156                          170$:  CKLOOP                              ;LOOP ON ERROR, IF FLAG SET
     046156  104406                                                                TRAP    C$CLP1
3322
3323                                 ;      Do a Write Subsystem READ STATUS
3324                                 ;      If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
3325                                 ;        signals NOT=0 Then Print Error
3326 046160  004737  050034                      JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
3327 046164  012704  050420                      MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3328 046170  010465  000000                      MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3329 046174  004737  016336                      JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3330 046200                                      FORCERROR       177$                    ;@@DFORCE ERROR IF FORCER=1
3331 046214  103407                              BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
3332 046216  010001                              MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3333 046220                                      NEXT.ERRNO
3334 046220                          177$:  ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     046220  104455                                                                TRAP    C$ERDF
     046222  001211                                                                .WORD   649
     046224  046776                                                                .WORD   T173SSR
     046226  012046                                                                .WORD   PKTSSR
3335 046230  005237  002222                      INC     FATFLG                  ;SET FATAL ERROR FLAG
3336 046234                          180$:  CKLOOP                              ;LOOP ON ERROR, IF FLAG SET
     046234  104406                                                                TRAP    C$CLP1
3337 046236  004737  050216                      JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
3338 046242  012701  046452                      MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
3339 046246  012702  050312                      MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
3340 046252  011211                              MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
3341 046254  042711  002000                      BIC     #S1.ICER,(R1)           ;SET EXPD ICER  =0
3342 046260  042711  001000                      BIC     #S1.IFMK,(R1)           ;SET EXPD IFMK  =0
3343 046264  042711  000400                      BIC     #S1.IHER,(R1)           ;SET EXPD IHER  =0
3344 046270  016261  000002  000002              MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
3345 046276  005000                              CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
3346 046300  012701  050272                      MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
3347 046304  012702  046432                      MOV     #T17EXP,R2              ;EXPD ADDRESS
3348 046310  012703  000024                      MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
3349 046314  004737  011500                      JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3350 046320                                      FORCERROR       192$,NOTSSR     ;@@D
3351 046330  103404                              BCS     200$                    ;BR IF YES
3352 046332                                      NEXT.ERRNO
3353 046332                          192$:  ERRHRD  ERRNO,T177CMP,MSGSTAT   ;REPORT ERROR
     046332  104456                                                                TRAP    C$ERHRD
     046334  001212                                                                .WORD   650
     046336  047724                                                                .WORD   T177CMP
     046340  012350                                                                .WORD   MSGSTAT
3354 046342                          200$:  CKLOOP                              ;LOOP ON ERROR, IF FLAG SET
     046342  104406                                                                TRAP    C$CLP1
3355
3356
3357 046344                          250$:
3358 046344                                      FORCEXIT        260$
3359 046354  005237  002310                      INC     COUNT                   ;GET NEXT BYTE COUNT
3360 046360  023727  002310  000101              CMP     COUNT,#65.              ;DONE ALL BYTES?
```

```
3361 046366  101002                        BHI     260$                ;BR IF YES
3362 046370  000137  046014                JMP     150$                ;DO ANOTHER BYTE COUNT
3363 046374                          260$:
3364
3365 046374                                ENDSUB                      ;////////////// END SUBTEST //////////
     046374                                                                      L10064:
     046374  104403                                                                TRAP    C$ESUB
3366
3367 046376  005737  002222                TST     FATFLG              ;ANY FATAL ERRORS ?
3368 046402  001402                        BEQ     300$                ;BRANCH IF NOT
3369 046404  004737  017202                JSR     PC,CKDROP           ;TRY TO DROP THE UNIT
3370 046410  004737  016456          300$: JSR     PC,TSTLOOP          ;DO ITERATIONS?
3371 046414  103002                        BCC     305$                ;BR IF NO
3372 046416  000137  040454                JMP     T17LOOP             ;LOOP UNTIL ITERATIONS DONE
3373 046422                          305$:
3374
3375 046422                                EXIT    TST                 ;////////// EXIT TEST //////////
     046422  104432                                                            TRAP    C$EXIT
     046424  002112                                                            .WORD   L10056-.
3376
3377
3378                                 ;+
3379                                 ;LOCAL STORAGE FOR THIS TEST
3380                                 ;-
3381
3382
3383 046426                          T17MSK:                             ;MASK OF UNTESTED BITS IN READ STATUS BYTES
3384                                                                     ;UNTESTED BITS ARE SET TO 1
3385 046426      377                         .BYTE   ^C<000>             ;BYTE 0 MASK
3386 046427      037                         .BYTE   ^C<340>             ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
3387 046430      360                         .BYTE   ^C<017>             ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
3388 046431      000                         .BYTE   0                   ;MAKE IT EVEN
3389
3390 046432                          T17EXP:                             ;BEGIN EXPECTED DATA BUFFER
3391 046432  000000                         .WORD   0                   ;MESSAGE TYPE
3392 046434  000000                         .WORD   0                   ;DATA FIELD LENGTH
3393 046436  000000                         .WORD   0                   ;RBPCR
3394 046440  000000                         .WORD   0                   ;XST0
3395 046442  000000                         .WORD   0                   ;XST1
3396 046444  000000                         .WORD   0                   ;XST2
3397 046446  000000                         .WORD   0                   ;XST3
3398 046450  000000                         .WORD   0                   ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
3399 046452                          T17EXSTA: .BLKB 66.                ;EXPECTED READ STATUS AND WRITE FIFO DATA
3400 046554                          T17EXEND:                                   ;END EXPECTED DATA BUFFER
3401
3402 046554                          T17WFDATA: .BLKB 66.                ;WRITE FIFO EXPECTED DATA BUFFER
3403
3404                                 ;+
3405                                 ;LOCAL TEXT MESSAGES FOR TEST
3406                                 ;-
3407
3408 046656  106  111  106  TST17ID:         .ASCIZ  'FIFO Exerciser'
3409 046675  127  122  111  T17SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
3410 046732  127  122  111  T172SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
3411 046776  127  122  111  T173SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
3412 047043  127  122  111  T174SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
3413 047106  127  122  111  T175SSR:.ASCIZ  'WRITE SUBSYSTEM (Write FIFO) Failed'
```

```
3414 047152     127     122     111  T176SSR:.ASCIZ  'WRITE SUBSYSTEM (Read FIFO) Failed'
3415 047215     106     111     106  T171CMP:.ASCIZ  'FIFO Status in WORD #9 Incorrect after Initialize'
3416 047277     122     145     141  T172CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
3417 047373     106     111     106  T173CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after WRITE FIFO'
3418 047457     106     111     106  T174CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after READ FIFO'
3419 047542     122     145     141  T175CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO Data'
3420 047616     106     111     106  T176CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after READ FIFO from an Empty FIFO'
3421 047724     106     111     106  T177CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after RESET FIFO'
3422                                              .EVEN
3423
3424                                  ;+
3425                                  ; CLEAR MESSAGE BUFFER
3426                                  ;-
3427 050010                           T17CLRBUF:                                      ;
3428 050010                                   SAVREG                                  ;SAVE R1-R5 UNTIL NEXT RETURN
3429 050014  012701  050272                   MOV     #T17BFR,R1                      ;GET MESSAGE BUFFER ADDRESS
3430 050020  012702  000120                   MOV     #T17BEND-T17BFR,R2              ;SIZE OF MESSAGE BUFFER IN BYTES
3431 050024  105021                   10$:    CLRB    (R1)+                           ;CLEAR A BYTE
3432 050026  005302                           DEC     R2                              ;DONE?
3433 050030  003375                           BGT     10$                             ;BR IF NO
3434 050032  000207                           RTS     PC                              ;RETURN
3435
3436                                  ;+
3437                                  ; SETUP T17PK2 PACKET FOR READ STATUS
3438                                  ;-
3439 050034                           T17SRD:
3440 050034  004737  050010                   JSR     PC,T17CLRBUF                    ;CLEAR MESSAGE BUFFER
3441 050040  012700  050430                   MOV     #T17DT2,R0                      ;WRITE SUBSYSTEM DATA BUFFER
3442 050044  112720  000005                   MOVB    #PW.RDSTATUS,(R0)+              ;STORE READ STATUS COMMAND IN BSEL0
3443 050050  105010                           CLRB    (R0)                            ;CLEAR BSEL1
3444 050052  000207                           RTS     PC                              ;RETURN
3445
3446                                  ;+
3447                                  ; SETUP T17PK2 PACKET FOR WRITE MISC RESET FIFO
3448                                  ;-
3449 050054                           T17RSFIF:
3450 050054  004737  050010                   JSR     PC,T17CLRBUF                    ;CLEAR MESSAGE BUFFER
3451 050060  012700  050430                   MOV     #T17DT2,R0                      ;WRITE SUBSYSTEM DATA BUFFER
3452 050064  112720  000010                   MOVB    #PW.WMISC,(R0)+                 ;STORE WRITE MISCELLANEOUS IN BSEL0
3453 050070  112710  000030                   MOVB    #MS.RSFIF!MS.RSTAP,(R0)         ;STORE BSEL1 CLEAR FIFO CODES
3454 050074  000207                           RTS     PC                              ;RETURN
3455
3456                                  ;+
3457                                  ; SETUP T17PK2 PACKET FOR WRITE NPR
3458                                  ;
3459                                  ; INPUT:
3460                                  ;       R0 CONTAINS BSEL1 NPR DATA
3461                                  ;
3462                                  ;       SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
3463                                  ;-
3464 050076                           T17SNPR:
3465 050076  004737  050010                   JSR     PC,T17CLRBUF                    ;CLEAR MESSAGE BUFFER
3466 050102  012701  050430                   MOV     #T17DT2,R1                      ;WRITE SUBSYSTEM DATA BUFFER
3467 050106  112721  000011                   MOVB    #PW.WNPR,(R1)+                  ;STORE WRITE NPR IN BSEL0
3468 050112  052700  000020                   BIS     #NP.WRP,R0                      ;DON'T WRITE WRONG PARITY
3469 050116  110011                           MOVB    R0,(R1)                         ;STORE NPR DATA IN BSEL1
3470 050120  000207                           RTS     PC                              ;RETURN
```

```
3471
3472                                    ;+
3473                                    ; SETUP T17PK2 PACKET FOR WRITE FIFO
3474                                    ;
3475                                    ; INPUT:
3476                                    ;       R0 CONTAINS BYTE COUNT
3477                                    ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
3478                                    ;-
3479 050122                            T17WFIF:
3480 050122                                    SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
3481 050126  004737  050010                    JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
3482 050132  012702  050430                    MOV     #T17DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
3483 050136  112722  000004                    MOVB    #PW.WFIFO,(R2)+         ;STORE WRITE FIFO IN BSEL0
3484 050142  110022                            MOVB    R0,(R2)+                ;STORE BYTE COUNT IN BSEL1
3485 050144  005022                            CLR     (R2)+                   ;CLEAR SEL2 (UNUSED)
3486 050146  112122                    10$:    MOVB    (R1)+,(R2)+             ;STORE DATA PATTERN BYTE
3487 050150  005300                            DEC     R0                      ;DONE ALL BYTES?
3488 050152  003375                            BGT     10$                     ;BR IF NO
3489 050154  000207                            RTS     PC                      ;RETURN
3490
3491                                    ;+
3492                                    ; SETUP T17PK2 PACKET FOR READ FIFO
3493                                    ;
3494                                    ; INPUT:
3495                                    ;       R0 CONTAINS SEL2 BYTE COUNT
3496                                    ;-
3497 050156                            T17RFIF:
3498 050156  004737  050010                    JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
3499 050162  012701  050430                    MOV     #T17DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
3500 050166  112721  000003                    MOVB    #PW.RFIFO,(R1)+         ;STORE READ FIFO IN BSEL0
3501 050172  110021                            MOVB    R0,(R1)+                ;STORE BYTE COUNT IN BSEL1
3502 050174  000207                            RTS     PC                      ;RETURN
3503
3504                                    ;+
3505                                    ; CLEAR EXPECTED DATA MESSAGE BUFFER
3506 050176                            ;-
3507 050176  012701  046432            T17CLEXP:
3508 050202  012700  000122                    MOV     #T17EXP,R1              ;GET EXPD ADDRESS
3509 050206  105021                            MOV     #T17EXEND-T17EXP,R0     ;GET EXPD SIZE
3510 050210  005300                    10$:    CLRB    (R1)+                   ;CLEAR A BYTE
3511 050212  003375                            DEC     R0                      ;DONE?
3512 050214  000207                            BGT     10$                     ;BR IF NO
3513                                           RTS     PC                      ;RETURN
3514                                    ;+
3515                                    ;Set WORDS 0-7 of expd message buffer = to recv since not testing
3516                                    ;-
3517 050216                            T17SETEXP:
3518 050216  012702  046432                    MOV     #T17EXP,R2              ;GET EXPD
3519 050222  012703  050272                    MOV     #T17BFR,R3              ;GET READ STATUS RECV BUFFER
3520 050226  012700  000010                    MOV     #8.,R0                  ;SET WORDS 0-7 EXP=RECV
3521 050232  012322                    5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
3522 050234  005300                            DEC     R0                      ;DONE WORDS 0-7 WORDS?
3523 050236  003375                            BGT     5$                      ;BR IF NO
3524 050240  000207                            RTS     PC                      ;RETURN
3525
3527         050250                            .=<.+10>&177770
3529                                    ;
```

```
3530                                    ;WRITE CHARACTERISTICS COMMAND PACKET
3531                                    ;
3532 050250                             T17PACKET:                              ;COMMAND PACKET FOR TEST
3533 050250  100004                             .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
3534 050252  050260                             .WORD   T17DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
3535 050254  000000                             .WORD   0
3536 050256  000012                             .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
3537
3538 050260                             T17DATA:                                ;CHARACTERISTICS DATA BLOCK
3539 050260  050272                             .WORD   T17BFR                  ;ADDRESS OF MESSAGE BUFFER
3540 050262  000000                             .WORD   0
3541 050264  000024                             .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
3542 050266  000000                             .WORD   0                       ;ESS,ENB,EAI,ERI
3543 050270  000000                             .WORD   0                       ;EXTENDED FEATURES UNIT NO. ETC.
3544
3545
3546                                    ;MESSAGE BUFFER FOR ALL TEST 6 COMMANDS
3547
3548 050272                             T17BFR:                                 ;BEGIN MESSAGE BUFFER
3549 050272  000000                             .WORD   0                       ;MESSAGE TYPE
3550 050274  000000                             .WORD   0                       ;DATA FIELD LENGTH
3551 050276  000000                             .WORD   0                       ;RBPCR
3552 050300  000000                             .WORD   0                       ;XST0
3553 050302  000000                             .WORD   0                       ;XST1
3554 050304  000000                             .WORD   0                       ;XST2
3555 050306  000000                             .WORD   0                       ;XST3
3556 050310  000000                             .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
3557 050312                             T17BFSTA: .BLKB 64.                     ;READ STATUS AND WRITE FIFO BUFFER
3558 050412                             T17BEND:                                ;END OF MESSAGE BUFFER
3559                                    ;
3560                                    ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
3561                                    ;
3563         050420                             .=<.+10>&177770
3565 050420                             T17PK2:
3566 050420  100006                             .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
3567 050422  050430                             .WORD   T17DT2                  ;LOW ADDRESS OF DATA BLOCK
3568 050424  000000                             .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
3569 050426  000012                             .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
3570
3571 050430                             T17DT2:                                 ;DATA BLOCK
3572 050430     000                             .BYTE   0                       ;BSEL0
3573 050431     000                             .BYTE   0                       ;BSEL1
3574 050432  000000                             .WORD   0                       ;SEL2
3575 050434                                     .BLKB   66.                     ;WRITE FIFO DATA OUTPUT BUFFER
3576
3577 050536                                     ENDTST
     050536                                                             L10056:
     050536  104401                                                             TRAP    C$ETST
3578
3579                                            .SBTTL  TEST  7: STATIC TRANSPORT BUS INTERFACE TEST
3580                                    ;*
3581                                    ; TEST DESCRIPTION:
3582                                    ;
3583                                    ;
3584                                    ; TEST STEPS:
3585                                    ;
3586                                    ;    REPEAT FOR LOOPCNT
```

```
TEST  7: STATIC TRANSPORT BUS INTERFACE TEST

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

                                                BGNTST
                                                                                                        T7::
                                                MOV     #TST18ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
                                                JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
                                                MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
                                        T18LOOP:
                                        ;       Write to TSSR register to soft initialize the controller
                                        5$:
                                                JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
                                                BCS     10$                     ;BR IF SOFT INIT OKAY
                                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
                                                                                                        TRAP    C$ERDF
                                                                                                        .WORD   700
                                                                                                        .WORD   SFIERR
                                                                                                        .WORD   SFIMSG

                                        ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
                                        10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
                                                MOV     #T18PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
                                                JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
                                                FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
                                                BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
                                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                                NEXT.ERRNO
                                        12$:    ERRDF   ERRNO,T18SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                                                                                                        TRAP    C$ERDF
                                                                                                        .WORD   701
                                                                                                        .WORD   T18SSR
                                                                                                        .WORD   PKTSSR
                                                INC     FATFLG                  ;SET FATAL ERROR FLAG
                                        15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                                        TRAP    C$CLP1

                                        ;       If Extended Features Hardware Switch Clear then:
                                        ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
                                                MOV     #T18BFR,R1              ;MESSAGE BUFFER ADDRESS
                                                BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
                                                BNE     30$                     ;BR IF YES
                                                JSR     PC,T18SMISC             ;SETUP PACKET FOR WRITE MISCELLANEOUS
                                                MOV     #T18PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
                                                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
                                                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
                                                FORCERROR       22$                     ;@@DFORCE ERROR IF FORCER=1
                                                BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```
3638 050722  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3639 050724                                 NEXT.ERRNO
3640 050724                          22$:   ERRDF   ERRNO,T182SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     050724  104455                                                                TRAP    C$ERDF
     050726  001276                                                                .WORD   702
     050730  051342                                                                .WORD   T182SSR
     050732  012046                                                                .WORD   PKTSSR
3641 050734  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
3642 050740                          30$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     050740  104406                                                                TRAP    C$CLP1
3643
3644
3645                                 ;      Do WRITE CHARACTERISTICS to select reserved unit 7
3646 050742  005037  002222                 CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
3647 050746  012704  051730                 MOV     #T18PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3648 050752  004737  010662                 JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
3649 050756                                 FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
3650 050772  103407                        BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
3651 050774  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3652 050776                                 NEXT.ERRNO
3653 050776                          42$:   ERRDF   ERRNO,T18SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     050776  104455                                                                TRAP    C$ERDF
     051000  001277                                                                .WORD   703
     051002  051305                                                                .WORD   T18SSR
     051004  012046                                                                .WORD   PKTSSR
3654 051006  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
3655 051012                          50$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     051012  104406                                                                TRAP    C$CLP1
3656
3657                                 ;      Clear message buffer
3658 051014  012701  051752                 MOV     #T18BFR,R1              ;GET MESSAGE BUFFER ADDRESS
3659 051020  013700  051744                 MOV     T18DATA+4,R0            ;SIZE OF MESSAGE BUFFER IN BYTES
3660 051024  105021                  60$:   CLRB    (R1)+                   ;CLEAR A BYTE
3661 051026  005300                        DEC     R0                      ;DONE?
3662 051030  003375                        BGT     60$                     ;BR IF NO
3663                                 ;      Do a Write Subsystem READ STATUS
3664 051032  004737  051556                 JSR     PC,T18SRD               ;SETUP PACKET FOR READ STATUS
3665 051036  012704  052000                 MOV     #T18PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3666 051042  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3667 051046  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3668 051052                                 FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
3669 051066  103407                        BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
3670 051070  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3671 051072                                 NEXT.ERRNO
3672 051072                          62$:   ERRDF   ERRNO,T183SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     051072  104455                                                                TRAP    C$ERDF
     051074  001300                                                                .WORD   704
     051076  051406                                                                .WORD   T183SSR
     051100  012046                                                                .WORD   PKTSSR
3673 051102  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
3674 051106                          70$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     051106  104406                                                                TRAP    C$CLP1
3675
3676
3677                                 ;      Set first 8 words of expd message buffer = to recv since not testing
3678                                 ;      Set unused bits in Read Status expd equal rcvd
3679 051110  004737  051620                 JSR     PC,T18SETEXP            ;SET SOME EXPD TO RECV
```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```
3680                                    ;       If any transport interface signals are asserted then Print Error
3681 051114  005000                             CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
3682 051116  012701  051752                     MOV     #T18BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
3683 051122  012702  051216                     MOV     #T18EXP,R2              ;EXPD ADDRESS
3684 051126  012703  000012                     MOV     #10.,R3                 ;NUMBER OF WORDS TO COMPARE
3685 051132  004737  011500                     JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3686 051136                                     FORCERROR       82$,NOTSSR      ;@@D
3687 051146  103404                            BCS     90$                     ;BR IF YES
3688 051150                                     NEXT.ERRNO
3689 051150                             82$:    ERRHRD  ERRNO,T18CMP,MSGSTAT    ;REPORT ERROR
     051150  104456                                                                    TRAP    C$ERHRD
     051152  001301                                                                    .WORD   705
     051154  051453                                                                    .WORD   T18CMP
     051156  012350                                                                    .WORD   MSGSTAT
3690 051160                             90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     051160  104406                                                                    TRAP    C$CLP1
3691
3692 051162  005737  002222                     TST     FATFLG                  ;ANY FATAL ERRORS ?
3693 051166  001402                            BEQ     160$                    ;BRANCH IF NOT
3694 051170  004737  017202                     JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
3695 051174  004737  016456             160$:   JSR     PC,TSTLOOP              ;DO ITERATIONS?
3696 051200  103002                            BCC     165$                    ;BR IF NO
3697 051202  000137  050556                     JMP     T18LOOP                 ;LOOP UNTIL ITERATIONS DONE
3698 051206                             165$:
3699 051206                                     EXIT    TST
     051206  104432                                                                    TRAP    C$EXIT
     051210  000606                                                                    .WORD   L10065-.
3700
3701
3702                                    ;*
3703                                    ;LOCAL STORAGE FOR THIS TEST
3704                                    ;-
3705
3706 051212                             T18MSK:                                 ;MASK OF UNUSED BITS IN READ STATUS BYTES
3707 051212     377                             .BYTE   ^C<000>                 ;BYTE 0 MASK
3708 051213     037                             .BYTE   ^C<340>                 ;BYTE 1
3709 051214     100                             .BYTE   ^C<277>                 ;BYTE 2
3710 051215     000                             .BYTE   0                       ;MAKE IT EVEN
3711
3712 051216                             T18EXP:                                 ;EXPECTED DATA BUFFER
3713 051216  000000                             .WORD   0                       ;MESSAGE TYPE
3714 051220  000000                             .WORD   0                       ;DATA FIELD LENGTH
3715 051222  000000                             .WORD   0                       ;RBPCR
3716 051224  000000                             .WORD   0                       ;XST0
3717 051226  000000                             .WORD   0                       ;XST1
3718 051230  000000                             .WORD   0                       ;XST2
3719 051232  000000                             .WORD   0                       ;XST3
3720 051234  000000                             .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUB)
3721 051236  000000                             .WORD   0                       ;READ STATUS BYTE 1/0
3722 051240  000000                             .WORD   0                       ;READ STATUS BYTE 2
3723
3724 051242     377     020             T18XS:  .BYTE   377,020                 ;READ STATUS BYTE 0/1 EXPECTED BASE
3725 051244  000000                             .WORD   0                       ;READ STATUS BYTE 2 EXPECTED BASE
3726
3727                                    ;*
3728                                    ;LOCAL TEXT MESSAGES FOR TEST
3729                                    ;-
```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```
3730
3731 051246     123     164     141  TST18ID:        .ASCIZ  'Static Transport Bus Interface'
3732 051305     127     122     111  T18SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
3733 051342     127     122     111  T182SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
3734 051406     127     122     111  T183SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
3735 051453     124     162     141  T18CMP: .ASCIZ  'Transport Bus Interface Signals NOT Negated After Unit 7 Selected'
3736                                                 .EVEN
3737
3738                                 ;+
3739                                 ; SETUP T18PK2 PACKET FOR READ STATUS
3740                                 ;-
3741 051556                         T18SRD:
3742 051556                                 SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
3743 051562  012700  052010                MOV     #T18DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
3744 051566  112720  000005                MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
3745 051572  105010                        CLRB    (R0)                    ;CLEAR BSEL1
3746 051574  000207                        RTS     PC                      ;RETURN
3747
3748                                 ;+
3749                                 ; SETUP T18PK2 PACKET FOR WRITE MISC.
3750                                 ;-
3751 051576                         T18SMISC:
3752 051576                                 SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
3753 051602  012700  052010                MOV     #T18DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
3754 051606  112720  000010                MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
3755 051612  112710  000200                MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
3756 051616  000207                        RTS     PC                      ;RETURN
3757
3758                                 ;+
3759                                 ;Set first 8 words of expd message buffer = to recv since not testing
3760                                 ;       Set unused bits in Read Status expd equal rcvd
3761                                 ;-
3762 051620                         T18SETEXP:
3763 051620  012702  051216                MOV     #T18EXP,R2              ;GET EXPD
3764 051624  012703  051752                MOV     #T18BFR,R3              ;GET READ STATUS RECV BUFFER
3765 051630  012700  000010                MOV     #8.,R0                  ;SET FIRST 8 WORDS EXP=RECV
3766 051634  012322                5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
3767 051636  005300                        DEC     R0                      ;DONE FIRST 8 WORDS?
3768 051640  003375                        BGT     5$                      ;BR IF NO
3769 051642  012701  051212                MOV     #T18MSK,R1              ;GET UNUSED BIT MASK
3770 051646  013712  051242                MOV     T18XS,(R2)              ;SETUP BASE EXPECTED BYTE 1/0
3771 051652  013762  051244  000002        MOV     T18XS+2,2(R2)           ;SETUP BASE EXPECTED BYTE 2
3772 051660  011300                        MOV     (R3),R0                 ;GET RECV BYTE 1 AND BYTE 0
3773 051662  041100                        BIC     (R1),R0                 ;CLEAR ALL BUT UNUSED
3774 051664  040012                        BIC     R0,(R2)                 ;CLEAR UNUSED IN EXP
3775 051666  050012                        BIS     R0,(R2)                 ;SET UNUSED EXPD=RECV FOR COMPARE
3776 051670  016300  000002                MOV     2(R3),R0                ;GET RECV BYTE 2
3777 051674  046100  000002                BIC     2(R1),R0                ;CLEAR ALL BUT UNUSED
3778 051700  040062  000002                BIC     R0,2(R2)                ;CLEAR UNUSED IN EXPD
3779 051704  050062  000002                BIS     R0,2(R2)                ;SET UNUSED EXPD=RECV FOR COMPARE
3780 051710  105062  000003                CLRB    3(R2)                   ;CLEAR EXPD BYTE 3 (UNUSED)
3781 051714  105063  000003                CLRB    3(R3)                   ;CLEAR RECV BYTE 3 (UNUSED)
3782 051720  000207                        RTS     PC                      ;RETURN
3783
3785         051730                        .=<.+10>&177770
3787                                 ;
3788                                 ;WRITE CHARARTERISTICS COMMAND PACKET
```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```
3789                                    ;
3790 051730                             T18PACKET:                      ;COMMAND PACKET FOR TEST
3791 051730  100004                             .WORD   100004          ;WRITE CHARACTERISTICS COMMAND, WITH ACK
3792 051732  051740                             .WORD   T18DATA         ;ADDRESS OF CHARACTERISTICS BLOCK
3793 051734  000000                             .WORD   0
3794 051736  000012                             .WORD   10.             ;MESSAGE PACKET MINIMUM SIZE
3795
3796 051740                             T18DATA:                        ;CHARACTERISTICS DATA BLOCK
3797 051740  051752                             .WORD   T18BFR          ;ADDRESS OF MESSAGE BUFFER
3798 051742  000000                             .WORD   0
3799 051744  000024                             .WORD   20.             ;LENGTH OF MESSAGE BUFFER
3800 051746  000000                             .WORD   0               ;ESS,ENB,EAI,ERI
3801 051750  000007                             .WORD   7               ;SELECT RESERVED UNIT 7
3802
3803
3804 051752                             T18BFR:                         ;MESSAGE BUFFER
3805 051752  000000                             .WORD   0               ;MESSAGE TYPE
3806 051754  000000                             .WORD   0               ;DATA FIELD LENGTH
3807 051756  000000                             .WORD   0               ;RBPCR
3808 051760  000000                             .WORD   0               ;XST0
3809 051762  000000                             .WORD   0               ;XST1
3810 051764  000000                             .WORD   0               ;XST2
3811 051766  000000                             .WORD   0               ;XST3
3812 051770  000000                             .WORD   0               ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
3813 051772  000000                             .WORD   0               ;READ STATUS BYTE 1/O RETURNED
3814 051774  000000                             .WORD   0               ;READ STATUS BYTE 2
3815                                    ;
3816                                    ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
3817                                    ;
3819         052000                             .=<.+10>&177770
3821 052000                             T18PK2:
3822 052000  100006                             .WORD   P.WRTSUB!P.ACK  ;WRITE SUBSYSTEM WITH ACK
3823 052002  052010                             .WORD   T18DT2          ;LOW ADDRESS OF DATA BLOCK
3824 052004  000000                             .WORD   0               ;HIGH ADDRESS OF DATA BLOCK
3825 052006  000010                             .WORD   8.              ;BUFFER EXTENT
3826
3827 052010                             T18DT2:                         ;DATA BLOCK
3828 052010     000                             .BYTE   0               ;BSEL0
3829 052011     000                             .BYTE   0               ;BSEL1
3830 052012  000000                             .WORD   0               ;SEL2
3831 052014  000000                             .WORD   0               ;DATA
3832
3833
3834 052016                                     ENDTST
     052016                                                             L10065:
     052016  104401                                                             TRAP    C$ETST
3835                                            .SBTTL  TEST  8: TRANSPORT BUS INTERFACE LOOPBACK TEST
3836                                    ;++
3837                                    ; TEST DESCRIPTION:
3838                                    ;
3839                                    ;       This test verifies the controller's Transport Bus
3840                                    ;       drivers, receivers, and signal loopback logic. Note
3841                                    ;       that the Static Transport Bus test must have run
3842                                    ;       correctly for this test to provide meaningful results.
3843                                    ;
3844                                    ; TEST STEPS:
3845                                    ;
```

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
052020                                                                                  T8::
052020
052020  012700  060232                          MOV     #TST19ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
052024  004737  016510                          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
052030  012737  000012  002216                  MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
052036                                  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,ITADO,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
                                        ;--
052036                                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
052036                                                                                  T8.1:
052036  104402                                                                          TRAP    C$BSUB

                                        ;       Write to TSSR register to soft initialize the controller
052040                                  5$:
052040  004737  015774                          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
052044  103405                                  BCS     10$                     ;BR IF SOFT INIT OKAY
052046  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
052050                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
052050  104455                                                                          TRAP    C$ERDF
052052  001440                                                                          .WORD   800
052054  003652                                                                          .WORD   SFIERR
052056  012034                                                                          .WORD   SFIMSG

                                        ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
052060  005037  002222                  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
052064  012704  062360                          MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
052070  004737  010662                          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
052074                                          FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
052110  103407                                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
052112  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
052114                                          NEXT.ERRNO
052114                                  12$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
052114  104455                                                                          TRAP    C$ERDF
052116  001441                                                                          .WORD   801
052120  060273                                                                          .WORD   T19SSR
052122  012046                                                                          .WORD   PKTSSR
052124  005237  002222                          INC     FATFLG                  ;SET FATAL ERROR FLAG
052130                                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
052130  104406                                                                          TRAP    C$CLP1

                                        ;       If Extended Features Hardware Switch Clear then:
                                        ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
052132  012701  062402                          MOV     #T19BFR,R1              ;MESSAGE BUFFER ADDRESS
052136  032761  000200  000012                  BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
052144  001026                                  BNE     30$                     ;BR IF YES
052146  004737  062232                          JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
052152  012704  062530                          MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
052156  010465  000000                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
052162  004737  016336                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
052166                                          FORCERROR       22$                     ;@@DFORCE ERROR IF FORCER=1
052202  103407                                  BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
052204  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
052206                                          NEXT.ERRNO
052206                                  22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
052206  104455                                                                          TRAP    C$ERDF
052210  001442                                                                          .WORD   802
052212  060330                                                                          .WORD   T192SSR
052214  012046                                                                          .WORD   PKTSSR
052216  005237  002222                          INC     FATFLG                  ;SET FATAL ERROR FLAG
052222                                  30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
      052222  104406                                                        TRAP    C$CLP1
3948                          ;       Do WRITE CHARACTERISTICS to select reserved unit 7
3949 052224  005037  002222           CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
3950 052230  012704  062360           MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3951 052234  004737  010662           JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
3952 052240                           FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
3953 052254  103407                   BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
3954 052256  010001                   MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3955 052260                           NEXT.ERRNO
3956 052260                   42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      052260  104455                                                        TRAP    C$ERDF
      052262  001443                                                        .WORD   803
      052264  060273                                                        .WORD   T19SSR
      052266  012046                                                        .WORD   PKTSSR
3957 052270  005237  002222           INC     FATFLG                  ;SET FATAL ERROR FLAG
3958 052274                   50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      052274  104406                                                        TRAP    C$CLP1
3959                          ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
3960 052276  012700  000100           MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
3961 052302  052700  000040           BIS     #NP.LOOP,R0             ;SET LOOPBACK ENABLE
3962 052306  004737  062072           JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
3963 052312  012704  062530           MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3964 052316  010465  000000           MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3965 052322  004737  016336           JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3966 052326                           FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
3967 052342  103407                   BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
3968 052344  010001                   MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3969 052346                           NEXT.ERRNO
3970 052346                   62$:    ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      052346  104455                                                        TRAP    C$ERDF
      052350  001444                                                        .WORD   804
      052352  060441                                                        .WORD   T194SSR
      052354  012046                                                        .WORD   PKTSSR
3971 052356  005237  002222           INC     FATFLG                  ;SET FATAL ERROR FLAG
3972 052362                   70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      052362  104406                                                        TRAP    C$CLP1
3973                          ;       Do Write Subsystem Write Control to CLEAR loopback signals group 1.
3974                          ;       Do Write Subsystem Write Format to CLEAR loopback signals group 2.
3975                          ;        (the loopback signals have to be cleared here due to the flip-flops
3976                          ;          that are set on a 1 to 0 transition (IHER,IFMK,ICER))
3977 052364  005000                   CLR     R0                      ;WRITE 0'S
3978 052366  042700  000200           BIC     #WC.IFAD,R0             ;IFAD MUST  ALWAYS =0
3979 052372  042700  000100           BIC     #WC.IOTAD,R0            ;ITADO MUST ALWAYS =0
3980 052376  042700  000040           BIC     #WC.I1TAD,R0            ;ITAD1 MUST ALWAYS =0
3981 052402  004737  062172           JSR     PC,T19WCTL              ;SETUP PACKET FOR WRITE CONTROL
3982 052406  012704  062530           MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3983 052412  010465  000000           MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3984 052416  004737  016336           JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3985 052422                           FORCERROR       82$                     ;@@DFORCE ERROR IF FORCER=1
3986 052436  103407                   BCS     90$                     ;BR IF CARRY SET (GOOD RETURN)
3987 052440  010001                   MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3988 052442                           NEXT.ERRNO
3989 052442                   82$:    ERRDF   ERRNO,T197SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      052442  104455                                                        TRAP    C$ERDF
      052444  001445                                                        .WORD   805
      052446  060613                                                        .WORD   T197SSR
      052450  012046                                                        .WORD   PKTSSR
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
3990 052452  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
3991 052456                        90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     052456  104406                                                        TRAP    C$CLP1
3992 052460  005000                        CLR     R0                      ;SET FORMAT DRIVE DATA=0
3993 052462  004737  062212                JSR     PC,T19WFMT              ;SETUP PACKET FOR WRITE FORMAT
3994 052466  012704  062530                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3995 052472  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3996 052476  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3997 052502                                FORCERROR       102$                    ;@@FORCE ERROR IF FORCER=1
3998 052516  103407                        BCS     110$                    ;BR IF CARRY SET (GOOD RETURN)
3999 052520  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4000 052522                                NEXT.ERRNO
4001 052522                        102$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     052522  104455                                                        TRAP    C$ERDF
     052524  001446                                                        .WORD   806
     052526  060662                                                        .WORD   T198SSR
     052530  012046                                                        .WORD   PKTSSR
4002 052532  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4003 052536                        110$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     052536  104406                                                        TRAP    C$CLP1
4004                               ;       Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
4005 052540  004737  062050                JSR     PC,T19RSFIF             ;SETUP PKT FOR WRITE MISC Reset Tape Status F-FLOPS
4006 052544  012704  062530                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4007 052550  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4008 052554  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4009 052560                                FORCERROR       122$                    ;@@FORCE ERROR IF FORCER=1
4010 052574  103407                        BCS     130$                    ;BR IF CARRY SET (GOOD RETURN)
4011 052576  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4012 052600                                NEXT.ERRNO
4013 052600                        122$:   ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     052600  104455                                                        TRAP    C$ERDF
     052602  001447                                                        .WORD   807
     052604  060330                                                        .WORD   T192SSR
     052606  012046                                                        .WORD   PKTSSR
4014 052610  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4015 052614                        130$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     052614  104406                                                        TRAP    C$CLP1
4016                               ;       Do a Write Subsystem READ STATUS
4017                               ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
4018                               ;         signals NOT=0 Then Print Error
4019 052616  004737  062030                JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
4020 052622  012704  062530                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4021 052626  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4022 052632  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4023 052636                                FORCERROR       132$                    ;@@FORCE ERROR IF FORCER=1
4024 052652  103407                        BCS     140$                    ;BR IF CARRY SET (GOOD RETURN)
4025 052654  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4026 052656                                NEXT.ERRNO
4027 052656                        132$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     052656  104455                                                        TRAP    C$ERDF
     052660  001450                                                        .WORD   808
     052662  060374                                                        .WORD   T193SSR
     052664  012046                                                        .WORD   PKTSSR
4028 052666  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4029 052672                        140$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     052672  104406                                                        TRAP    C$CLP1
4030 052674  004737  062270                JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
4031 052700  012701  060132                MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4032 052704  012702  062422                MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
4033 052710  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
4034 052712  042711  002000                BIC     #S1.ICER,(R1)           ;SET EXPD ICER  =0
4035 052716  042711  001000                BIC     #S1.IFMK,(R1)           ;SET EXPD IFMK  =0
4036 052722  042711  000400                BIC     #S1.IHER,(R1)           ;SET EXPD IHER  =0
4037 052726  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
4038 052734  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4039 052736  012701  062402                MOV     #T198FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4040 052742  012702  060112                MOV     #T19EXP,R2              ;EXPD ADDRESS
4041 052746  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
4042 052752  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4043 052756                                FORCERROR       152$,NOTSSR     ;@@D
4044 052766  103404                        BCS     160$                    ;BR IF YES
4045 052770                                NEXT.ERRNO
4046 052770                        152$:   ERRHRD  ERRNO,T197CMP,MSGLOOP   ;REPORT ERROR
     052770  104456                                                                TRAP    C$ERHRD
     052772  001451                                                                .WORD   809
     052774  061333                                                                .WORD   T197CMP
     052776  013064                                                                .WORD   MSGLOOP
4047 053000                        160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053000  104406                                                                TRAP    C$CLP1
4048                               ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4049 053002  005037  060044                CLR     T19PREV                 ;INIT 1-0 TRANSITION FLAG
4050 053006  012703  002752                MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
4051 053012  012300                200$:   MOV     (R3)+,R0                ;GET A TEST PATTERN
4052 053014  010337  002316                MOV     R3,TSTPTR               ;SAVE POINTER INTO TSTBLK
4053 053020  042700  000200                BIC     #WC.IFAD,R0             ;IFAD MUST  ALWAYS =0
4054 053024  042700  000100                BIC     #WC.IOTAD,R0            ;ITADO MUST ALWAYS =0
4055 053030  042700  000040                BIC     #WC.I1TAD,R0            ;ITAD1 MUST ALWAYS =0
4056 053034  010037  002312                MOV     R0,DATA                 ;SET DATA PATTERN
4057                               ;       Do Write Subsystem Write Control to Drive loopback signals group 1.
4058                               ;@@D    CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
4059 053040  013700  002312                MOV     DATA,R0                 ;GET TEST PATTERN
4060 053044  004737  062314                JSR     PC,T19CNVT              ;CONVERT PATTERN TO CONTROL DRIVE MASK
4061                                                                       ;R0 CONTAINS WRITE CONTROL DATA HERE
4062 053050  004737  062172                JSR     PC,T19WCTL              ;SETUP PACKET FOR WRITE CONTROL
4063 053054  012704  062530                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4064 053060  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4065 053064  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4066 053070                                FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
4067 053104  103407                        BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
4068 053106  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4069 053110                                NEXT.ERRNO
4070 053110                        212$:   ERRDF   ERRNO,T197SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053110  104455                                                                TRAP    C$ERDF
     053112  001452                                                                .WORD   810
     053114  060613                                                                .WORD   T197SSR
     053116  012046                                                                .WORD   PKTSSR
4071 053120  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4072 053124                        220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053124  104406                                                                TRAP    C$CLP1
4073
4074                               ;       Do Write Subsystem Write Format to Drive loopback signals group 2.
4075                               ;@@D    CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
4076 053126  013700  002312                MOV     DATA,R0                 ;GET TEST PATTERN
4077 053132  004737  062314                JSR     PC,T19CNVT              ;CONVERT PATTERN TO FORMAT DRIVE MASK
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
4078 053136  000300                    SWAB    R0                      ;WRITE FORMAT DATA RETURNED IN HIGH BYTE
4079 053140  004737  062212            JSR     PC,T19WFMT              ;SETUP PACKET FOR WRITE FORMAT
4080 053144  012704  062530            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4081 053150  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4082 053154  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4083 053160                            FORCERROR       232$                    ;@@DFORCE ERROR IF FORCER=1
4084 053174  103407                    BCS     240$                    ;BR IF CARRY SET (GOOD RETURN)
4085 053176  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4086 053200                            NEXT.ERRNO
4087 053200                    232$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053200  104455                                                            TRAP    C$ERDF
     053202  001453                                                            .WORD   811
     053204  060662                                                            .WORD   T198SSR
     053206  012046                                                            .WORD   PKTSSR
4088 053210  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4089 053214                    240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053214  104406                                                            TRAP    C$CLP1
4090                           ;       Do a Write Subsystem READ STATUS
4091 053216  004737  062030            JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
4092 053222  012704  062530            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4093 053226  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4094 053232  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4095 053236                            FORCERROR       252$                    ;@@DFORCE ERROR IF FORCER=1
4096 053252  103407                    BCS     260$                    ;BR IF CARRY SET (GOOD RETURN)
4097 053254  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4098 053256                            NEXT.ERRNO
4099 053256                    252$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053256  104455                                                            TRAP    C$ERDF
     053260  001454                                                            .WORD   812
     053262  060374                                                            .WORD   T193SSR
     053264  012046                                                            .WORD   PKTSSR
4100 053266  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4101 053272                    260$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053272  104406                                                            TRAP    C$CLP1
4102                           ;       If loopback data NOT= data sent Then Print Error
4103 053274  004737  062270            JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4104 053300  012701  060132            MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4105 053304  012702  062422            MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
4106 053310  013711  002312            MOV     DATA,(R1)               ;SET EXPD WORD #8 TO TEST DATA FIRST
4107 053314  013700  060044            MOV     T19PREV,R0              ;GET PREVIOUS DATA PATTERN
4108 053320  013703  002312            MOV     DATA,R3                 ;GET CURRENT PATTERN
4109 053324  012704  000400            MOV     #S1.IHER,R4             ;SETUP IHER EXPECTED
4110 053330  040411                    BIC     R4,(R1)                 ;SET EXPD IHER  =0
4111 053332  030400                    BIT     R4,R0                   ;PREVIOUS =1?
4112 053334  001403                    BEQ     275$                    ;BR IF NO
4113 053336  030403                    BIT     R4,R3                   ;CURRENT  =0?
4114 053340  001001                    BNE     275$                    ;BR IF NO
4115 053342  050411                    BIS     R4,(R1)                 ;SET EXPD IHER  =1
4116 053344  012704  001000    275$:   MOV     #S1.IFMK,R4             ;SETUP IFMK EXPECTED
4117 053350  040411                    BIC     R4,(R1)                 ;SET EXPD IFMK  =0
4118 053352  030400                    BIT     R4,R0                   ;PREVIOUS =1?
4119 053354  001403                    BEQ     280$                    ;BR IF NO
4120 053356  030403                    BIT     R4,R3                   ;CURRENT  =0?
4121 053360  001001                    BNE     280$                    ;BR IF NO
4122 053362  050411                    BIS     R4,(R1)                 ;SET EXPD IFMK  =1
4123 053364  012704  002000    280$:   MOV     #S1.ICER,R4             ;SETUP ICER EXPECTED
4124 053370  040411                    BIC     R4,(R1)                 ;SET EXPD ICER  =0
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
4125 053372  030400                        BIT     R4,R0                   ;PREVIOUS =1?
4126 053374  001403                        BEQ     285$                    ;BR IF NO
4127 053376  030403                        BIT     R4,R3                   ;CURRENT  =0?
4128 053400  001001                        BNE     285$                    ;BR IF NO
4129 053402  050411                        BIS     R4,(R1)                 ;SET EXPD ICER  =1
4130 053404  011100                285$:   MOV     (R1),R0                 ;GET EXPD WORD
4131                                       If previous IIDENT=1 and current is IIDENT=1 then EXPD= 0 else 1
4132 053406  012704  004000                MOV     #S1.IIDENT,R4           ;IIDENT
4133 053412  050400                        BIS     R4,R0                   ;ASSUME EXPD=1
4134 053414  030437  060044                BIT     R4,T19PREV              ;PREVIOUS IIDENT=1?
4135 053420  001403                        BEQ     288$                    ;BR IF NO
4136 053422  030403                        BIT     R4,R3                   ;IS CURRENT IIDENT=1?
4137 053424  001401                        BEQ     288$                    ;BR IF NO
4138 053426  040400                        BIC     R4,R0                   ;SET EXPD=0
4139 053430  052700  040000          288$: BIS     #S1.I2RES,R0            ;IRESV2 EXPD ALWAYS=1
4140 053434  052700  020000                BIS     #S1.I1RES,R0            ;IRESV1 EXPD ALWAYS=1
4141 053440  042700  100000                BIC     #S1.PARERR,R0           ;IGNORE PARERR
4142 053444  032712  100000                BIT     #S1.PARERR,(R2)         ;IS PARERR SET IN RECV?
4143 053450  001402                        BEQ     290$                    ;BR IF NO
4144 053452  052700  100000                BIS     #S1.PARERR,R0           ;SET IN EXPD
4145 053456  010011                290$:   MOV     R0,(R1)                 ;SETUP FINAL EXPD IN WORD #8
4146 053460  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
4147 053466  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4148 053470  012701  062402                MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4149 053474  012702  060112                MOV     #T19EXP,R2              ;EXPD ADDRESS
4150 053500  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
4151 053504  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4152 053510                                FORCERROR       302$,NOTSSR     ;@@D
4153 053520  103404                        BCS     310$                    ;BR IF YES
4154 053522                                NEXT.ERRNO
4155 053522                          302$: ERRHRD  ERRNO,T198CMP,MSGLOOP   ;REPORT ERROR
     053522  104456                                                        TRAP    C$ERHRD
     053524  001455                                                        .WORD   813
     053526  061421                                                        .WORD   T198CMP
     053530  013064                                                        .WORD   MSGLOOP
4156 053532                          310$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053532  104406                                                        TRAP    C$CLP1
4157                                ;       Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
4158 053534  004737  062050                JSR     PC,T19RSFIF             ;SETUP PKT FOR WRITE MISC Reset STATUS
4159 053540  012704  062530                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4160 053544  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4161 053550  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4162 053554                                FORCERROR       322$            ;@@DFORCE ERROR IF FORCER=1
4163 053570  103407                        BCS     330$                    ;BR IF CARRY SET (GOOD RETURN)
4164 053572  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4165 053574                                NEXT.ERRNO
4166 053574                          322$: ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053574  104455                                                        TRAP    C$ERDF
     053576  001456                                                        .WORD   814
     053600  060330                                                        .WORD   T192SSR
     053602  012046                                                        .WORD   PKTSSR
4167 053604  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4168 053610                          330$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053610  104406                                                        TRAP    C$CLP1
4169                                ;       Do a Write Subsystem READ STATUS
4170 053612  004737  062030                JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
4171 053616  012704  062530                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
4172 053622  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4173 053626  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4174 053632                                    FORCERROR       342$            ;@@DFORCE ERROR IF FORCER=1
4175 053646  103407                           BCS     350$                    ;BR IF CARRY SET (GOOD RETURN)
4176 053650  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4177 053652                                    NEXT.ERRNO
4178 053652                            342$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053652  104455                                                                    TRAP    C$ERDF
     053654  001457                                                                    .WORD   815
     053656  060374                                                                    .WORD   T193SSR
     053660  012046                                                                    .WORD   PKTSSR
4179 053662  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4180 053666                            350$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053666  104406                                                                    TRAP    C$CLP1
4181 053670  004737  062270                    JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4182 053674  012701  060132                    MOV     @T19EXSTA,R1            ;GET EXPECTED READ STATUS
4183 053700  012702  062422                    MOV     @T198FSTA,R2            ;GET RECV READ STATUS
4184 053704  011211                           MOV     (R2),(R1)               ;SET EXPD WORD @8 = RECV TEMP
4185 053706  042711  002000                    BIC     @S1.ICER,(R1)           ;SET EXPD ICER  =0
4186 053712  042711  001000                    BIC     @S1.IFMK,(R1)           ;SET EXPD IFMK  =0
4187 053716  042711  000400                    BIC     @S1.IHER,(R1)           ;SET EXPD IHER  =0
4188 053722  016261  000002  000002            MOV     2(R2),2(R1)             ;SET EXPD WORD @9 = RECV (NOT TESTING)
4189 053730  005000                           CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4190 053732  012701  062402                    MOV     @T198FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4191 053736  012702  060112                    MOV     @T19EXP,R2              ;EXPD ADDRESS
4192 053742  012703  000024                    MOV     @20.,R3                 ;NUMBER OF BYTES TO COMPARE
4193 053746  004737  011500                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4194 053752                                    FORCERROR       362$,NOTSSR     ;@@D
4195 053762  103404                           BCS     370$                    ;BR IF YES
4196 053764                                    NEXT.ERRNO
4197 053764                            362$:   ERRHRD  ERRNO,T197CMP,MSGSTAT   ;REPORT ERROR
     053764  104456                                                                    TRAP    C$ERHRD
     053766  001460                                                                    .WORD   816
     053770  061333                                                                    .WORD   T197CMP
     053772  012350                                                                    .WORD   MSGSTAT
4198 053774                            370$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053774  104406                                                                    TRAP    C$CLP1
4199
4200 053776  013737  002312  060044    ,       MOV     DATA,T19PREV            ;SETUP PREVIOUS DATA FOR EXPD CALC.
4201 054004  013703  002316                    MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
4202 054010  020327  003062                    CMP     R3,@TBLEND              ;END OF TSTBLK?
4203 054014  103002                           BHIS    400$                    ;BR IF YES
4204 054016  000137  053012                    JMP     200$                    ;DO NEXT TSTBLK PATTERN
4205 054022                            400$:
4206
4207 054022                                    ENDSUB                          ;////////////// END SUBTEST //////////
     054022                                                                            L10067:
     054022  104403                                                                    TRAP    C$ESUB
4208
4209 054024  005737  002222                    TST     FATFLG                  ;ANY FATAL ERRORS ?
4210 054030  001402                           BEQ     460$                    ;BRANCH IF NOT
4211 054032  004737  017202                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4212 054036                            460$:
4213
4214
4215
4216
```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```
4217
4218                                        .SBTTL  TEST  8: SUBTEST 2: LOOPBACK READ/WRITE SIGNALS TEST
4219
4220                                ;++
4221                                ; TEST 8: SUBTEST 2:
4222                                ;
4223                                ; SUBTEST DESCRIPTION:
4224                                ;
4225                                ;       This subtest verifies the Read/Write data loopback path.
4226                                ;       The Read/Write data signals are IR<7:0> and IW<7:0>
4227                                ;       respectively.
4228                                ;
4229                                ; TEST STEPS:
4230                                ;
4231                                ;
4232                                ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4233                                ;    BEGIN
4234                                ;       Write to TSSR register to soft initialize the controller
4235                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
4236                                ;       If Extended Features Hardware Switch Clear then:
4237                                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
4238                                ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
4239                                ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
4240                                ;       Do a WRITE NPR to set loopback and tape direction OUT
4241                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4242                                ;       Do a READ FIFO with tape direction OUT to load tape out write latch
4243                                ;       Do a WRITE NPR to set loopback and tape direction IN
4244                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
4245                                ;         to strobe loopback data into FIFO.
4246                                ;       Do a READ FIFO with tape direction IN to read data
4247                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
4248                                ;       Do a Write Subsystem READ STATUS
4249                                ;       If Input Ready  NOT=1    Then Print Error
4250                                ;       If Output Ready NOT=0    Then Print Error
4251                                ;       If Data In Miss NOT=0    Then Print Error
4252                                ;    END
4253                                ;--
4254 054036                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////
      054036                                                                              T8.2:
      054036  104402                                                                             TRAP    C$BSUB
4255                                ;       Write to TSSR register to soft initialize the controller
4256 054040                         5$:
4257 054040  004737  015774                JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
4258 054044  103405                        BCS     10$                     ;BR IF SOFT INIT OKAY
4259 054046  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4260 054050                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      054050  104455                                                                             TRAP    C$ERDF
      054052  001460                                                                             .WORD   816
      054054  003652                                                                             .WORD   SFIERR
      054056  012034                                                                             .WORD   SFIMSG
4261                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
4262 054060  005037  002222        10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
4263 054064  012704  062360                MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
4264 054070  004737  010662                JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
4265 054074                                FORCERROR       12$                     ;@@@FORCE ERROR IF FORCER=1
4266 054110  103407                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
4267 054112  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

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

```
                                        NEXT.ERRNO
                        12$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
054114  104455                                                          TRAP    C$ERDF
054116  001461                                                          .WORD   817
054120  060273                                                          .WORD   T19SSR
054122  012046                                                          .WORD   PKTSSR
                        ;       INC     FATFLG                  ;SET FATAL ERROR FLAG
                        15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
054130  104406                                                          TRAP    C$CLP1
                        ;       If Extended Features Hardware Switch Clear then:
                        ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
                                MOV     @T19BFR,R1              ;MESSAGE BUFFER ADDRESS
                                BIT     @X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
                                BNE     30$                     ;BR IF YES
                                JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
                                MOV     @T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
                                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
                                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
                                FORCERROR       22$                     ;@@DFORCE ERROR IF FORCER=1
                                BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                NEXT.ERRNO
                        22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
054206  104455                                                          TRAP    C$ERDF
054210  001462                                                          .WORD   818
054212  060330                                                          .WORD   T192SSR
054214  012046                                                          .WORD   PKTSSR
                                INC     FATFLG                  ;SET FATAL ERROR FLAG
                        30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
054222  104406                                                          TRAP    C$CLP1
                        ;       Do WRITE CHARACTERISTICS to select reserved unit 7
                                MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
                                JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
                                FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
                                BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                NEXT.ERRNO
                        42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
054254  104455                                                          TRAP    C$ERDF
054256  001463                                                          .WORD   819
054260  060273                                                          .WORD   T19SSR
054262  012046                                                          .WORD   PKTSSR
                                INC     FATFLG                  ;SET FATAL ERROR FLAG
                        50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
054270  104406                                                          TRAP    C$CLP1


                        ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
                                MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
                        100$:   MOV     (R3)+,DATA              ;GET A TEST PATTERN
                                BIC     #^C<377>,DATA           ;DATA IS BYTE
                                MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
                        ;       Do a WRITE NPR to set loopback and tape direction OUT
                                MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
                                BIS     #NP.LOOP,R0             ;SET LOOPBACK
                                JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
                                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
```

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

```
4310 054334  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4311 054340  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4312 054344                                  FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
4313 054360  103407                          BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
4314 054362  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4315 054364                                  NEXT.ERRNO
4316 054364                          102$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054364  104455                                                                  TRAP    C$ERDF
     054366  001464                                                                  .WORD   820
     054370  060441                                                                  .WORD   T194SSR
     054372  012046                                                                  .WORD   PKTSSR
4317 054374  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
4318 054400                          105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054400  104406                                                                  TRAP    C$CLP1
4319                                 ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4320 054402  012700  000001                  MOV     #1,R0                   ;WRITE 1 BYTE
4321 054406  012701  002312                  MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
4322 054412  004737  062136                  JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
4323 054416  012704  062530                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4324 054422  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4325 054426  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4326 054432                                  FORCERROR       107$                    ;@@DFORCE ERROR IF FORCER=1
4327 054446  103407                          BCS     110$                    ;BR IF CARRY SET (GOOD RETURN)
4328 054450  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4329 054452                                  NEXT.ERRNO
4330 054452                          107$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054452  104455                                                                  TRAP    C$ERDF
     054454  001465                                                                  .WORD   821
     054456  060504                                                                  .WORD   T195SSR
     054460  012046                                                                  .WORD   PKTSSR
4331 054462  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
4332 054466                          110$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054466  104406                                                                  TRAP    C$CLP1
4333                                 ;       Do a READ FIFO with tape direction OUT to load tape out write latch
4334 054470  012700  000001                  MOV     #1,R0                   ;SET READ BYTE COUNT
4335 054474  004737  062116                  JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
4336 054500  012704  062530                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4337 054504  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4338 054510  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4339 054514                                  FORCERROR       122$                    ;@@DFORCE ERROR IF FORCER=1
4340 054530  103407                          BCS     130$                    ;BR IF CARRY SET (GOOD RETURN)
4341 054532  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4342 054534                                  NEXT.ERRNO
4343 054534                          122$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054534  104455                                                                  TRAP    C$ERDF
     054536  001466                                                                  .WORD   822
     054540  060550                                                                  .WORD   T196SSR
     054542  012046                                                                  .WORD   PKTSSR
4344 054544  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
4345 054550                          130$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054550  104406                                                                  TRAP    C$CLP1
4346                                 ;       Do a WRITE NPR to set loopback and tape direction IN
4347 054552  005000                          CLR     R0                      ;CLR NP.OUT TO SET TAPE DIRECTION IN
4348 054554  052700  000040                  BIS     #NP.LOOP,R0             ;SET LOOPBACK
4349 054560  004737  062072                  JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
4350 054564  012704  062530                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4351 054570  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
```

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

```
4352 054574  004737  016356                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4353 054600                                    FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
4354 054614  103407                            BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
4355 054616  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4356 054620                                    NEXT.ERRNO
4357 054620                            142$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054620  104455                                                                    TRAP    C$ERDF
     054622  001467                                                                    .WORD   823
     054624  060441                                                                    .WORD   T194SSR
     054626  012046                                                                    .WORD   PKTSSR
4358 054630  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4359 054634                            150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054634  104406                                                                    TRAP    C$CLP1
4360                                   ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
4361 054636  012700  000001                    MOV     #1,R0                   ;WRITE 1 BYTE
4362 054642  012701  002312                    MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
4363 054646  004737  062136                    JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
4364 054652  012704  062530                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4365 054656  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4366 054662  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4367 054666                                    FORCERROR       162$                    ;@@DFORCE ERROR IF FORCER=1
4368 054702  103407                            BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
4369 054704  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4370 054706                                    NEXT.ERRNO
4371 054706                            162$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054706  104455                                                                    TRAP    C$ERDF
     054710  001470                                                                    .WORD   824
     054712  060504                                                                    .WORD   T195SSR
     054714  012046                                                                    .WORD   PKTSSR
4372 054716  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4373 054722                            170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054722  104406                                                                    TRAP    C$CLP1
4374                                   ;       Do a READ FIFO with tape direction IN to read data
4375                                   ;       If Data read from FIFO NOT= to Data sent Then Print Error
4376 054724  012700  000001                    MOV     #1,R0                   ;SET READ BYTE COUNT
4377 054730  004737  062116                    JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
4378 054734  012704  062530                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4379 054740  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4380 054744  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4381 054750                                    FORCERROR       182$                    ;@@DFORCE ERROR IF FORCER=1
4382 054764  103407                            BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
4383 054766  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4384 054770                                    NEXT.ERRNO
4385 054770                            182$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054770  104455                                                                    TRAP    C$ERDF
     054772  001471                                                                    .WORD   825
     054774  060550                                                                    .WORD   T196SSR
     054776  012046                                                                    .WORD   PKTSSR
4386 055000  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4387 055004                            190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055004  104406                                                                    TRAP    C$CLP1
4388 055006  004737  062270                    JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4389 055012  012701  060132                    MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4390 055016  012702  062422                    MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
4391 055022  013711  002312                    MOV     DATA,(R1)               ;SET EXPD WORD #8 = DATA
4392 055026  016261  000002  000002            MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
4393 055034  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
```

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

```
4394 055036  012701  062402                MOV     #T198FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4395 055042  012702  060112                MOV     #T19EXP,R2              ;EXPD ADDRESS
4396 055046  012703  000022                MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
4397 055052  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4398 055056                                FORCERROR       202$,NOTSSR     ;@@D
4399 055066  103404                        BCS     210$                    ;BR IF YES
4400 055070                                NEXT.ERRNO
4401 055070                        202$:   ERRHRD  ERRNO,T199CMP,MSGSUB    ;REPORT ERROR
     055070  104456                                                            TRAP    C$ERHRD
     055072  001472                                                            .WORD   826
     055074  061510                                                            .WORD   T199CMP
     055076  013742                                                            .WORD   MSGSUB
4402 055100                        210$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055100  104406                                                            TRAP    C$CLP1
4403                               ;       Do a Write Subsystem READ STATUS
4404                               ;       If Input Ready  NOT=1   Then Print Error
4405                               ;       If Output Ready NOT=0   Then Print Error
4406                               ;       If Data In Miss NOT=0   Then Print Error
4407 055102  004737  062030                JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
4408 055106  012704  062530                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4409 055112  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4410 055116  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4411 055122                                FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
4412 055136  103407                        BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
4413 055140  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4414 055142                                NEXT.ERRNO
4415 055142                        212$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     055142  104455                                                            TRAP    C$ERDF
     055144  001473                                                            .WORD   827
     055146  060374                                                            .WORD   T193SSR
     055150  012046                                                            .WORD   PKTSSR
4416 055152  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4417 055156                        220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055156  104406                                                            TRAP    C$CLP1
4418 055160  004737  062270                JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4419 055164  012701  060132                MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4420 055170  012702  062422                MOV     #T198FSTA,R2            ;GET RECV READ STATUS
4421 055174  012221                        MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
4422 055176  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
4423 055200  052711  000020                BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
4424 055204  042711  000040                BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
4425 055210  042711  000200                BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
4426 055214  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4427 055216  012701  062402                MOV     #T198FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4428 055222  012702  060112                MOV     #T19EXP,R2              ;EXPD ADDRESS
4429 055226  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
4430 055232  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4431 055236                                FORCERROR       232$,NOTSSR     ;@@D
4432 055246  103404                        BCS     240$                    ;BR IF YES
4433 055250                                NEXT.ERRNO
4434 055250                        232$:   ERRHRD  ERRNO,T196CMP,MSGSTAT   ;REPORT ERROR
     055250  104456                                                            TRAP    C$ERHRD
     055252  001474                                                            .WORD   828
     055254  061250                                                            .WORD   T196CMP
     055256  012350                                                            .WORD   MSGSTAT
4435 055260                        240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055260  104406                                                            TRAP    C$CLP1
```

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

```
4436
4437
4438
4439 055262                                    FORCEXIT        255$            ;@@D
4440 055272  013703  002316                    MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
4441 055276  020327  003062                    CMP     R3,#TBLEND              ;END OF TSTBLK?
4442 055302  103002                           BHIS    255$                    ;BR IF YES
4443 055304  000137  054276                    JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
4444 055310                                255$:
4445
4446 055310                                    ENDSUB                          ;////////////// END SUBTEST //////////
     055310                                                                                    L10070:
     055310  104403                                                                                TRAP    C$ESUB
4447
4448 055312  005737  002222                    TST     FATFLG                  ;ANY FATAL ERRORS ?
4449 055316  001402                           BEQ     260$                    ;BRANCH IF NOT
4450 055320  004737  017202                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4451 055324                                260$:
4452
4453                                           .SBTTL  TEST  8: SUBTEST 3: LOOPBACK WRITE STROBE TEST
4454
4455
4456                                       ;++
4457                                       ; TEST 8: SUBTEST 3:
4458                                       ;
4459                                       ; SUBTEST DESCRIPTION:
4460                                       ;
4461                                       ;       This subtest verifies the Write Strobe loopback path
4462                                       ;       can strobe data from the FIFO to the Data lines.
4463                                       ;       The signal IRESV3 drives IWSTR (write strobe) to write
4464                                       ;       data from the FIFO to the tape data out latch.
4465                                       ;
4466                                       ; TEST STEPS:
4467                                       ;
4468                                       ;
4469                                       ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4470                                       ;    BEGIN
4471                                       ;       Write to TSSR register to soft initialize the cor:roller
4472                                       ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
4473                                       ;       If Extended Features Hardware Switch Clear then:
4474                                       ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
4475                                       ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
4476                                       ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
4477                                       ;       Do a WRITE NPR to set loopback and tape direction OUT
4478                                       ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
4479                                       ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4480                                       ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 0 to load write data latch
4481                                       ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
4482                                       ;       Do a WRITE NPR to set loopback and tape direction IN
4483                                       ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
4484                                       ;         to strobe loopback data into FIFO.
4485                                       ;       Do a READ FIFO with tape direction IN to read data
4486                                       ;       If Data read from FIFO NOT= to Data sent Then Print Error
4487                                       ;    END
4488                                       ;--
4489 055324                                    BGNSUB                          ;////////////// BEGIN SUBTEST //////////
     055324                                                                                    T8.3:
```

```
     055324  104402                                                                  TRAP    C$BSUB
4490                          ;       Write to TSSR register to soft initialize the controller
4491 055326                   5$:
4492 055326  004737  015774           JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
4493 055332  103405                   BCS     10$                     ;BR IF SOFT INIT OKAY
4494 055334  010001                   MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4495 055336                           ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     055336  104455                                                                  TRAP    C$ERDF
     055340  001474                                                                  .WORD   828
     055342  003652                                                                  .WORD   SFIERR
     055344  012034                                                                  .WORD   SFIMSG
4496                          ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
4497 055346  005037  002222   10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
4498 055352  012704  062360           MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
4499 055356  004737  010662           JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
4500 055362                           FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
4501 055376  103407                   BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
4502 055400  010001                   MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4503 055402                           NEXT.ERRNO
4504 055402                   12$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     055402  104455                                                                  TRAP    C$ERDF
     055404  001475                                                                  .WORD   829
     055406  060273                                                                  .WORD   T19SSR
     055410  012046                                                                  .WORD   PKTSSR
4505 055412  005237  002222           INC     FATFLG                  ;SET FATAL ERROR FLAG
4506 055416                   15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055416  104406                                                                  TRAP    C$CLP1
4507                          ;       If Extended Features Hardware Switch Clear then:
4508                          ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
4509 055420  012701  062402           MOV     #T19BFR,R1              ;MESSAGE BUFFER ADDRESS
4510 055424  032761  000200  000012   BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
4511 055432  001026                   BNE     30$                     ;BR IF YES
4512 055434  004737  062232           JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
4513 055440  012704  062530           MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4514 055444  010465  000000           MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4515 055450  004737  016336           JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4516 055454                           FORCERROR       22$                     ;@@DFORCE ERROR IF FORCER=1
4517 055470  103407                   BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
4518 055472  010001                   MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4519 055474                           NEXT.ERRNO
4520 055474                   22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     055474  104455                                                                  TRAP    C$ERDF
     055476  001476                                                                  .WORD   830
     055500  060330                                                                  .WORD   T192SSR
     055502  012046                                                                  .WORD   PKTSSR
4521 055504  005237  002222           INC     FATFLG                  ;SET FATAL ERROR FLAG
4522 055510                   30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055510  104406                                                                  TRAP    C$CLP1
4523                          ;       Do WRITE CHARACTERISTICS to select reserved unit 7
4524 055512  012704  062360           MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
4525 055516  004737  010662           JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
4526 055522                           FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
4527 055536  103407                   BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
4528 055540  010001                   MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4529 055542                           NEXT.ERRNO
4530 055542                   42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     055542  104455                                                                  TRAP    C$ERDF
```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```
                055544  001477                                                          .WORD   831
                055546  060273                                                          .WORD   T19SSR
                055550  012046                                                          .WORD   PKTSSR
4531            055552  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
4532            055556                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                055556  104406                                                          TRAP    C$CLP1
4533
4534                                            ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4535            055560  012703  002752                  MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
4536            055564  012337  002312          100$:   MOV     (R3)+,DATA              ;GET A TEST PATTERN
4537            055570  042737  177400  002312          BIC     #^C<377>,DATA           ;DATA IS BYTE
4538            055576  010337  002316                  MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
4539                                            ;       Do a WRITE NPR to set loopback and tape direction OUT
4540            055602  012700  000100                  MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
4541            055606  052700  000040                  BIS     #NP.LOOP,R0             ;SET LOOPBACK
4542            055612  004737  062072                  JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
4543            055616  012704  062530                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4544            055622  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4545            055626  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4546            055632                                  FORCERROR       102$            ;@@DFORCE ERROR IF FORCER=1
4547            055646  103407                          BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
4548            055650  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4549            055652                                  NEXT.ERRNO
4550            055652                          102$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                055652  104455                                                          TRAP    C$ERDF
                055654  001500                                                          .WORD   832
                055656  060441                                                          .WORD   T194SSR
                055660  012046                                                          .WORD   PKTSSR
4551            055662  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
4552            055666                          105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                055666  104406                                                          TRAP    C$CLP1
4553                                            ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
4554            055670  012700  000002                  MOV     #WF.I3RES,R0            ;IRESV3==>IWSTR=1
4555            055674  004737  062212                  JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
4556            055700  012704  062530                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4557            055704  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4558            055710  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4559            055714                                  FORCERROR       112$            ;@@DFORCE ERROR IF FORCER=1
4560            055730  103407                          BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
4561            055732  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4562            055734                                  NEXT.ERRNO
4563            055734                          112$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                055734  104455                                                          TRAP    C$ERDF
                055736  001501                                                          .WORD   833
                055740  060662                                                          .WORD   T198SSR
                055742  012046                                                          .WORD   PKTSSR
4564            055744  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
4565            055750                          120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                055750  104406                                                          TRAP    C$CLP1
4566                                            ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4567            055752  012700  000001                  MOV     #1,R0                   ;WRITE 1 BYTE
4568            055756  012701  002312                  MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
4569            055762  004737  062136                  JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
4570            055766  012704  062530                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4571            055772  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4572            055776  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4573            056002                                  FORCERROR       132$            ;@@DFORCE ERROR IF FORCER=1
```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```
4574 056016  103407                    BCS     140$                    ;BR IF CARRY SET (GOOD RETURN)
4575 056020  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4576 056022                            NEXT.ERRNO
4577 056022                    132$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     056022  104455                                                            TRAP    C$ERDF
     056024  001502                                                            .WORD   834
     056026  060504                                                            .WORD   T195SSR
     056030  012046                                                            .WORD   PKTSSR
4578 056032  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4579 056036                    140$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056036  104406                                                            TRAP    C$CLP1
4580                           ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 0
4581 056040  005000                    CLR     R0                      ;SET IRESV3==>IWSTR=0
4582 056042  004737  062212            JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
4583 056046  012704  062530            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4584 056052  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4585 056056  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4586 056062                            FORCERROR       152$                    ;@@DFORCE ERROR IF FORCER=1
4587 056076  103407                    BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
4588 056100  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4589 056102                            NEXT.ERRNO
4590 056102                    152$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     056102  104455                                                            TRAP    C$ERDF
     056104  001503                                                            .WORD   835
     056106  060662                                                            .WORD   T198SSR
     056110  012046                                                            .WORD   PKTSSR
4591 056112  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4592 056116                    160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056116  104406                                                            TRAP    C$CLP1
4593                           ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
4594 056120  012700  000002            MOV     #WF.I3RES,R0            ;IRESV3==>IWSTR=1
4595 056124  004737  062212            JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
4596 056130  012704  062530            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4597 056134  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4598 056140  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4599 056144                            FORCERROR       172$                    ;@@DFORCE ERROR IF FORCER=1
4600 056160  103407                    BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
4601 056162  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4602 056164                            NEXT.ERRNO
4603 056164                    172$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     056164  104455                                                            TRAP    C$ERDF
     056166  001504                                                            .WORD   836
     056170  060662                                                            .WORD   T198SSR
     056172  012046                                                            .WORD   PKTSSR
4604 056174  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4605 056200                    180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056200  104406                                                            TRAP    C$CLP1
4606
4607                           ;       Do a WRITE NPR to set loopback and tape direction IN
4608 056202  005000                    CLR     R0                      ;CLR NP.OUT TO SET TAPE DIRECTION IN
4609 056204  052700  000040            BIS     #NP.LOOP,R0             ;SET LOOPBACK
4610 056210  004737  062072            JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
4611 056214  012704  062530            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4612 056220  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4613 056224  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4614 056230                            FORCERROR       182$                    ;@@DFORCE ERROR IF FORCER=1
4615 056244  103407                    BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```
4616 056246  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4617 056250                                 NEXT.ERRNO
4618 056250                         182$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     056250  104455                                                                TRAP    C$ERDF
     056252  001505                                                                .WORD   837
     056254  060441                                                                .WORD   T194SSR
     056256  012046                                                                .WORD   PKTSSR
4619 056260  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4620 056264                         190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056264  104406                                                                TRAP    C$CLP1
4621                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
4622 056266  012700  000001                MOV     #1,R0                   ;WRITE 1 BYTE
4623 056272  012701  002312                MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
4624 056276  004737  062136                JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
4625 056302  012704  062530                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4626 056306  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4627 056312  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4628 056316                                FORCERROR       202$                    ;@@DFORCE ERROR IF FORCER=1
4629 056332  103407                        BCS     210$                    ;BR IF CARRY SET (GOOD RETURN)
4630 056334  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4631 056336                                 NEXT.ERRNO
4632 056336                         202$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     056336  104455                                                                TRAP    C$ERDF
     056340  001506                                                                .WORD   838
     056342  060504                                                                .WORD   T195SSR
     056344  012046                                                                .WORD   PKTSSR
4633 056346  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4634 056352                         210$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056352  104406                                                                TRAP    C$CLP1
4635                                ;       Do a READ FIFO with tape direction IN to read data
4636 056354  012700  000001                MOV     #1,R0                   ;SET READ BYTE COUNT
4637 056360  004737  062116                JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
4638 056364  012704  062530                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4639 056370  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4640 056374  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4641 056400                                FORCERROR       222$                    ;@@DFORCE ERROR IF FORCER=1
4642 056414  103407                        BCS     230$                    ;BR IF CARRY SET (GOOD RETURN)
4643 056416  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4644 056420                                 NEXT.ERRNO
4645 056420                         222$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     056420  104455                                                                TRAP    C$ERDF
     056422  001507                                                                .WORD   839
     056424  060550                                                                .WORD   T196SSR
     056426  012046                                                                .WORD   PKTSSR
4646 056430  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4647 056434                         230$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056434  104406                                                                TRAP    C$CLP1
4648                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
4649 056436  004737  062270                JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4650 056442  012701  060132                MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4651 056446  012702  062422                MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
4652 056452  013711  002312                MOV     DATA,(R1)               ;SET EXPD WORD #8 = DATA
4653 056456  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
4654 056464  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4655 056466  012701  062402                MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4656 056472  012702  060112                MOV     #T19EXP,R2              ;EXPD ADDRESS
4657 056476  012703  000022                MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```
4658 056502  004737  011500                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4659 056506                                  FORCERROR       242$,NOTSSR     ;@@D
4660 056516  103404                          BCS     250$                    ;BR IF YES
4661 056520                                  NEXT.ERRNO
4662 056520                          242$:   ERRHRD  ERRNO,T19WSTR,MSGSUB    ;REPORT ERROR
     056520  104456                                                                  TRAP    C$ERHRD
     056522  001510                                                                  .WORD   840
     056524  061573                                                                  .WORD   T19WSTR
     056526  013742                                                                  .WORD   MSGSUB
4663 056530                          250$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056530  104406                                                                  TRAP    C$CLP1
4664
4665
4666 056532                                  FORCEXIT        255$            ;@@D
4667 056542  013703  002316                  MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
4668 056546  020327  003062                  CMP     R3,#TBLEND              ;END OF TSTBLK?
4669 056552  103002                          BHIS    255$                    ;BR IF YES
4670 056554  000137  055564                  JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
4671 056560                          255$:
4672
4673 056560                                  ENDSUB                          ;////////////// END SUBTEST //////////
     056560                                                                        L10071:
     056560  104403                                                                  TRAP    C$ESUB
4674
4675 056562  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
4676 056566  001402                          BEQ     260$                    ;BRANCH IF NOT
4677 056570  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4678 056574                          260$:
4679                                         .SBTTL  TEST  8: SUBTEST 4 LOOPBACK READ STROBE TEST
4680
4681                                 ;++
4682                                 ; TEST 8: SUBTEST 4:
4683                                 ;
4684                                 ; SUBTEST DESCRIPTION:
4685                                 ;
4686                                 ;       This subtest verifies the Read Strobe loopback path
4687                                 ;       can strobe the data from the Data lines to the FIFO.
4688                                 ;       The signal IRESV4 drives IRSTR (read strobe) to write
4689                                 ;       from the data lines to the FIFO.
4690                                 ;
4691                                 ; TEST STEPS:
4692                                 ;
4693                                 ;
4694                                 ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4695                                 ;    BEGIN
4696                                 ;       Write to TSSR register to soft initialize the controller
4697                                 ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
4698                                 ;       If Extended Features Hardware Switch Clear then:
4699                                 ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
4700                                 ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
4701                                 ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
4702                                 ;       Do a WRITE NPR to set loopback and tape direction OUT
4703                                 ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
4704                                 ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4705                                 ;       Do a READ FIFO with tape direction OUT to load tape out write latch
4706                                 ;       Do a WRITE NPR to set loopback and tape direction IN
4707                                 ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 to write loop data to FIFO
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
4708                                    ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
4709                                    ;         (to strobe loopback data into FIFO.)
4710                                    ;       Do a READ FIFO with tape direction IN to read data
4711                                    ;       If Data read from FIFO NOT= to Data sent Then Print Error
4712                                    ;   END
4713                                    ;--
4714 056574                                     BGNSUB                          ;////////////// BEGIN SUBTEST //////////
      056574                                                                                      T8.4:
      056574  104402                                                                              TRAP    C$BSUB
4715                                    ;       Write to TSSR register to soft initialize the controller
4716 056576                             5$:
4717 056576  004737  015774                    JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
4718 056602  103405                            BCS     10$                     ;BR IF SOFT INIT OKAY
4719 056604  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4720 056606                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      056606  104455                                                                              TRAP    C$ERDF
      056610  001510                                                                              .WORD   840
      056612  003652                                                                              .WORD   SFIERR
      056614  012034                                                                              .WORD   SFIMSG
4721                                    ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
4722 056616  005037  002222             10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
4723 056622  012704  062360                    MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
4724 056626  004737  010662                    JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
4725 056632                                    FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
4726 056646  103407                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
4727 056650  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4728 056652                                    NEXT.ERRNO
4729 056652                             12$:   ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      056652  104455                                                                              TRAP    C$ERDF
      056654  001511                                                                              .WORD   841
      056656  060273                                                                              .WORD   T19SSR
      056660  012046                                                                              .WORD   PKTSSR
4730 056662  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4731 056666                             15$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      056666  104406                                                                              TRAP    C$CLP1
4732                                    ;       If Extended Features Hardware Switch Clear then:
4733                                    ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
4734 056670  012701  062402                    MOV     #T19BFR,R1              ;MESSAGE BUFFER ADDRESS
4735 056674  032761  000200  000012            BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
4736 056702  001026                            BNE     30$                     ;BR IF YES
4737 056704  004737  062232                    JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
4738 056710  012704  062530                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4739 056714  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4740 056720  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4741 056724                                    FORCERROR       22$                     ;@@DFORCE ERROR IF FORCER=1
4742 056740  103407                            BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
4743 056742  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4744 056744                                    NEXT.ERRNO
4745 056744                             22$:   ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      056744  104455                                                                              TRAP    C$ERDF
      056746  001512                                                                              .WORD   842
      056750  060330                                                                              .WORD   T192SSR
      056752  012046                                                                              .WORD   PKTSSR
4746 056754  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4747 056760                             30$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      056760  104406                                                                              TRAP    C$CLP1
4748                                    ;       Do WRITE CHARACTERISTICS to select reserved unit 7
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
4749 056762  012704  062360                    MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
4750 056766  004737  010662                    JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
4751 056772                                    FORCERROR       42$             ;@@DFORCE ERROR IF FORCER=1
4752 057006  103407                           BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
4753 057010  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4754 057012                                    NEXT.ERRNO
4755 057012                            42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     057012  104455                                                                TRAP    C$ERDF
     057014  001513                                                                .WORD   843
     057016  060273                                                                .WORD   T19SSR
     057020  012046                                                                .WORD   PKTSSR
4756 057022  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4757 057026                            50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057026  104406                                                                TRAP    C$CLP1
4758
4759                                   ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4760 057030  012703  002752                    MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
4761 057034  012337  002312            100$:   MOV     (R3)+,DATA              ;GET A TEST PATTERN
4762 057040  042737  177400  002312            BIC     #^C<377>,DATA           ;DATA IS BYTE
4763 057046  010337  002316                    MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
4764                                   ;        Do a WRITE NPR to set loopback and tape direction OUT
4765 057052  012700  000100                    MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
4766 057056  052700  000040                    BIS     #NP.LOOP,R0             ;SET LOOPBACK
4767 057062  004737  062072                    JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
4768 057066  012704  062530                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4769 057072  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4770 057076  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4771 057102                                    FORCERROR       102$            ;@@DFORCE ERROR IF FORCER=1
4772 057116  103407                           BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
4773 057120  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4774 057122                                    NEXT.ERRNO
4775 057122                            102$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057122  104455                                                                TRAP    C$ERDF
     057124  001514                                                                .WORD   844
     057126  060441                                                                .WORD   T194SSR
     057130  012046                                                                .WORD   PKTSSR
4776 057132  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4777 057136                            105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057136  104406                                                                TRAP    C$CLP1
4778                                   ;        Do a WRITE FORMAT to set IRESV4-->IRSTR = 1
4779 057140  012700  000001                    MOV     #WF.I4RES,R0            ;IRESV4-->IRSTR=1
4780 057144  004737  062212                    JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
4781 057150  012704  062530                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4782 057154  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4783 057160  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4784 057164                                    FORCERROR       112$            ;@@DFORCE ERROR IF FORCER=1
4785 057200  103407                           BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
4786 057202  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4787 057204                                    NEXT.ERRNO
4788 057204                            112$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057204  104455                                                                TRAP    C$ERDF
     057206  001515                                                                .WORD   845
     057210  060662                                                                .WORD   T198SSR
     057212  012046                                                                .WORD   PKTSSR
4789 057214  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4790 057220                            120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057220  104406                                                                TRAP    C$CLP1
```

```
4791                                    ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4792 057222  012700  000001                    MOV     #1,R0                   ;WRITE 1 BYTE
4793 057226  012701  002312                    MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
4794 057232  004737  062136                    JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
4795 057236  012704  062530                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4796 057242  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4797 057246  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4798 057252                                    FORCERROR       132$                    ;@@DFORCE ERROR IF FORCER=1
4799 057266  103407                            BCS     140$                    ;BR IF CARRY SET (GOOD RETURN)
4800 057270  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4801 057272                                    NEXT.ERRNO
4802 057272                             132$:  ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057272  104455                                                                 TRAP    C$ERDF
     057274  001516                                                                 .WORD   846
     057276  060504                                                                 .WORD   T195SSR
     057300  012046                                                                 .WORD   PKTSSR
4803 057302  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4804 057306                             140$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057306  104406                                                                 TRAP    C$CLP1
4805                                    ;       Do a READ FIFO with tape direction OUT to load tape out write latch
4806 057310  012700  000001                    MOV     #1,R0                   ;SET READ BYTE COUNT
4807 057314  004737  062116                    JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
4808 057320  012704  062530                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4809 057324  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4810 057330  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4811 057334                                    FORCERROR       152$                    ;@@DFORCE ERROR IF FORCER=1
4812 057350  103407                            BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
4813 057352  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4814 057354                                    NEXT.ERRNO
4815 057354                             152$:  ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057354  104455                                                                 TRAP    C$ERDF
     057356  001517                                                                 .WORD   847
     057360  060550                                                                 .WORD   T196SSR
     057362  012046                                                                 .WORD   PKTSSR
4816 057364  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4817 057370                             160$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057370  104406                                                                 TRAP    C$CLP1
4818                                    ;       Do a WRITE NPR to set loopback and tape direction IN
4819 057372  005000                            CLR     R0                      ;CLR NP.OUT TO SET TAPE DIRECTION IN
4820 057374  052700  000040                    BIS     #NP.LOOP,R0             ;SET LOOPBACK
4821 057400  004737  062072                    JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
4822 057404  012704  062530                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4823 057410  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4824 057414  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4825 057420                                    FORCERROR       182$                    ;@@DFORCE ERROR IF FORCER=1
4826 057434  103407                            BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
4827 057436  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4828 057440                                    NEXT.ERRNO
4829 057440                             182$:  ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057440  104455                                                                 TRAP    C$ERDF
     057442  001520                                                                 .WORD   848
     057444  060441                                                                 .WORD   T194SSR
     057446  012046                                                                 .WORD   PKTSSR
4830 057450  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4831 057454                             190$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057454  104406                                                                 TRAP    C$CLP1
4832                                    ;       Do a WRITE FORMAT to set IRESV4-->IRSTR = 0
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
4833 057456  005000                        CLR     R0                      ;SET IRESV4==>IRSTR=0
4834 057460  004737  062212                JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
4835 057464  012704  062530                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4836 057470  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4837 057474  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4838 057500                                FORCERROR       202$                    ;@@DFORCE ERROR IF FORCER=1
4839 057514  103407                        BCS     210$                    ;BR IF CARRY SET (GOOD RETURN)
4840 057516  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4841 057520                                NEXT.ERRNO
4842 057520                        202$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057520  104455                                                                TRAP    C$ERDF
     057522  001521                                                                .WORD   849
     057524  060662                                                                .WORD   T198SSR
     057526  012046                                                                .WORD   PKTSSR
4843 057530  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4844 057534                        210$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057534  104406                                                                TRAP    C$CLP1
4845                               ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
4846 057536  012700  000001                MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR=1
4847 057542  004737  062212                JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
4848 057546  012704  062530                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4849 057552  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4850 057556  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4851 057562                                FORCERROR       222$                    ;@@DFORCE ERROR IF FORCER=1
4852 057576  103407                        BCS     230$                    ;BR IF CARRY SET (GOOD RETURN)
4853 057600  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4854 057602                                NEXT.ERRNO
4855 057602                        222$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057602  104455                                                                TRAP    C$ERDF
     057604  001522                                                                .WORD   850
     057606  060662                                                                .WORD   T198SSR
     057610  012046                                                                .WORD   PKTSSR
4856 057612  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4857 057616                        230$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057616  104406                                                                TRAP    C$CLP1
4858                               ;       Do a READ FIFO with tape direction IN to read data
4859 057620  012700  000001                MOV     #1,R0                   ;SET READ BYTE COUNT
4860 057624  004737  062116                JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
4861 057630  012704  062530                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4862 057634  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4863 057640  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4864 057644                                FORCERROR       282$                    ;@@DFORCE ERROR IF FORCER=1
4865 057660  103407                        BCS     290$                    ;BR IF CARRY SET (GOOD RETURN)
4866 057662  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4867 057664                                NEXT.ERRNO
4868 057664                        282$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057664  104455                                                                TRAP    C$ERDF
     057666  001523                                                                .WORD   851
     057670  060550                                                                .WORD   T196SSR
     057672  012046                                                                .WORD   PKTSSR
4869 057674  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4870 057700                        290$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057700  104406                                                                TRAP    C$CLP1
4871                               ;       If Data read from FIFO NOT= to Data sent Then Print Error
4872 057702  004737  062270                JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4873 057706  012701  060132                MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4874 057712  012702  062422                MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
4875 057716  013711  002312                  MOV     DATA,(R1)               ;SET EXPD WORD #8 = DATA
4876 057722  016261  000002  000002          MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
4877 057730  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4878 057732  012701  062402                  MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4879 057736  012702  060112                  MOV     #T19EXP,R2              ;EXPD ADDRESS
4880 057742  012703  000022                  MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
4881 057746  004737  011500                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4882 057752                                  FORCERROR       302$,NOTSSR     ;SSD
4883 057762  103404                          BCS     310$                    ;BR IF YES
4884 057764                                  NEXT.ERRNO
4885 057764                          302$:   ERRHRD  ERRNO,T19RSTR,MSGSUB    ;REPORT ERROR
     057764  104456                                                                  TRAP    C$ERHRD
     057766  001524                                                                  .WORD   852
     057770  061700                                                                  .WORD   T19RSTR
     057772  013742                                                                  .WORD   MSGSUB
4886 057774                          310$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057774  104406                                                                  TRAP    C$CLP1
4887
4888
4889 057776                                  FORCEXIT        355$            ;SSD
4890 060006  013703  002316                  MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
4891 060012  020327  003062                  CMP     R3,#TBLEND              ;END OF TSTBLK?
4892 060016  103002                          BHIS    355$                    ;BR IF YES
4893 060020  000137  057034                  JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
4894 060024                          355$:
4895
4896 060024                                  ENDSUB                          ;////////////// END SUBTEST //////////
     060024                                                                         L10072:
     060024  104403                                                                  TRAP    C$ESUB
4897
4898 060026  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
4899 060032  001402                          BEQ     360$                    ;BRANCH IF NOT
4900 060034  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4901 060040                          360$:
4902
4903 060040                                  EXIT    TST                     ;////////// EXIT TEST //////////
     060040  104432                                                                  TRAP    C$EXIT
     060042  002602                                                                  .WORD   L10066-.
4904
4905
4906                                 ;+
4907                                 ;LOCAL STORAGE FOR THIS TEST
4908                                 ;-
4909
4910 060044  000000                  T19PREV:        .WORD   0               ;DRIVE SIGNAL 1-0 TRANSITION FLAG
4911                                 ;+
4912                                 ; LOOPBACK DRIVE SIGNAL TABLE
4913                                 ; THIS TABLE IS USED BY T19CNVT TO SETUP
4914                                 ; A DRIVE PATTERN FROM THE TEST DATA INPUT PATTERN.
4915                                 ;
4916                                 ; WRITE CONTROL SIGNALS ARE OF FORM WC.XXX
4917                                 ; WRITE FORMAT SIGNALS ARE OF FORM WF.XXXX
4918                                 ;-
4919 060046                          T19BFC1:                                ;WRITE CONTROL DRIVE SIGNALS
4920 060046  000001                          WC.IGO                          ;IGO==>IFPT     DATA<0>
4921 060050  000002                          WC.IFEN                         ;IFEN==>IFBY    DATA<1>
4922 060052  000004                          WC.IRWU                         ;IRWU==>IRWD    DATA<2>
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
4923 060054  000010                        WC.IREW                 ;IREW==>IDBY    DATA<3>
4924 060056  002000                        WF.IERASE*256.          ;IFAD==>ILDP    DATA<4>
4925 060060  000040                        WC.I1TAD                ;ITAD1==>IONL   DATA<5>
4926 060062  000100                        WC.IOTAD                ;ITAD0==>IRDY   DATA<6>
4927 060064  000200                        WC.IFAD                 ;IERASE==>ISPEED DATA<7>
4928 060066  004000                        WF.IEDIT*256.           ;IEDIT==>IHER   DATA<8>
4929 060070  010000                        WF.IWFM*256.            ;IWFM==>IFMK    DATA<9>
4930 060072  020000                        WF.IREV*256.            ;IREV==>ICER    DATA<10>
4931 060074  040000                        WF.IWRT*256.            ;IWRT==>IIDENT  DATA<11>
4932 060076  100000                        WF.IHISP*256.           ;IHISP==>IEOT   DATA<12>
4933 060100  000000                        .WORD   0               ;IRESV2 (UNUSED)DATA<13>
4934 060102  000000                        .WORD   0               ;IRESV1 (UNUSED)DATA<14>
4935 060104  000000                        .WORD   0               ;PARERR (UNTESTED)DATA<15>
4936
4937 060106                        T19MSK:                         ;MASK OF UNTESTED BITS IN READ STATUS BYTES
4938                                                               ;UNTESTED BITS ARE SET TO 1
4939 060106      377                       .BYTE   ^C<000>         ;BYTE 0 MASK
4940 060107      037                       .BYTE   ^C<340>         ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
4941 060110      360                       .BYTE   ^C<017>         ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
4942 060111      000                       .BYTE   0               ;MAKE IT EVEN
4943
4944 060112                        T19EXP:                         ;BEGIN EXPECTED DATA BUFFER
4945 060112  000000                        .WORD   0               ;MESSAGE TYPE
4946 060114  000000                        .WORD   0               ;DATA FIELD LENGTH
4947 060116  000000                        .WORD   0               ;RBPCR
4948 060120  000000                        .WORD   0               ;XST0
4949 060122  000000                        .WORD   0               ;XST1
4950 060124  000000                        .WORD   0               ;XST2
4951 060126  000000                        .WORD   0               ;XST3
4952 060130  000000                        .WORD   0               ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
4953 060132                        T19EXSTA: .BLKB 64.             ;EXPECTED READ STATUS AND WRITE FIFO DATA
4954 060232                        T19EXEND:                               ;END EXPECTED DATA BUFFER
4955                               ;*
4956                               ;LOCAL TEXT MESSAGES FOR TEST
4957                               ;-
4958
4959 060232  124  162  141  TST19ID:        .ASCIZ  'Transport Bus Interface Loopback'
4960 060273  127  122  111  T19SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
4961 060330  127  122  111  T192SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
4962 060374  127  122  111  T193SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
4963 060441  127  122  111  T194SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
4964 060504  127  122  111  T195SSR:.ASCIZ  'WRITE SUBSYSTEM (Write FIFO) Failed'
4965 060550  127  122  111  T196SSR:.ASCIZ  'WRITE SUBSYSTEM (Read FIFO) Failed'
4966 060613  127  122  111  T197SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Control) Failed'
4967 060662  127  122  111  T198SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Format) Failed'
4968 060730  106  111  106  T191CMP:.ASCIZ  'FIFO Status in WORD #9 Incorrect after Initialize'
4969 061012  122  145  141  T192CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
4970 061106  124  141  160  T193CMP:.ASCIZ  'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
4971 061174  122  145  141  T195CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO Data'
4972 061250  106  111  106  T196CMP:.ASCIZ  'FIFO Status (in WORD #9) Incorrect after READ FIFO'
4973 061333  124  141  160  T197CMP:.ASCIZ  'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
4974 061421  103  157  156  T198CMP:.ASCIZ  'Control Signal Loopback Data Error, Data is in WORD #8'
4975 061510  122  145  141  T199CMP:.ASCIZ  'Read/Write Loopback Data Error, Data is in WORD #8'
4976 061573  114  157  157  T19WSTR:.ASCIZ  'Loopback Data Error when strobed by Write strobe, Data is in WORD #8'
4977 061700  114  157  157  T19RSTR:.ASCIZ  'Loopback Data Error when strobed by Read Strobe, Data is in WORD #8'
4978
4979                                       .EVEN
```

```
4980
4981                                    ;+
4982                                    ; CLEAR MESSAGE BUFFER
4983                                    ;-
4984 062004                             T19CLRBUF:                                      ;
4985 062004                                     SAVREG                                  ;SAVE R1-R5 UNTIL NEXT RETURN
4986 062010  012701  062402                     MOV     #T19BFR,R1                      ;GET MESSAGE BUFFER ADDRESS
4987 062014  012702  000120                     MOV     #T19BEND-T19BFR,R2              ;SIZE OF MESSAGE BUFFER IN BYTES
4988 062020  105021                     10$:    CLRB    (R1)+                           ;CLEAR A BYTE
4989 062022  005302                             DEC     R2                              ;DONE?
4990 062024  003375                             BGT     10$                             ;BR IF NO
4991 062026  000207                             RTS     PC                              ;RETURN
4992
4993                                    ;+
4994                                    ; SETUP T19PK2 PACKET FOR READ STATUS
4995                                    ;-
4996 062030                             T19SRD:
4997 062030  004737  062004                     JSR     PC,T19CLRBUF                    ;CLEAR MESSAGE BUFFER
4998 062034  012700  062540                     MOV     #T19DT2,R0                      ;WRTTE SUBSYSTEM DATA BUFFER
4999 062040  112720  000005                     MOVB    #PW.RDSTATUS,(R0)+              ;STORE READ STATUS COMMAND IN BSEL0
5000 062044  105010                             CLRB    (R0)                            ;CLEAR BSEL1
5001 062046  000207                             RTS     PC                              ;RETURN
5002
5003                                    ;+
5004                                    ; SETUP T19PK2 PACKET FOR WRITE MISC Reset Tape Status F-FLOPS
5005                                    ;-
5006 062050                             T19RSFIF:
5007 062050  004737  062004                     JSR     PC,T19CLRBUF                    ;CLEAR MESSAGE BUFFER
5008 062054  012700  062540                     MOV     #T19DT2,R0                      ;WRITE SUBSYSTEM DATA BUFFER
5009 062060  112720  000010                     MOVB    #PW.WMISC,(R0)+                 ;STORE WRITE MISCELLANEOUS IN BSEL0
5010 062064  112710  000030                     MOVB    #MS.RSFIF!MS.RSTAP,(R0)         ;STORE BSEL1 CLEAR FIFO CODES
5011 062070  000207                             RTS     PC                              ;RETURN
5012
5013                                    ;+
5014                                    ; SETUP T19PK2 PACKET FOR WRITE NPR
5015                                    ;
5016                                    ; INPUT:
5017                                    ;       R0 CONTAINS BSEL1 NPR DATA
5018                                    ;
5019                                    ;       SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
5020                                    ;-
5021 062072                             T19SNPR:
5022 062072  004737  062004                     JSR     PC,T19CLRBUF                    ;CLEAR MESSAGE BUFFER
5023 062076  012701  062540                     MOV     #T19DT2,R1                      ;WRITE SUBSYSTEM DATA BUFFER
5024 062102  112721  000011                     MOVB    #PW.WNPR,(R1)+                  ;STORE WRITE NPR IN BSEL0
5025 062106  052700  000020                     BIS     #NP.WRP,R0                      ;DON'T WRITE WRONG PARITY
5026 062112  110011                             MOVB    R0,(R1)                         ;STORE NPR DATA IN BSEL1
5027 062114  000207                             RTS     PC                              ;RETURN
5028
5029                                    ;+
5030                                    ; SETUP T19PK2 PACKET FOR READ FIFO
5031                                    ;
5032                                    ; INPUT:
5033                                    ;       R0 CONTAINS SEL2 BYTE COUNT
5034                                    ;-
5035 062116                             T19RFIF:
5036 062116  004737  062004                     JSR     PC,T19CLRBUF                    ;CLEAR MESSAGE BUFFER
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
5037 062122  012701  062540                MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5038 062126  112721  000003                MOVB    #PW.RFIFO,(R1)+         ;STORE READ FIFO IN BSEL0
5039 062132  110021                        MOVB    R0,(R1)+                ;STORE BYTE COUNT IN BSEL1
5040 062134  000207                        RTS     PC                      ;RETURN
5041                                ;+
5042                                ; SETUP T19PK2 PACKET FOR WRITE FIFO
5043                                ;
5044                                ; INPUT:
5045                                ;       R0 CONTAINS BYTE COUNT
5046                                ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
5047                                ;-
5048 062136                        T19WFIF:
5049 062136                                SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
5050 062142  004737  062004                JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
5051 062146  012702  062540                MOV     #T19DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
5052 062152  112722  000004                MOVB    #PW.WFIFO,(R2)+         ;STORE WRITE FIFO IN BSEL0
5053 062156  110022                        MOVB    R0,(R2)+                ;STORE BYTE COUNT IN BSEL1
5054 062160  005022                        CLR     (R2)+                   ;CLEAR SEL2 (UNUSED)
5055 062162  112122                10$:    MOVB    (R1)+,(R2)+             ;STORE DATA PATTERN BYTE
5056 062164  005300                        DEC     R0                      ;DONE ALL BYTES?
5057 062166  003375                        BGT     10$                     ;BR IF NO
5058 062170  000207                        RTS     PC                      ;RETURN
5059                                ;+
5060                                ; SETUP T19PK2 FOR WRITE CONTROL
5061                                ;
5062                                ; INPUT:
5063                                ;       R0 CONTAINS DRIVING DATA PATTERN
5064                                ;-
5065 062172                        T19WCTL:
5066 062172  004737  062004                JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
5067 062176  012701  062540                MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5068 062202  112721  000006                MOVB    #PW.WCTL,(R1)+          ;STORE WRITE CONTROL IN BSEL0
5069 062206  110021                        MOVB    R0,(R1)+                ;STORE DATA WORD IN BSEL1
5070 062210  000207                        RTS     PC                      ;RETURN
5071                                ;+
5072                                ; SETUP T19PK2 FOR WRITE FORMAT TRANSPORT REGISTER
5073                                ;
5074                                ; INPUT:
5075                                ;       R0 CONTAINS DRIVING DATA PATTERN
5076                                ;-
5077 062212                        T19WFMT:
5078 062212  004737  062004                JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
5079 062216  012701  062540                MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5080 062222  112721  000007                MOVB    #PW.WFMT,(R1)+          ;STORE WRITE FORMAT IN BSEL0
5081 062226  110021                        MOVB    R0,(R1)+                ;STORE DATA WORD IN BSEL1
5082 062230  000207                        RTS     PC                      ;RETURN
5083                                ;+
5084                                ; SETUP T19PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
5085                                ;-
5086 062232                        T19SEXT:
5087 062232  012700  062540                MOV     #T19DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
5088 062236  112720  000010                MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
5089 062242  112710  000200                MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
5090 062246  000207                        RTS     PC                      ;RETURN
5091                                ;+
5092                                ; CLEAR EXPECTED DATA MESSAGE BUFFER
5093                                ;-
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
5094 062250                            T19CLEXP:
5095 062250  012701  060112                MOV     #T19EXP,R1              ;GET EXPD ADDRESS
5096 062254  012700  000120                MOV     #T19EXEND-T19EXP,R0     ;GET EXPD SIZE
5097 062260  105021                10$:    CLRB    (R1)+                   ;CLEAR A BYTE
5098 062262  005300                        DEC     R0                      ;DONE?
5099 062264  003375                        BGT     10$                     ;BR IF NO
5100 062266  000207                        RTS     PC                      ;RETURN
5101
5102                                   ;+
5103                                   ;Set WORDS 0-7 of expd message BUFFER = to recv since not testing
5104                                   ;-
5105 062270                            T19SETEXP:
5106 062270  012702  060112                MOV     #T19EXP,R2              ;GET EXPD
5107 062274  012703  062402                MOV     #T19BFR,R3              ;GET READ STATUS RECV BUFFER
5108 062300  012700  000010                MOV     #8.,R0                  ;SET WORDS 0-7 EXP=RECV
5109 062304  012322                5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
5110 062306  005300                        DEC     R0                      ;DONE WORDS 0-7 WORDS?
5111 062310  003375                        BGT     5$                      ;BR IF NO
5112 062312  000207                        RTS     PC                      ;RETURN
5113
5114                                   ;+
5115                                   ; CONVERT A TEST PATTERN DATA WORD TO LOOPBACK DRIVE SIGNALS
5116                                   ;
5117                                   ; INPUTS:
5118                                   ;
5119                                   ;       R0      TEST PATTERN
5120                                   ;
5121                                   ; IMPLICIT INPUTS:
5122                                   ;
5123                                   ;       T19BFCTL  - CONTAINS WRITE CONTROL / WRITE FORMAT CONVERSION BITS
5124                                   ;
5125                                   ; OUTPUTS:
5126                                   ;
5127                                   ;       R0      - LOW BYTE CONTAINS WRITE CONTROL DATA
5128                                   ;               - HIGH BYTE CONTAINS WRITE FORMAT DATA
5129                                   ;-
5130 062314                            T19CNVT:
5131 062314                                SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
5132 062320  012701  060046                MOV     #T19BFCTL,R1            ;CONVERSION TABLE ADDRESS
5133 062324  005002                        CLR     R2                      ;INIT RESULT OF CONVERSION
5134 062326  012703  000020                MOV     #16.,R3                 ;BIT COUNT
5135 062332  006000                10$:    ROR     R0                      ;IS THIS BIT EQUAL TO 1?
5136 062334  103001                        BCC     20$                     ;BR IF NO
5137 062336  051102                        BIS     (R1),R2                 ;SET CONVERTED BIT
5138 062340  005721                20$:    TST     (R1)+                   ;POINT TO NEXT BIT IN CONVERSION TABLE
5139 062342  005303                        DEC     R3                      ;DONE?
5140 062344  003372                        BGT     10$                     ;BR IF NO
5141 062346  010200                        MOV     R2,R0                   ;COPY RESULT
5142 062350  000207                        RTS     PC                      ;RETURN
5143
5144
5145
5147         062360                        .=<.+10>&177770
5149                                   ;
5150                                   ;WRITE CHARACTERISTICS COMMAND PACKET
5151                                   ;
5152 062360                            T19PACKET:                              ;COMMAND PACKET FOR TEST
```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```
5153 062360  100004                    .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
5154 062362  062370                    .WORD   T19DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
5155 062364  000000                    .WORD   0
5156 062366  000012                    .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
5157
5158 062370                          T19DATA:                          ;CHARACTERISTICS DATA BLOCK
5159 062370  062402                    .WORD   T19BFR                  ;ADDRESS OF MESSAGE BUFFER
5160 062372  000000                    .WORD   0
5161 062374  000024                    .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
5162 062376  000000                    .WORD   0                       ;ESS,ENB,EAI,ERI
5163 062400  000007                    .WORD   7                       ;EXTENDED FEATURES UNIT NO.
5164
5165
5166                                  ;MESSAGE BUFFER FOR ALL TEST 8 COMMANDS
5167
5168 062402                          T19BFR:                           ;BEGIN MESSAGE BUFFER
5169 062402  000000                    .WORD   0                       ;MESSAGE TYPE
5170 062404  000000                    .WORD   0                       ;DATA FIELD LENGTH
5171 062406  000000                    .WORD   0                       ;RBPCR
5172 062410  000000                    .WORD   0                       ;XST0
5173 062412  000000                    .WORD   0                       ;XST1
5174 062414  000000                    .WORD   0                       ;XST2
5175 062416  000000                    .WORD   0                       ;XST3
5176 062420  000000                    .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
5177 062422                          T19BFSTA: .BLKB 64.               ;READ STATUS AND WRITE FIFO BUFFER
5178 062522                          T19BEND:                          ;END OF MESSAGE BUFFER
5179                                  ;
5180                                  ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
5181                                  ;
5183         062530                      .=<.+10>&177770
5185 062530                          T19PK2:
5186 062530  100006                    .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
5187 062532  062540                    .WORD   T19DT2                  ;LOW ADDRESS OF DATA BLOCK
5188 062534  000000                    .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
5189 062536  000012                    .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
5190
5191 062540                          T19DT2:                           ;DATA BLOCK
5192 062540     000                    .BYTE   0                       ;BSEL0
5193 062541     000                    .BYTE   0                       ;BSEL1
5194 062542  000000                    .WORD   0                       ;SEL2
5195 062544                            .BLKB   64.                     ;WRITE FIFO DATA OUTPUT BUFFER
5196
5197
5198 062644                            ENDTST
     062644                                                                 L10066:
     062644  104401                                                                TRAP    C$ETST
5199                                    .SBTTL  TEST  9: READ/WRITE DATA PARITY TEST
5200                                  ;++
5201                                  ; TEST DESCRIPTION:
5202                                  ;
5203                                  ;       This test verifies that the Write Data Parity generator
5204                                  ;       and the Read Data Parity checker operate properly. The
5205                                  ;       Transport Bus signal loopback mode is enabled and a
5206                                  ;       Set Wrong parity function is executed. Then various
5207                                  ;       Write Subsystem Memory functions are performed to
5208                                  ;       write data to and from the FIFO in loopback mode.
5209                                  ;       The program then checks to insure a Read Data parity
```

```
5210                                    ;       error occurred.
5211                                    ;       A Reset FIFO is done and the Read Data parity
5212                                    ;       error bit is again tested to insure it cleared.
5213                                    ;       Finally a Clear wrong parity function is done
5214                                    ;       and it is verified the data word can pass ir loopback
5215                                    ;       mode without setting Read Data parity error.
5216                                    ;
5217                                    ; TEST STEPS:
5218                                    ;
5219                                    ;  REPEAT FOR LOOPCNT
5220                                    ;    BEGIN
5221                                    ;       Write to TSSR register to soft initialize the controller
5222                                    ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
5223                                    ;       If Extended Features Hardware Switch Clear then:
5224                                    ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
5225                                    ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
5226                                    ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
5227                                    ;    BEGIN
5228                                    ;       (* Verify Write Wrong Parity Sets Parity Error *)
5229                                    ;       Do a WRITE NPR to set loopback and tape direction OUT
5230                                    ;              and SET Write Wrong Parity.
5231                                    ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
5232                                    ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
5233                                    ;       Do a READ FIFO with tape direction OUT to load tape out write latch
5234                                    ;         (this is when wrong parity (IWP) is set)
5235                                    ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 0 (sets read strobe low)
5236                                    ;             (Read Strobe sets PAR IN H [Parity Error])
5237                                    ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
5238                                    ;       Do a Write Subsystem READ STATUS
5239                                    ;       If Read Data parity error NOT=1  Then Print Error
5240                                    ;       Do a Write Misc to RESET FIFO
5241                                    ;       Do a Write Subsystem READ STATUS
5242                                    ;       If Read Data parity error NOT=0  Then Print Error
5243                                    ;
5244                                    ;       (* Verify Data can be transferred without a Parity Error *)
5245                                    ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
5246                                    ;       Do a WRITE NPR to set loopback and tape direction OUT
5247                                    ;              and CLEAR Write Wrong Parity.
5248                                    ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
5249                                    ;       Do a READ FIFO with tape direction OUT to load tape out write latch
5250                                    ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 0 (sets read strobe low)
5251                                    ;             (Read Strobe should NOT set PAR IN H [Parity Error] here)
5252                                    ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
5253                                    ;       Do a Write Subsystem READ STATUS
5254                                    ;       If Read Data parity error NOT=0  Then Print Error
5255                                    ;
5256                                    ;    END
5257                                    ;--
5258
5259
5260 062646                                     BGNTST
     062646                                                                                    T9::
5265 062646  012700  065232                     MOV     #TST20ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
5266 062652  004737  016510                     JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
5267 062656  012737  000012  002216            MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
5268 062664                                T20LOOP:
5269                 .
```

TEST 9: READ/WRITE DATA PARITY TEST

```
5270 062664                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////
     062664                                                                                      T9.1:
     062664  104402                                                                             TRAP    C$BSUB
5271                                ;      Write to TSSR register to soft initialize the controller
5272 062666                         5$:
5273 062666  004737  015774                JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
5274 062672  103405                        BCS     10$                     ;BR IF SOFT INIT OKAY
5275 062674  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5276 062676                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     062676  104455                                                                             TRAP    C$ERDF
     062700  001604                                                                             .WORD   900
     062702  003652                                                                             .WORD   SFIERR
     062704  012034                                                                             .WORD   SFIMSG
5277                                ;      Do WRITE CHARACTERISTICS to check for Extended Features Switch
5278 062706  005037  002222         10$:   CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
5279 062712  012704  066430               MOV     #T20PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
5280 062716  004737  010662               JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
5281 062722                                FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
5282 062736  103407                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
5283 062740  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5284 062742                                NEXT.ERRNO
5285 062742                         12$:   ERRDF   ERRNO,T20SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     062742  104455                                                                             TRAP    C$ERDF
     062744  001605                                                                             .WORD   901
     062746  065261                                                                             .WORD   T20SSR
     062750  012046                                                                             .WORD   PKTSSR
5286 062752  005237  002222               INC     FATFLG                  ;SET FATAL ERROR FLAG
5287 062756                         15$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     062756  104406                                                                             TRAP    C$CLP1
5288                                ;      If Extended Features Hardware Switch Clear then:
5289                                ;        Do Write Subsystem Write Miscellaneous to Set Extended Features.
5290 062760  012701  066452               MOV     #T20BFR,R1              ;MESSAGE BUFFER ADDRESS
5291 062764  032761  000200  000012       BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
5292 062772  001026                        BNE     30$                     ;BR IF YES
5293 062774  004737  066346               JSR     PC,T20SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
5294 063000  012704  066600               MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5295 063004  010465  000000               MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5296 063010  004737  016336               JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5297 063014                                FORCERROR       22$                     ;@@DFORCE ERROR IF FORCER=1
5298 063030  103407                        BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
5299 063032  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5300 063034                                NEXT.ERRNO
5301 063034                         22$:   ERRDF   ERRNO,T202SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063034  104455                                                                             TRAP    C$ERDF
     063036  001606                                                                             .WORD   902
     063040  065316                                                                             .WORD   T202SSR
     063042  012046                                                                             .WORD   PKTSSR
5302 063044  005237  002222               INC     FATFLG                  ;SET FATAL ERROR FLAG
5303 063050                         30$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063050  104406                                                                             TRAP    C$CLP1
5304                                ;      Do WRITE CHARACTERISTICS to select reserved unit 7
5305 063052  012704  066430               MOV     #T20PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
5306 063056  004737  010662               JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
5307 063062                                FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
5308 063076  103407                        BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
5309 063100  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5310 063102                                NEXT.ERRNO
```

TEST 9: READ/WRITE DATA PARITY TEST

```
5311 063102                          42$:   ERRDF   ERRNO,T20SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     063102  104455                                                                TRAP    C$ERDF
     063104  001607                                                                .WORD   903
     063106  065261                                                                .WORD   T20SSR
     063110  012046                                                                .WORD   PKTSSR
5312 063112  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
5313 063116                          50$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063116  104406                                                                TRAP    C$CLP1
5314
5315
5316                                 ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
5317 063120  012703  002752                 MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
5318 063124  012337  002312          100$:  MOV     (R3)+,DATA              ;GET A TEST PATTERN
5319 063130  042737  177400  002312        BIC     #^C<377>,DATA           ;DATA IS BYTE
5320 063136  010337  002316                 MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
5321                                 ;       Do a WRITE NPR to set loopback and tape direction OUT and
5322                                 ;                 and SET Write Wrong Parity.
5323 063142  012700  000100                 MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
5324 063146  052700  000040                 BIS     #NP.LOOP,R0             ;SET LOOPBACK
5325 063152  042700  000020                 BIC     #NP.WRP,R0              ;SET WRITE WRONG PARITY (INVERTED)
5326 063156  004737  066216                 JSR     PC,T20WNPR              ;SETUP T20PK2 FOR WRITE NPR
5327 063162  012704  066600                 MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5328 063166  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5329 063172  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5330 063176                                 FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
5331 063212  103407                         BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
5332 063214  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5333 063216                                 NEXT.ERRNO
5334 063216                          102$:  ERRDF   ERRNO,T204SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063216  104455                                                                TRAP    C$ERDF
     063220  001610                                                                .WORD   904
     063222  065427                                                                .WORD   T204SSR
     063224  012046                                                                .WORD   PKTSSR
5335 063226  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
5336 063232                          105$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063232  104406                                                                TRAP    C$CLP1
5337                                 ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
5338 063234  012700  000001                 MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR = 1
5339 063240  004737  066312                 JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
5340 063244  012704  066600                 MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5341 063250  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5342 063254  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5343 063260                                 FORCERROR       112$                    ;@@DFORCE ERROR IF FORCER=1
5344 063274  103407                         BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
5345 063276  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5346 063300                                 NEXT.ERRNO
5347 063300                          112$:  ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063300  104455                                                                TRAP    C$ERDF
     063302  001611                                                                .WORD   905
     063304  065601                                                                .WORD   T208SSR
     063306  012046                                                                .WORD   PKTSSR
5348 063310  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
5349 063314                          120$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063314  104406                                                                TRAP    C$CLP1
5350                                 ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
5351 063316  012700  000001                 MOV     #1,R0                   ;WRITE 1 BYTE
5352 063322  012701  002312                 MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
```

```
5353 063326  004737  066256                JSR     PC,T20WFIF              ;SETUP T20PK2 FOR WRITE FIFO
5354 063332  012704  066600                MOV     @T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5355 063336  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5356 063342  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5357 063346                                FORCERROR       152$                    ;@@DFORCE ERROR IF FORCER=1
5358 063362  103407                        BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
5359 063364  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5360 063366                                NEXT.ERRNO
5361 063366                        152$:   ERRDF   ERRNO,T205SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063366  104455                                                                TRAP    C$ERDF
     063370  001612                                                                .WORD   906
     063372  065472                                                                .WORD   T205SSR
     063374  012046                                                                .WORD   PKTSSR
5362 063376  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
5363 063402                        160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063402  104406                                                                TRAP    C$CLP1
5364
5365                               ;       Do a READ FIFO with tape direction OUT to load tape out write latch
                                   ;         (this is when wrong parity (IWP) is set)
5366 063404  012700  000001                MOV     #1,R0                   ;SET READ BYTE COUNT
5367 063410  004737  066236                JSR     PC,T20RFIF              ;SETUP T20PK2 FOR READ FIFO
5368 063414  012704  066600                MOV     @T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5369 063420  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5370 063424  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5371 063430                                FORCERROR       172$                    ;@@DFORCE ERROR IF FORCER=1
5372 063444  103407                        BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
5373 063446  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5374 063450                                NEXT.ERRNO
5375 063450                        172$:   ERRDF   ERRNO,T206SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063450  104455                                                                TRAP    C$ERDF
     063452  001613                                                                .WORD   907
     063454  065536                                                                .WORD   T206SSR
     063456  012046                                                                .WORD   PKTSSR
5376 063460  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
5377 063464                        180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063464  104406                                                                TRAP    C$CLP1
5378
5379                               ;       Do a WRITE FORMAT  to set IRESV4-->IRSTR = 0 (sets read strobe low)
                                   ;               (Read Strobe sets PAR IN H [Parity Error])
5380 063466  005000                        CLR     R0                      ;IRESV4-->IRSTR = 0
5381 063470  004737  066312                JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
5382 063474  012704  066600                MOV     @T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5383 063500  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5384 063504  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5385 063510                                FORCERROR       192$                    ;@@DFORCE ERROR IF FORCER=1
5386 063524  103407                        BCS     200$                    ;BR IF CARRY SET (GOOD RETURN)
5387 063526  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5388 063530                                NEXT.ERRNO
5389 063530                        192$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063530  104455                                                                TRAP    C$ERDF
     063532  001614                                                                .WORD   908
     063534  065601                                                                .WORD   T208SSR
     063536  012046                                                                .WORD   PKTSSR
5390 063540  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
5391 063544                        200$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063544  104406                                                                TRAP    C$CLP1
5392
5393 063546  012700  000001        ;       Do a WRITE FORMAT  to set IRESV4-->IRSTR = 1 (sets read strobe high)
5394 063552  004737  066312                MOV     #WF.I4RES,R0            ;IRESV4-->IRSTR = 1
                                           JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
```

TEST 9: READ/WRITE DATA PARITY TEST

```
5395 063556  012704  066600                  MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5396 063562  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5397 063566  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5398 063572                                  FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
5399 063606  103407                         BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
5400 063610  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5401 063612                                  NEXT.ERRNO
5402 063612                          212$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063612  104455                                                                  TRAP    C$ERDF
     063614  001615                                                                  .WORD   909
     063616  065601                                                                  .WORD   T208SSR
     063620  012046                                                                  .WORD   PKTSSR
5403 063622  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
5404 063626                          220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063626  104406                                                                  TRAP    C$CLP1
5405                                 ;       Do a Write Subsystem READ STATUS
5406 063630  004737  066176                  JSR     PC,T20SRD               ;SETUP PACKET FOR READ STATUS
5407 063634  012704  066600                  MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5408 063640  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5409 063644  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5410 063650                                  FORCERROR       232$                    ;@@DFORCE ERROR IF FORCER=1
5411 063664  103407                         BCS     240$                    ;BR IF CARRY SET (GOOD RETURN)
5412 063666  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5413 063670                                  NEXT.ERRNO
5414 063670                          232$:   ERRDF   ERRNO,T203SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063670  104455                                                                  TRAP    C$ERDF
     063672  001616                                                                  .WORD   910
     063674  065362                                                                  .WORD   T203SSR
     063676  012046                                                                  .WORD   PKTSSR
5415 063700  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
5416 063704                          240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063704  104406                                                                  TRAP    C$CLP1
5417                                 ;       If Read Data parity error NOT=1  Then Print Error
5418 063706  004737  066404                  JSR     PC,T20SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
5419 063712  012701  065132                  MOV     #T20EXSTA,R1            ;GET EXPECTED READ STATUS
5420 063716  012702  066472                  MOV     #T208FSTA,R2            ;GET RECV READ STATUS
5421 063722  011211                         MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
5422 063724  016261  000002  000002          MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTED)
5423 063732  052711  100000                  BIS     #S1.PARERR,(R1)         ;SET EXP PAR ERR  =1
5424 063736  005000                         CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
5425 063740  012701  066452                  MOV     #T208FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
5426 063744  012702  065112                  MOV     #T20EXP,R2              ;EXPD ADDRESS
5427 063750  012703  000024                  MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
5428 063754  004737  011500                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
5429 063760                                  FORCERROR       252$,NOTSSR             ;@@D
5430 063770  103404                         BCS     260$                    ;BR IF YES
5431 063772                                  NEXT.ERRNO
5432 063772                          252$:   ERRHRD  ERRNO,T20SWP,MSGSTAT    ;REPORT ERROR
     063772  104456                                                                  TRAP    C$ERHRD
     063774  001617                                                                  .WORD   911
     063776  065647                                                                  .WORD   T20SWP
     064000  012350                                                                  .WORD   MSGSTAT
5433 064002                          260$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064002  104406                                                                  TRAP    C$CLP1
5434                                 ;       Do a Write Misc to RESET FIFO
5435 064004  012700  000020                  MOV     #MS.RSFIF,R0            ;SET RESET FIFO COMMAND
5436 064010  004737  066332                  JSR     PC,T20WMISC             ;SETUP T20PK2 FOR WRITE MISC
```

```
5437 064014  012704  066600                MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5438 064020  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5439 064024  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5440 064030                                FORCERROR       282$                    ;@@DFORCE ERROR IF FORCER=1
5441 064044  103407                        BCS     290$                    ;BR IF CARRY SET (GOOD RETURN)
5442 064046  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5443 064050                                NEXT.ERRNO
5444 064050                        282$:   ERRDF   ERRNO,T202SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064050  104455                                                                TRAP    C$ERDF
     064052  001620                                                                .WORD   912
     064054  065316                                                                .WORD   T202SSR
     064056  012046                                                                .WORD   PKTSSR
5445 064060  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
5446 064064                        290$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064064  104406                                                                TRAP    C$CLP1
5447                                ;       Do a Write Subsystem READ STATUS
5448                                ;       If Read Data parity error NOT=0  Then Print Error
5449 064066  004737  066404                JSR     PC,T20SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
5450 064072  012701  065132                MOV     #T20EXSTA,R1            ;GET EXPECTED READ STATUS
5451 064076  012702  066472                MOV     #T208FSTA,R2            ;GET RECV READ STATUS
5452 064102  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
5453 064104  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTED)
5454 064112  042711  100000                BIC     #S1.PARERR,(R1)         ;SET EXP PAR ERR  =0
5455 064116  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
5456 064120  012701  066452                MOV     #T208FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
5457 064124  012702  065112                MOV     #T20EXP,R2              ;EXPD ADDRESS
5458 064130  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
5459 064134  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
5460 064140                                FORCERROR       302$,NOTSSR             ;@@D
5461 064150  103404                        BCS     320$                    ;BR IF YES
5462 064152                                NEXT.ERRNO
5463 064152                        302$:   ERRHRD  ERRNO,T20RSF,MSGSTAT    ;REPORT ERROR
     064152  104456                                                                TRAP    C$ERHRD
     064154  001621                                                                .WORD   913
     064156  065756                                                                .WORD   T20RSF
     064160  012350                                                                .WORD   MSGSTAT
5464 064162                        320$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064162  104406                                                                TRAP    C$CLP1
5465
5466                                ;       (* Verify Data can be transferred without a Parity Error *)
5467 064164  012700  000001                ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
5468 064170  004737  066312                MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR = 1
5469 064174  012704  066600                JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
5470 064200  010465  000000                MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5471 064204  004737  016336                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5472 064210                                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5473 064224  103407                        FORCERROR       332$                    ;@@DFORCE ERROR IF FORCER=1
5474 064226  010001                        BCS     340$                    ;BR IF CARRY SET (GOOD RETURN)
5475 064230                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5476 064230                                NEXT.ERRNO
     064230  104455                332$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064232  001622                                                                TRAP    C$ERDF
     064234  065601                                                                .WORD   914
     064236  012046                                                                .WORD   T208SSR
5477 064240  005237  002222                                                        .WORD   PKTSSR
5478 064244                        INC     FATFLG                  ;SET FATAL ERROR FLAG
     064244  104406                340$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                TRAP    C$CLP1
```

```
5479
5480
5481 064246  012700  000100          ;       Do a WRITE NPR to set loopback and tape direction OUT and
5482 064252  052700  000040          ;                and CLEAR Write Wrong Parity.
5483 064256  052700  000020                  MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
5484 064262  004737  066216                  BIS     #NP.LOOP,R0             ;SET LOOPBACK
5485 064266  012704  066600                  BIS     #NP.WRP,R0              ;CLEAR WRITE WRONG PARITY (INVERTED)
5486 064272  010465  000000                  JSR     PC,T20WNPR              ;SETUP T20PK2 FOR WRITE NPR
5487 064276  004737  016336                  MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5488 064302                                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5489 064316  103407                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5490 064320  010001                          FORCERROR       352$                    ;@@DFORCE ERROR IF FORCER=1
5491 064322                                  BCS     360$                    ;BR IF CARRY SET (GOOD RETURN)
5492 064322                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
     064322  104455                          NEXT.ERRNO
     064324  001623                  352$:   ERRDF   ERRNO,T204SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064326  065427                                                                  TRAP    C$ERDF
     064330  012046                                                                  .WORD   915
5493 064332  005237  002222                                                          .WORD   T204SSR
5494 064336                                                                         .WORD   PKTSSR
     064336  104406                  INC     FATFLG                  ;SET FATAL ERROR FLAG
5495                                 360$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
5496 064340  012700  000001                                                        TRAP    C$CLP1
5497 064344  012701  002312          ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
5498 064350  004737  066256                  MOV     #1,R0                   ;WRITE 1 BYTE
5499 064354  012704  066600                  MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
5500 064360  010465  000000                  JSR     PC,T20WFIF              ;SETUP T20PK2 FOR WRITE FIFO
5501 064364  004737  016336                  MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5502 064370                                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5503 064404  103407                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5504 064406  010001                          FORCERROR       372$                    ;@@DFORCE ERROR IF FORCER=1
5505 064410                                  BCS     380$                    ;BR IF CARRY SET (GOOD RETURN)
5506 064410                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
     064410  104455                          NEXT.ERRNO
     064412  001624                  372$:   ERRDF   ERRNO,T205SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064414  065472                                                                  TRAP    C$ERDF
     064416  012046                                                                  .WORD   916
5507 064420  005237  002222                                                          .WORD   T205SSR
5508 064424                                                                         .WORD   PKTSSR
     064424  104406                  INC     FATFLG                  ;SET FATAL ERROR FLAG
5509                                 380$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
5510 064426  012700  000001                                                        TRAP    C$CLP1
5511 064432  004737  066236          ;       Do a READ FIFO with tape direction OUT to load tape out write latch
5512 064436  012704  066600                  MOV     #1,R0                   ;SET READ BYTE COUNT
5513 064442  010465  000000                  JSR     PC,T20RFIF              ;SETUP T20PK2 FOR READ FIFO
5514 064446  004737  016336                  MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5515 064452                                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5516 064466  103407                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5517 064470  010001                          FORCERROR       392$                    ;@@DFORCE ERROR IF FORCER=1
5518 064472                                  BCS     400$                    ;BR IF CARRY SET (GOOD RETURN)
5519 064472                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
     064472  104455                          NEXT.ERRNO
     064474  001625                  392$:   ERRDF   ERRNO,T206SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064476  065536                                                                  TRAP    C$ERDF
     064500  012046                                                                  .WORD   917
5520 064502  005237  002222                                                          .WORD   T206SSR
5521 064506                                                                         .WORD   PKTSSR
                                     INC     FATFLG                  ;SET FATAL ERROR FLAG
                                     400$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
```

```
        064506  104406                                                          TRAP    C$CLP1
5522                            ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 0 (sets read strobe low)
5523                            ;               (Read Strobe sets PAR IN H (Parity Error))
5524 064510  005000                    CLR     R0                      ;IRESV4==>IRSTR = 0
5525 064512  004737  066312            JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
5526 064516  012704  066600            MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5527 064522  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5528 064526  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5529 064532                            FORCERROR       412$                    ;@@DFORCE ERROR IF FORCER=1
5530 064546  103407                    BCS     420$                    ;BR IF CARRY SET (GOOD RETURN)
5531 064550  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5532 064552                            NEXT.ERRNO
5533 064552                    412$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        064552  104455                                                          TRAP    C$ERDF
        064554  001626                                                          .WORD   918
        064556  065601                                                          .WORD   T208SSR
        064560  012046                                                          .WORD   PKTSSR
5534 064562  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
5535 064566                    420$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        064566  104406                                                          TRAP    C$CLP1
5536                            ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
5537 064570  012700  000001            MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR = 1
5538 064574  004737  066312            JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
5539 064600  012704  066600            MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5540 064604  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5541 064610  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5542 064614                            FORCERROR       432$                    ;@@DFORCE ERROR IF FORCER=1
5543 064630  103407                    BCS     440$                    ;BR IF CARRY SET (GOOD RETURN)
5544 064632  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5545 064634                            NEXT.ERRNO
5546 064634                    432$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        064634  104455                                                          TRAP    C$ERDF
        064636  001627                                                          .WORD   919
        064640  065601                                                          .WORD   T208SSR
        064642  012046                                                          .WORD   PKTSSR
5547 064644  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
5548 064650                    440$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        064650  104406                                                          TRAP    C$CLP1
5549
5550                            ;       Do a Write Subsystem READ STATUS
5551 064652  004737  066176            JSR     PC,T20SRD               ;SETUP PACKET FOR READ STATUS
5552 064656  012704  066600            MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5553 064662  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5554 064666  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5555 064672                            FORCERROR       452$                    ;@@DFORCE ERROR IF FORCER=1
5556 064706  103407                    BCS     460$                    ;BR IF CARRY SET (GOOD RETURN)
5557 064710  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5558 064712                            NEXT.ERRNO
5559 064712                    452$:   ERRDF   ERRNO,T203SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        064712  104455                                                          TRAP    C$ERDF
        064714  001630                                                          .WORD   920
        064716  065362                                                          .WORD   T203SSR
        064720  012046                                                          .WORD   PKTSSR
5560 064722  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
5561 064726                    460$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        064726  104406                                                          TRAP    C$CLP1
5562                            ;       If Read Data parity error NOT=0  Then Print Error
```

TEST 9: READ/WRITE DATA PARITY TEST

```
5563 064730  004737  066404                  JSR     PC,T20SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
5564 064734  012701  065132                  MOV     #T20EXSTA,R1            ;GET EXPECTED READ STATUS
5565 064740  012702  066472                  MOV     #T208FSTA,R2            ;GET RECV READ STATUS
5566 064744  011211                          MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
5567 064746  016261  000002  000002          MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTED)
5568 064754  042711  100000                  BIC     #S1.PARERR,(R1)         ;SET EXP PAR ERR  =0
5569 064760  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
5570 064762  012701  066452                  MOV     #T208FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
5571 064766  012702  065112                  MOV     #T20EXP,R2              ;EXPD ADDRESS
5572 064772  012703  000024                  MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
5573 064776  004737  011500                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
5574 065002                                  FORCERROR       472$,NOTSSR     ;@@D
5575 065012  103404                          BCS     480$                    ;BR IF YES
5576 065014                                  NEXT.ERRNO
5577 065014                          472$:   ERRHRD  ERRNO,T20CWP,MSGSTAT    ;REPORT ERROR
     065014  104456                                                                  TRAP    C$ERHRD
     065016  001631                                                                  .WORD   921
     065020  066057                                                                  .WORD   T20CWP
     065022  012350                                                                  .WORD   MSGSTAT
5578 065024                          480$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     065024  104406                                                                  TRAP    C$CLP1
5579
5580 065026                                  FORCEXIT        555$            ;@@D
5581 065036  013703  002316                  MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
5582 065042  020327  003062                  CMP     R3,#TBLEND              ;END OF TSTBLK?
5583 065046  103002                          BHIS    555$                    ;BR IF YES
5584 065050  000137  063124                  JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
5585 065054                          555$:
5586
5587 065054                                  ENDSUB                          ;////////////// END SUBTEST //////////
     065054                                                                  L10074:
     065054  104403                                                                  TRAP    C$ESUB
5588
5589 065056  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
5590 065062  001402                          BEQ     560$                    ;BRANCH IF NOT
5591 065064  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
5592 065070                          560$:
5593 065070  004737  016456                  JSR     PC,TSTLOOP              ;DO ITERATIONS?
5594 065074  103002                          BCC     565$                    ;BR IF NO
5595 065076  000137  050556                  JMP     T18LOOP                 ;LOOP UNTIL ITERATIONS DONE
5596 065102                          565$:
5597 065102                                  EXIT    TST                     ;////////// EXIT TEST //////////
     065102  104432                                                                  TRAP    C$EXIT
     065104  001610                                                                  .WORD   L10073-.
5598
5599
5600                                 ;+
5601                                 ;LOCAL STORAGE FOR THIS TEST
5602                                 ;-
5603
5604
5605 065106                          T20MSK:                                 ;MASK OF UNTESTED BITS IN READ STATUS
5606                                                                          ;UNTESTED BITS ARE SET TO 1
5607 065106     377                          .BYTE   ^C<000>                 ;BYTE 0 MASK
5608 065107     037                          .BYTE   ^C<340>                 ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
5609 065110     360                          .BYTE   ^C<017>                 ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
5610 065111     000                          .BYTE   0                       ;MAKE IT EVEN
```

```
5611
5612 065112                                    T20EXP:                         ;BEGIN EXPECTED DATA BUFFER
5613 065112  000000                                .WORD   0                   ;MESSAGE TYPE
5614 065114  000000                                .WORD   0                   ;DATA FIELD LENGTH
5615 065116  000000                                .WORD   0                   ;RBPCR
5616 065120  000000                                .WORD   0                   ;XST0
5617 065122  000000                                .WORD   0                   ;XST1
5618 065124  000000                                .WORD   0                   ;XST2
5619 065126  000000                                .WORD   0                   ;XST3
5620 065130  000000                                .WORD   0                   ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
5621 065132                                    T20EXSTA: .BLKB 64.             ;EXPECTED READ STATUS AND WRITE FIFO DATA
5622 065232                                    T20EXEND:                               ;END EXPECTED DATA BUFFER
5623                                           ;+
5624                                           ;LOCAL TEXT MESSAGES FOR TEST
5625                                           ;-
5626
5627 065232      122     145     141   TST20ID:        .ASCIZ  'Read/Write Data Parity'
5628 065261      127     122     111   T20SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
5629 065316      127     122     111   T202SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
5630 065362      127     122     111   T203SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
5631 065427      127     122     111   T204SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
5632 065472      127     122     111   T205SSR:.ASCIZ  'WRITE SUBSYSTEM (Write FIFO) Failed'
5633 065536      127     122     111   T206SSR:.ASCIZ  'WRITE SUBSYSTEM (Read FIFO) Failed'
5634 065601      127     122     111   T208SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Format) Failed'
5635 065647      122     145     141   T20SWP: .ASCIZ  'Read Data Parity Error (PARERR) Failed to Set after Write Wrong Parity'
5636 065756      122     145     141   T20RSF: .ASCIZ  'Read Data Parity Error (PARERR) Failed to Clear after RESET FIFO'
5637 066057      122     145     141   T20CWP: .ASCIZ  'Read Data Parity Error (PARERR) occurred in Data Loopback'
5638                                                   .EVEN
5639
5640                                           ;+
5641                                           ; CLEAR MESSAGE BUFFER
5642                                           ;-
5643 066152                                    T20CLRBUF:
5644 066152                                            SAVREG                  ;
5645 066156  012701  066452                            MOV     #T20BFR,R1      ;SAVE R1-R5 UNTIL NEXT RETURN
5646 066162  012702  000120                            MOV     #T20BEND-T20BFR,R2 ;GET MESSAGE BUFFER ADDRESS
5647 066166  105021                            10$:    CLRB    (R1)+           ;SIZE OF MESSAGE BUFFER IN BYTES
5648 066170  005302                                    DEC     R2              ;CLEAR A BYTE
5649 066172  003375                                    BGT     10$             ;DONE?
5650 066174  000207                                    RTS     PC              ;BR IF NO
5651                                                                           ;RETURN
5652
5653                                           ;+
5654                                           ; SETUP T20PK2 PACKET FOR READ STATUS
5655 066176                                    ;-
5656 066176  004737  066152                    T20SRD:
5657 066202  012700  066610                            JSR     PC,T20CLRBUF    ;CLEAR MESSAGE BUFFER
5658 066206  112720  000005                            MOV     #T20DT2,R0      ;WRITE SUBSYSTEM DATA BUFFER
5659 066212  105010                                    MOVB    #PW.RDSTATUS,(R0)+ ;STORE READ STATUS COMMAND IN BSEL0
5660 066214  000207                                    CLRB    (R0)            ;CLEAR BSEL1
5661                                                   RTS     PC              ;RETURN
5662
5663
5664                                           ;+
5665                                           ; SETUP T20PK2 PACKET FOR WRITE NPR
5666                                           ;
5667                                           ; INPUT:
                                               ;       R0 CONTAINS BSEL1 NPR DATA
```

```
5668                                    ;
5669                                    ;-
5670 066216                             T20WNPR:
5671 066216  004737  066152                    JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
5672 066222  012701  066610                    MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5673 066226  112721  000011                    MOVB    #PW.WNPR,(R1)+          ;STORE WRITE NPR IN BSEL0
5674 066232  110011                            MOVB    R0,(R1)                 ;STORE NPR DATA IN BSEL1
5675 066234  000207                            RTS     PC                      ;RETURN
5676
5677                                    ;+
5678                                    ; SETUP T20PK2 PACKET FOR READ FIFO
5679                                    ;
5680                                    ; INPUT:
5681                                    ;       R0 CONTAINS SEL2 BYTE COUNT
5682                                    ;-
5683 066236                             T20RFIF:
5684 066236  004737  066152                    JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
5685 066242  012701  066610                    MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5686 066246  112721  000003                    MOVB    #PW.RFIFO,(R1)+         ;STORE READ FIFO IN BSEL0
5687 066252  110021                            MOVB    R0,(R1)+                ;STORE BYTE COUNT IN BSEL1
5688 066254  000207                            RTS     PC                      ;RETURN
5689                                    ;+
5690                                    ; SETUP T20PK2 PACKET FOR WRITE FIFO
5691                                    ;
5692                                    ; INPUT:
5693                                    ;       R0 CONTAINS BYTE COUNT
5694                                    ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
5695                                    ;-
5696 066256                             T20WFIF:
5697 066256                                    SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
5698 066262  004737  066152                    JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
5699 066266  012702  066610                    MOV     #T20DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
5700 066272  112722  000004                    MOVB    #PW.WFIFO,(R2)+         ;STORE WRITE FIFO IN BSEL0
5701 066276  110022                            MOVB    R0,(R2)+                ;STORE BYTE COUNT IN BSEL1
5702 066300  005022                            CLR     (R2)+                   ;CLEAR SEL2 (UNUSED)
5703 066302  112122                     10$:   MOVB    (R1)+,(R2)+             ;STORE DATA PATTERN BYTE
5704 066304  005300                            DEC     R0                      ;DONE ALL BYTES?
5705 066306  003375                            BGT     10$                     ;BR IF NO
5706 066310  000207                            RTS     PC                      ;RETURN
5707
5708                                    ;+
5709                                    ; SETUP T20PK2 FOR WRITE FORMAT TRANSPORT REGISTER
5710                                    ;
5711                                    ; INPUT:
5712                                    ;       R0 CONTAINS DRIVING DATA PATTERN
5713                                    ;-
5714 066312                             T20WFMT:
5715 066312  004737  066152                    JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
5716 066316  012701  066610                    MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5717 066322  112721  000007                    MOVB    #PW.WFMT,(R1)+          ;STORE WRITE FORMAT IN BSEL0
5718 066326  110021                            MOVB    R0,(R1)+                ;STORE DATA WORD IN BSEL1
5719 066330  000207                            RTS     PC                      ;RETURN
5720                                    ;+
5721                                    ; SETUP T20PK2 PACKET FOR WRITE MISC.
5722                                    ;
5723                                    ;       R0 CONTAINS WRITE MISC DATA
5724                                    ;-
```

```
5725 066332
5726 066332  012701  066610          T20WMISC:
5727 066336  112721  000010                  MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5728 066342  110011                          MOVB    #PW.WMISC,(R1)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
5729 066344  000207                          MOVB    R0,(R1)                 ;STORE INVERT EXTENDED FEATURES IN BSEL1
5730                                         RTS     PC                      ;RETURN
5731                                 ;+
5732                                 ; SETUP T20PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
5733 066346                          ;-
5734 066346  012700  066610          T20SEXT:
5735 066352  112720  000010                  MOV     #T20DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
5736 066356  112710  000200                  MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
5737 066362  000207                          MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
5738                                         RTS     PC                      ;RETURN
5739                                 ;+
5740                                 ; CLEAR EXPECTED DATA MESSAGE BUFFER
5741 066364                          ;-
5742 066364  012701  065112          T20CLEXP:
5743 066370  012700  000120                  MOV     #T20EXP,R1              ;GET EXPD ADDRESS
5744 066374  105021                          MOV     #T20EXEND-T20EXP,R0     ;GET EXPD SIZE
5745 066376  005300          10$:    CLRB    (R1)+                   ;CLEAR A BYTE
5746 066400  003375                  DEC     R0                      ;DONE?
5747 066402  000207                  BGT     10$                     ;BR IF NO
5748                                 RTS     PC                      ;RETURN
5749
5750                                 ;+
5751                                 ;Set WORDS 0-7 of expd message BUFFER = to recv since not testing
5752 066404                          ;-
5753 066404  012702  065112          T20SETEXP:
5754 066410  012703  066452                  MOV     #T20EXP,R2              ;GET EXPD
5755 066414  012700  000010                  MOV     #T20BFR,R3              ;GET READ STATUS RECV BUFFER
5756 066420  012322                          MOV     #8.,R0                  ;SET WORDS 0-7 EXP=RECV
5757 066422  005300          5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
5758 066424  003375                  DEC     R0                      ;DONE WORDS 0-7 WORDS?
5759 066426  000207                  BGT     5$                      ;BR IF NO
5760                                 RTS     PC                      ;RETURN
5761
5762
5766
5767                                 ;
5768                                 ;WRITE CHARACTERISTICS COMMAND PACKET
5769 066430                          ;
5770 066430  100004          T20PACKET:                              ;COMMAND PACKET FOR TEST
5771 066432  066440                  .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
5772 066434  000000                  .WORD   T20DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
5773 066436  000012                  .WORD   0
5774                                 .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
5775 066440
5776 066440  066452          T20DATA:                                ;CHARACTERISTICS DATA BLOCK
5777 066442  000000                  .WORD   T20BFR                  ;ADDRESS OF MESSAGE BUFFER
5778 066444  000024                  .WORD   0
5779 066446  000000                  .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
5780 066450  000007                  .WORD   0                       ;ESS,ENB,EAI,ERI
5781                                 .WORD   7                       ;EXTENDED FEATURES UNIT NO.
5782
5783
5784                                 ;MESSAGE BUFFER FOR ALL TEST 17 COMMANDS
```

TEST 9: READ/WRITE DATA PARITY TEST

```
5785 066452                    T20BFR:                         ;BEGIN MESSAGE BUFFER
5786 066452  000000                    .WORD   0               ;MESSAGE TYPE
5787 066454  000000                    .WORD   0               ;DATA FIELD LENGTH
5788 066456  000000                    .WORD   0               ;RBPCR
5789 066460  000000                    .WORD   0               ;XST0
5790 066462  000000                    .WORD   0               ;XST1
5791 066464  000000                    .WORD   0               ;XST2
5792 066466  000000                    .WORD   0               ;XST3
5793 066470  000000                    .WORD   0               ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
5794 066472                    T20BFSTA: .BLKB 64.               ;READ STATUS AND WRITE FIFO BUFFER
5795 066572                    T20BEND:                        ;END OF MESSAGE BUFFER
5796                           ;
5797                           ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
5798                           ;
5800         066600                    .=<.+10>&177770
5802 066600                    T20PK2:
5803 066600  100006                    .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
5804 066602  066610                    .WORD   T20DT2                  ;LOW ADDRESS OF DATA BLOCK
5805 066604  000000                    .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
5806 066606  000012                    .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
5807
5808 066610                    T20DT2:                         ;DATA BLOCK
5809 066610     000                    .BYTE   0               ;BSEL0
5810 066611     000                    .BYTE   0               ;BSEL1
5811 066612  000000                    .WORD   0               ;SEL2
5812 066614                            .BLKB   64.             ;WRITE FIFO DATA OUTPUT BUFFER
5813
5814
5815 066714                    L10073: ENDTST
     066714                                    TRAP    C$ETST
     066714  104401
5816                                   .SBTTL  TEST 10: MANUAL INTERVENTION
5817
5818                           ;THE MANUAL INTERVENTION TEST IS A STANDALONE ROUTINE (NOT REALLY A "TEST")
5819                           ;THAT ALLOWS THE OPERATOR TO CHECK OUT VARIOUS ELEMENTS AND FUNCTIONS OF
5820                           ;THE SUBSYSTEM THAT CANNOT BE MANIPULATED BY THE PROGRAM ALONE. WHEN
5821                           ;THIS ROUTINE IS STARTED, IT FIRST PRINTS OUT A MENU OF SELECTABLE
5822                           ;SUBTESTS AND THEN WAITS FOR THE OPERATOR TO TYPE IN A SELECTION CODE.
5823                           ;THE ONLY WAYS TO EXIT THIS ROUTINE AND RETURN TO THE DIAGNOSTIC SUPERVISOR
5824                           ;ARE BY TYPING <CTRL-C> OR SELECTING CODE 7.
5825                           ;SELECTION CODES AND SUBROUTINES ARE:
5926                           ;
5827                           ;
5828                           ;       CODE    ROUTINE
5829                           ;
5830                           ;         0     HELP. PRINTS THIS MENU.
5831                           ;         1     TURN ON ALL M7196 LED INDICATORS
5832                           ;         2     TURN OFF ALL M7196 LED INDICATORS
5833                           ;         3     OFFLINE/ONLINE ATTENTION TEST
5834                           ;         4     WRITE-PROTECT TEST
5835                           ;         5     INITIATE TRANSPORT SERVO EXERCISER
5836                           ;         6     PRINT EXTENDED TRANSPORT STATUS
5837                           ;         7     EXIT (RETURN TO SUPERVISOR)
5838                           ;
5839                           ;
5840                           ;EACH MENU ITEM CORRESPONDS TO A SUBTEST, AS FOLLOWS:
5841                           ;
```

```
5842                                    ;
5843                                    ;
5844                                    ;PRINTS OUT THE MENU ON THE CONSOLE TERMINAL.
5845                                    ;
5846                                    ;
5847                                    ;CAUSES ALL THREE LED INDICATORS ON THE M7196 MODULE
5848                                    ;TO BE ILLUMINATED. AFTER INITIATING THIS ROUTINE, THE OPERATOR
5849                                    ;SHOULD OBSERVE THE LED'S AND VERIFY THAT THEY ARE INDEED ALL LIT.
5850                                    ;THIS ROUTINE FIRST USES THE WRITE SUBSYSTEM MEMORY COMMAND TO
5851                                    ;SET THE FORCE WRONG PARITY FLIP-FLOP, WHICH SERVES TO DRIVE THE
5852                                    ;"PROCESSOR NOT OK" LED. THEN IT ENTERS A LOOP THAT CONTINUALLY
5853                                    ;WRITES THE LOW BYTE OF TSDB AND READS THE TSSR. THESE LATTER TWO
5854                                    ;OPERATIONS WILL CAUSE THE "NOT SSR" AND "DRIVING BUS" LED'S TO
5855                                    ;GLOW -- THEY ARE NOT REALLY LIT AT ALL TIMES BUT SHOULD APPEAR
5856                                    ;REASONABLY VISIBLE.
5857                                    ;
5858                                    ;
5859                                    ;INITIALIZES THE CONTROLLER TO CAUSE ALL LED'S TO
5860                                    ;EXTINGUISH.
5861                                    ;
5862                                    ;
5863                                    ;
5864                                    ;
5865                                    ;THIS ROUTINE INITIALIZES THE CONTROLLER, ISSUES A
5866                                    ;WRITE CHARACTERISTICS COMMAND TO ENABLE ATTENTION INTERRUPTS,
5867                                    ;ISSUES A MESSAGE BUFFER RELEASE COMMAND, PRINTS A MESSAGE ON THE
5868                                    ;CONSOLE TERMINAL INSTRUCTING THE OPERATOR TO TOGGLE THE ON-LINE
5869                                    ;SWITCH ON THE TRANSPORT, THEN WAITS FOR AN ATTENTION INTERRUPT.
5870                                    ;EACH TIME THE TRANSPORT TRANSITIONS FROM ON-LINE TO OFF-LINE OR
5871                                    ;VICE-VERSA, AN ATTENTION INTERRUPT SHOULD BE GENERATED. THE PROGRAM
5872                                    ;WILL REPORT THE INTERRUPT AND THE CURRENT STATE OF THE TRANSPORT.
5873                                    ;THE OPERATOR SHOULD VERIFY THAT THE REPORTED STATE MATCHES THE
5874                                    ;STATE INDICATED BY THE LED ON THE FRONT PANEL OF THE TRANSPORT.
5875                                    ;IN ADDITION, WHEN THE TRANSPORT IS PLACED OFF-LINE, THE PROGRAM
5876                                    ;ISSUES A SEQUENCE OF TAPE-MOTION COMMANDS (READ, WRITE, POSITION, ETC.
5877                                    ;AND VERIFIES THAT, FOR EACH COMMAND, FUNCTION REJECT TERMINATION
5878                                    ;RESULTS, ALONG WITH THE NON-EXECUTABLE FUNCTION (NEF) ERROR BIT BEING
5879                                    ;SET.
5880                                    ;
5881                                    ;THIS ROUTINE INSTRUCTS THE OPERATOR TO MOUNT A SCRATCH
5882                                    ;TAPE REEL THAT DOES NOT HAVE A WRITE-ENABLE RING INSTALLED, THEN
5883                                    ;WAITS FOR THE OPERATOR TO RESPOND THAT THIS HAS BEEN ACCOMPLISHED.
5884                                    ;UPON THE RESPONSE, THE PROGRAM VERIFIES THAT THE TRANSPORT SHOWS
5885                                    ;A WRITE-PROTECTED STATUS, THEN ATTEMPTS TO WRITE DATA ON THE
5886                                    ;TAPE AND EXPECTS THE APPROPRIATE ERROR TERMINATION INDICATING THAT
5887                                    ;THE WRITE FUNCTION COULD NOT BE PERFORMED BECAUSE THE REEL IS
5888                                    ;WRITE-PROTECTED.  IF THE APPROPRIATE TERMINATION IS NOT RECEIVED,
5889                                    ;AN ERROR IS REPORTED.
5890                                    ;
5891                                    ;
5892                                    ;
5893                                    ;INSTRUCTS THE OPERATOR TO PLACE THE TAPE TRANSPORT(S)
5894                                    ;ON-LINE (IF ANY ARE OFF-LINE) THEN ATTEMPTS TO PERFORM AN EXTENDED
5895                                    ;STATUS READOUT.  FOR EACH TRANSPORT EQUIPPED WITH THIS FEATURE,
5896                                    ;THE PROGRAM FORMATS AND PRINTS OUT THE RESULTING STATUS. IF THE
5897                                    ;TRANSPORT IS NOT EQUIPPED WITH THIS FEATURE, A MESSAGE INDICATING
5898                                    ;SUCH IS ISSUED.
```

TEST 10: MANUAL INTERVENTION

```
5899                                    ;
5900                                    ;
5901                                    ;
5902                                    ;
5903 066716                             BGNTST
     066716                                                          T10::
5908 066716                             RFLAGS  R0              ;GET OPERATOR FLAGS
     066716  104421                                                     TRAP    C$RFLA
5909 066720  001403                     BEQ     21$             ;BR, IF OK TO RUN
5910 066722  012700  072300             MOV     #T38NE,R0       ;"TEST NOT EXECUTED"
5911 066726  000402                     BR      3$              ;JUMP IF NOT FIRST TEST
5912 066730                         21$:
5913 066730  012700  073415             MOV     #T38ID,R0       ;TEST ID MESSAGE
5914 066734  004737  016510         3$: JSR     PC,TSTSETUP     ;DO THE COMMON SETUP
5915 066740  004737  020500             JSR     PC,CHKMAN       ;IS MANUAL INTERVENTION ALLOWED?
5916 066744  103402                     BCS     22$             ;BR, IF MANUAL INTER ALLOWED
5917 066746  000137  071500             JMP     64$             ;JUMP IF NOT ALLOWED
5918 066752                         22$:
5922 066752  005037  002222         2$: CLR     FATFLG          ;CLEAR THE FATAL ERROR FLAG
5923 066756  012737  176750  071512     MOV     #65000.,T38DLY  ;SET UP DELAY COUNTER
5924 066764  004737  015774         5$: JSR     PC,SOFINIT      ;DO A SOFT INIT
5925 066770  103427                     BCS     23$             ;BRANCH IF OK
5926 066772  010001                     MOV     R0,R1           ;CONTENTS OF TSSR REGISTER
5927 066774  032701  000200             BIT     #SSR,R1         ;CHECK FOR TSSR SET
5928 067000  001023                     BNE     23$             ;KEEP GOING IF NOT SET
5929 067002                             DELAY   250             ;CALL DELAY ROUTINE
     067002  012727  000250                                             MOV     #250,(PC)+
     067006  000000                                                     .WORD   0
     067010  013727  002116                                             MOV     L$DLY,(PC)+
     067014  000000                                                     .WORD   0
     067016  005367  177772                                             DEC     -6(PC)
     067022  001375                                                     BNE     .-4
     067024  005367  177756                                             DEC     -22(PC)
     067030  001367                                                     BNE     .-20
5930 067032  005337  071512             DEC     T38DLY          ;BUMP COUNTER DOWN
5931 067036  001352                     BNE     5$              ;BR, IF MORE TIME LEFT
5932 067040                             ERRDF   ERRNO,SFIERR,SFIMSG     ;REPORT FATAL ERROR
     067040  104455                                                     TRAP    C$ERDF
     067042  001751                                                     .WORD   1001
     067044  003652                                                     .WORD   SFIERR
     067046  012034                                                     .WORD   SFIMSG
5933 067050  012700  073442         23$: MOV    #MIMENU,R0      ;MENU OF MANUAL INTERVENTIONS
5934 067054  012701  000006             MOV     #6,R1           ;MAXIMUM ALLOWED SELECTION
5935 067060  004737  020256             JSR     PC,GETSEL       ;GO GET THE OPERATORS SELECTION
5936 067064  010004                     MOV     R0,R4           ;GET NUMBER FROM ROUTINE
5937 067066  006304                     ASL     R4              ;CONVERT TO WORD OFFSET
5938 067070  000174  067074             JMP     @6$(R4)         ;JUMP TO PROPER LOOP
5939 067074  066752         6$: .WORD   2$              ;RETYPE THE MENU
5940 067076  067112             .WORD   10$             ; 1 TURN ON LED'S
5941 067100  067374             .WORD   15$             ; 2 TURN OFF LED'S
5942 067102  067626             .WORD   20$             ; 3 ONLINE ATTENTION
5943 067104  070262             .WORD   25$             ; 4 WRITE PROTECT
5944 067106  071216             .WORD   35$             ; 5 EXTENDED TRANSPORT STATUS
5945 067110  071474             .WORD   63$             ; 6 LEAVE THE TEST
5946 067112                     10$: PRINTF #T38MS2             ;TELL OPERATOR TO CNTRL-C FOR EXIT
     067112  012746  073311                                             MOV     #T38MS2,-(SP)
     067116  012746  000001                                             MOV     #1,-(SP)
```

```
                067122  010600                                                          MOV     SP,R0
                067124  104417                                                          TRAP    C$PNTF
                067126  062706  000004                                                  ADD     #4,SP
5947            067132  004737  074046          JSR     PC,T38REST      ;SET PACKET TO INITIAL VALUES
5948            067136  004737  015774          JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
5949            067142  103405                  BCS     100$            ;BR IF SOFT INIT = OK
5953            067144  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
5954            067146                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
                067146  104455                                                          TRAP    C$ERDF
                067150  001752                                                          .WORD   1002
                067152  003652                                                          .WORD   SFIERR
                067154  012034                                                          .WORD   SFIMSG
5955            067156  013737  002202  072240  100$:   MOV     UNITN,T38DSW    ;SET UNIT NUMBER
5956
5957            067164  012704  072220          MOV     #T38PK2,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
5958            067170  004737  010662          JSR     PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
5959            067174  103405                  BCS     110$            ;BR, IF COMMAND ISSUED OK
5963            067176  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
5964            067200                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
                067200  104456                                                          TRAP    C$ERHRD
                067202  001753                                                          .WORD   1003
                067204  005056                                                          .WORD   WRTMSG
                067206  012034                                                          .WORD   SFIMSG
5965            067210                  110$:
5966            067210  112737  000000  071531          MOVB    #0,T38BS1       ;CLEAR BIT #4
5967            067216  112737  000011  071530          MOVB    #11,T38BS0      ;WRITE MISC COMMAND
5968            067224  012704  071520                  MOV     #T38PACKET,R4   ;SET UP NEW WRT. SUBSYS MEM. COMMAND
5969                                            ;
5970                                            ;NOTE: THIS COMMAND TURNS ON THE PROCESSOR FAIL LED
5971                                            ;
5972            067230  010465  000000          MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
5973            067234  004737  016336          JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
5974            067240  103405                  BCS     150$            ;BR IF CARRY SET (GOOD RETURN)
5975            067242  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
5979            067244                          ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                067244  104455                                                          TRAP    C$ERDF
                067246  001754                                                          .WORD   1004
                067250  072716                                                          .WORD   T38SSR
                067252  012046                                                          .WORD   PKTSSR
5980            067254  104406          150$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
                                                                                        TRAP    C$CLP1
5981            067256                          SETPRI  #PRI06          ;RAISE THE PRIORITY
                067256  012700  000300                                                  MOV     #PRI06,R0
                067262  104441                                                          TRAP    C$SPRI
5982            067264  005037  071504          CLR     TTION2          ;ASSUME INTERRUPTS ARE ENABLED
5983            067270  032737  000100  177560  BIT     #100,@#TTICSR   ;ARE TTI INTERRUPTS ON ?
5984            067276  001005                  BNE     701$            ;BRANCH IF YES
5985            067300  005237  071504          INC     TTION2          ;FLAG SET IF INTERRUPTS OFF
5986            067304  052737  000100  177560  BIS     #100,@#TTICSR   ;ENABLE INTERRUPTS
5987            067312  012701  000060  701$:   MOV     #TTIVEC,R1      ;START OF TTI VECTORS
5988            067316  011137  071506          MOV     (R1),TVSAV2     ;SAVE THE CURRENT TTI VECTOR
5989            067322  012721  071000          MOV     #590$,(R1)+     ;SET NEW INTERRUPT ROUTINE
5990            067326  011137  071510          MOV     (R1),TPSAV2     ;SAVE THE VECTOR PRIORITY
5991            067332  012711  000300          MOV     #PRI06,(R1)     ;USE PRIORITY SIX
5992            067336                          SETPRI  #PRI00          ;LOWER INTERRUPT BR LEVEL
                067336  012700  000000                                                  MOV     #PRI00,R0
                067342  104441                                                          TRAP    C$SPRI
```

```
 5993 067344  012701  177777            MOV     #-1,R1                  ;DATA TO WRITE TO TSDB
 5994 067350  000240              12$:   NOP                             ;ALLOW OPERATOR TO TYPE ^C
 5995 067352  012702  001750            MOV     #1000.,R2               ;SET-UP INNER LOOP
 5996 067356  110165  000000      14$:   MOVB    R1,TSDB(R5)             ;WRITE DATA TO TSDB
 5997 067362  016500  000002            MOV     TSSR(R5),R0             ;READ TSSR
 5998 067366  005302                    DEC     R2                      ;REDUCE INNER COUNT
 5999 067370  001372                    BNE     14$                     ;LOOP TILL EXPIRES
 6000 067372  000766                    BR      12$                     ;LOOP UNTIL HALTED
 6001
 6002 067374                      15$:   PRINTF  #T38MS2                 ;TYPE CNTL C TO EXIT
      067374  012746  073311                                                    MOV     #T38MS2,-(SP)
      067400  012746  000001                                                    MOV     #1,-(SP)
      067404  010600                                                            MOV     SP,R0
      067406  104417                                                            TRAP    C$PNTF
      067410  062706  000004                                                    ADD     #4,SP
 6003 067414  004737  015774            JSR     PC,$OFINIT              ;DO SOFT INIT OF CONTROLLER
 6004 067420  103405                    BCS     200$                    ;BR IF SOFT INIT = OK
 6008 067422  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6009 067424                            ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      067424  104455                                                            TRAP    C$ERDF
      067426  001755                                                            .WORD   1005
      067430  003652                                                            .WORD   SFIERR
      067432  012034                                                            .WORD   SFIMSG
 6010 067434                      200$:
 6011 067434  013737  002202  072240        MOV     UNITN,T38DSW            ;SET UNIT NUMBER
 6012 067442  012704  072220            MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
 6013 067446  004737  010662            JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 6014 067452  103405                    BCS     210$                    ;BR, IF COMMAND ISSUED OK
 6018 067454  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6019 067456                            ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      067456  104456                                                            TRAP    C$ERHRD
      067460  001756                                                            .WORD   1006
      067462  005056                                                            .WORD   WRTMSG
      067464  012034                                                            .WORD   SFIMSG
 6020                             ;*****************************************************************
 6021                             ;
 6022                             ;       THIS WRITE SUB-SYSTEM MEMORY COMMAND JUST HOLDS THE LEDS OFF
 6023                             ;
 6024                             ;*****************************************************************
 6025 067466                      210$:
 6026 067466  112737  000000  071531        MOVB    #0,T38S1                ;CLEAR BIT #4
 6027 067474  112737  000025  071530        MOVB    #25,T38S0               ;STOP DRIVE TEST 22
 6028 067502  012704  071520            MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
 6029 067506  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 6030 067512  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 6031 067516  103405                    BCS     250$                    ;BR IF CARRY SET (GOOD RETURN)
 6032 067520  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6036 067522                            ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      067522  104455                                                            TRAP    C$ERDF
      067524  001757                                                            .WORD   1007
      067526  072716                                                            .WORD   T38SSR
      067530  012046                                                            .WORD   PKTSSR
 6037 067532                      250$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      067532  104406                                                            TRAP    C$CLP1
 6038 067534                            SETPRI  #PRI06                  ;RAISE THE PRIORITY
      067534  012700  000300                                                    MOV     #PRI06,R0
      067540  104441                                                            TRAP    C$SPRI
```

TEST 10: MANUAL INTERVENTION SEQ 0222

```
6039 067542  005037  071504                CLR     TTION2          ;ASSUME INTERRUPTS ARE ENABLED
6040 067546  032737  000100  177560        BIT     #100,@#TTICSR   ;ARE TTI INTERRUPTS ON ?
6041 067554  001005                        BNE     710$            ;BRANCH IF YES
6042 067556  005237  071504                INC     TTION2          ;FLAG SET IF INTERRUPTS OFF
6043 067562  052737  000100  177560        BIS     #100,@#TTICSR   ;ENABLE INTERRUPTS
6044 067570  012701  000060          710$: MOV     #TTIVEC,R1      ;START OF TTI VECTORS
6045 067574  011137  071506                MOV     (R1),TVSAV2     ;SAVE THE CURRENT TTI VECTOR
6046 067600  012721  071000                MOV     #590$,(R1)+     ;SET NEW INTERRUPT ROUTINE
6047 067604  011137  071510                MOV     (R1),TPSAV2     ;SAVE THE VECTOR PRIORITY
6048 067610  012711  000300                MOV     #PRI06,(R1)     ;USE PRIORITY SIX
6049 067614                                SETPRI  #PRI00          ;LOWER INTERRUPT BR LEVEL
     067614  012700  000000                                            MOV     #PRI00,R0
     067620  104441                                                    TRAP    C$SPRI
6050 067622  000240          260$: NOP                               ;ALLOW CNTL C
6051 067624  000776                BR      260$                    ;LOOP UNTIL STOPPED
6052
6053
6054 067626                        20$:    PRINTF  #T38MS2         ;TELL'EM WHAT TO TYPE
     067626  012746  073311                                            MOV     #T38MS2,-(SP)
     067632  012746  000001                                            MOV     #1,-(SP)
     067636  010600                                                    MOV     SP,R0
     067640  104417                                                    TRAP    C$PNTF
     067642  062706  000004                                            ADD     #4,SP
6055 067646                                SETPRI  #PRI00          ;LOWER PRIORITY TO ALLOW INTERRUPTS
     067646  012700  000000                                            MOV     #PRI00,R0
     067652  104441                                                    TRAP    C$SPRI
6056 067654  005037  002224                CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
6057 067660  004737  015774                JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
6058 067664  103405                        BCS     300$            ;BR IF SOFT INIT = OK
6062 067666  010001                        MOV     R0,R1           ;SAVE CONTENTS OF TSSR
6063 067670                                ERRDF   ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
     067670  104455                                                    TRAP    C$ERDF
     067672  001760                                                    .WORD   1008
     067674  003652                                                    .WORD   SFIERR
     067676  012034                                                    .WORD   SFIMSG
6064 067700                        300$:
6065 067700  013737  002202  072240        MOV     UNITN,T38DSW    ;SET UNIT NUMBER IN PACKET
6066 067706  012737  000040  072236        MOV     #BIT5,T38EAI    ;ENABLE ATTENTION INTERRUPTS
6067 067714  012704  072220                MOV     #T38PK2,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
6068 067720  004737  010662                JSR     PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
6069 067724  103405                        BCS     310$            ;BR, IF COMMAND ISSUED OK
6073 067726  010001                        MOV     R0,R1           ;SAVE CONTENTS OF TSSR
6074 067730                                ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED
     067730  104456                                                    TRAP    C$ERHRD
     067732  001761                                                    .WORD   1009
     067734  005056                                                    .WORD   WRTMSG
     067736  012034                                                    .WORD   SFIMSG
6075 067740                        310$:
6076 067740  012704  072250                MOV     #T38PK3,R4      ;SET UP NEW PACKET FOR MESS BUF REL
6077 067744  010465  000000                MOV     R4,TSDB(R5)     ;MESSAGE BUFFER RELEASE,ACK,CVC=1 CMD
6078 067750  004737  016250                JSR     PC,WAITF        ;WAIT FOR SSR TO SET
6079 067754  005002                        CLR     R2              ;MAKE SURE ALL IS CLEAR
6080 067756  016501  000002                MOV     TSSR(R5),R1     ;GET TSSR STATUS
6081 067762  032701  000100                BIT     #OFL,R1         ;IS OFL SET
6082 067766  001402                        BEQ     320$            ;BR, IF OFL IS NOT SET
6083 067770  052702  000100                BIS     #OFL,R2         ;SET OFL IN EXPECTED
6084 067774  052702  000200          320$: BIS     #SSR,R2         ;SET UP EXPECTED
```

```
6085 070000  020201                    CMP     R2,R1                   ;IS EVERYTHING OK
6086 070002  001404                    BEQ     350$                    ;BR, IF ALL IS WELL
6090 070004                            ERRHRD  ERRNO,T38SST,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     070004  104456                                                            TRAP    C$ERHRD
     070006  001762                                                            .WORD   1010
     070010  073126                                                            .WORD   T38SST
     070012  012046                                                            .WORD   PKTSSR
6091 070014                     350$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     070014  104406                                                            TRAP    C$CLP1
6092 070016                            PRINTF  #T38MS1                 ;TELL OPERATOR TO TOGGLE SWITCH
     070016  012746  073216                                                    MOV     #T38MS1,-(SP)
     070022  012746  000001                                                    MOV     #1,-(SP)
     070026  010600                                                            MOV     SP,R0
     070030  104417                                                            TRAP    C$PNTF
     070032  062706  000004                                                    ADD     #4,SP
6093 070036                            PRINTF  #T38MS2                 ;TELL OPERATOR TO DO ^C TO EXIT
     070036  012746  073311                                                    MOV     #T38MS2,-(SP)
     070042  012746  000001                                                    MOV     #1,-(SP)
     070046  010600                                                            MOV     SP,R0
     070050  104417                                                            TRAP    C$PNTF
     070052  062706  000004                                                    ADD     #4,SP
6094 070056                            SETPRI  #PRI06                  ;RAISE THE PRIORITY
     070056  012700  000300                                                    MOV     #PRI06,R0
     070062  104441                                                            TRAP    C$SPRI
6095 070064  005037  071504            CLR     TTION2                  ;ASSUME INTERRUPTS ARE ENABLED
6096 070070  032737  000100  177560    BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
6097 070076  001005                    BNE     720$                    ;BRANCH IF YES
6098 070100  005237  071504            INC     TTION2                  ;FLAG SET IF INTERRUPTS OFF
6099 070104  052737  000100  177560    BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
6100 070112  012701  000060     720$:  MOV     #TTIVEC,R1              ;START OF TTI VECTORS
6101 070116  011137  071506            MOV     (R1),TVSAV2             ;SAVE THE CURRENT TTI VECTOR
6102 070122  012721  071000            MOV     #590$,(R1)+             ;SET NEW INTERRUPT ROUTINE
6103 070126  011137  071510            MOV     (R1),TPSAV2             ;SAVE THE VECTOR PRIORITY
6104 070132  012711  000300            MOV     #PRI06,(R1)             ;USE PRIORITY SIX
6105 070136                            SETPRI  #PRI00                  ;LOWER INTERRUPT BR LEVEL
     070136  012700  000000                                                    MOV     #PRI00,R0
     070142  104441                                                            TRAP    C$SPRI
6106 070144  000240             360$:  NOP                             ;ALLOW CONTROL C
6107 070146  005737  002224            TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
6108 070152  001001                    BNE     370$                    ;BRANCH IF YES
6109 070154  000773                    BR      360$                    ;WAIT SOME MORE FOR INTERRUPT
6110 070156                     370$:  PRINTF  #T38INT                 ;"INTERRUPT RECEIVED"
     070156  012746  073006                                                    MOV     #T38INT,-(SP)
     070162  012746  000001                                                    MOV     #1,-(SP)
     070166  010600                                                            MOV     SP,R0
     070170  104417                                                            TRAP    C$PNTF
     070172  062706  000004                                                    ADD     #4,SP
6111 070176  016501  000002            MOV     TSSR(R5),R1             ;READ TSSR STATUS
6112 070202  032701  000100            BIT     #OFL,R1                 ;CHECK THE OFF-LINE BIT
6113 070206  001011                    BNE     380$                    ;BR, IF DRIVE IS OFF-LINE
6114 070210                            PRINTF  #T38ONL                 ;"DRIVE IS NOW ON-LINE"
     070210  012746  073036                                                    MOV     #T38ONL,-(SP)
     070214  012746  000001                                                    MOV     #1,-(SP)
     070220  010600                                                            MOV     SP,R0
     070222  104417                                                            TRAP    C$PNTF
     070224  062706  000004                                                    ADD     #4,SP
6115 070230  000410                    BR      390$                    ;ALMOST DONE
```

```
6116 070232
     070232  012746  073072          380$:   PRINTF  @T380FL                 ;"DRIVE IS NOW OFF-LINE"
     070236  012746  000001                                                          MOV     @T380FL,-(SP)
     070242  010600                                                                 MOV     #1,-(SP)
     070244  104417                                                                 MOV     SP,R0
     070246  062706  000004                                                         TRAP    C$PNTF
6117 070252  005037  002224          390$:   CLR     INTRECV                        ADD     #4,SP
6118 070256  000137  067700                  JMP     300$                    ;CLEAR INTERRUPT FLAG
6119 070262                          25$:    GMANIL  T38MSG,T38DAT,-1,NO     ;TRY AGAIN
     070262  104443                                                         ;WAIT FOR OPERATOR TO MOUNT TAPE
     070264  000404                                                                 TRAP    C$GMAN
     070266  074044                                                                 BR      10000$
     070270  000120                                                                 .WORD   T38DAT
     070272  073355                                                                 .WORD   T$CODE
     070274  177777                                                                 .WORD   T38MSG
     070276                                                                         .WORD   -1
6120 070276                                  BNCOMPLETE 25$                10000$:
     070276  103371                                                         ;RETRY IF ERROR
6121 070300  005737  074044                  TST     T38DAT                         BCC     25$
6122 070304  001002                          BNE     27$                     ;DID OPERATOR SAY 'YES' ?
6123 070306  000137  066752                  JMP     2$                      ;BRANCH IF YES
6124 070312                          27$:                                    ;RETURN TO MAIN MENU
6125 070312  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
6126 070316  103405                          BCS     400$                    ;BR IF SOFT INIT = OK
6130 070320  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6131 070322                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     070322  104455                                                                 TRAP    C$ERDF
     070324  001763                                                                 .WORD   1011
     070326  003652                                                                 .WORD   SFIERR
     070330  012034                                                                 .WORD   SFIMSG
6132 070332                          400$:   CKLOOP                          ;LOOP IF SELECTED
     070332  104406                                                                 TRAP    C$CLP1
6133 070334  013737  002202  072240          MOV     UNITN,T38DSW            ;SET UNIT NUMBER
6134 070342  012704  072220                  MOV     @T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
6135 070346  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
6136 070352  103405                          BCS     410$                    ;BR, IF COMMAND ISSUED OK
6140 070354  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6141 070356                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     070356  104456                                                                 TRAP    C$ERHRD
     070360  001764                                                                 .WORD   1012
     070362  005056                                                                 .WORD   WRTMSG
     070364  012034                                                                 .WORD   SFIMSG
6142 070366                          410$:   CKLOOP                          ;LOOP IF SELECTED
     070366  104406                                                                 TRAP    C$CLP1
6143 070370  013701  071544                  MOV     T38BFR+6,R1             ;PICK UP XSTO CONTENTS
6144 070374  010102                          MOV     R1,R2                   ;SET UP EXPECTED
6145 070376  052702  000004                  BIS     @BIT2,R2                ;SET UP THE WRITE LOCKED BIT
6146 070402  020102                          CMP     R1,R2                   ;ARE THEY CORRECT
6147 070404  001406                          BEQ     430$                    ;BR, IF ALL IS WELL (OK)
6151 070406                                  ERRHRD  ERRNO,T38WRL,EXPREC     ;"WRITE LOCKED BIT IS NOT SET ETC."
     070406  104456                                                                 TRAP    C$ERHRD
     070410  001765                                                                 .WORD   1013
     070412  072534                                                                 .WORD   T38WRL
     070414  015474                                                                 .WORD   EXPREC
6152 070416  005237  002222                  INC     FATFLG                  ;SET FATAL FLAG
6153 070422                          430$:   CKLOOP                          ;LOOP IF SELECTED
     070422  104406                                                                 TRAP    C$CLP1
```

F2

TEST 10: MANUAL INTERVENTION SEQ 0225

```
6154 070424  005737  002222                    TST     FATFLG                  ;WAS THE DRIVE NOT WRITE LOCKED
6155 070430  001402                            BEQ     435$                    ;BR, IF FLAG NOT SET
6156 070432  000137  066752                    JMP     2$                      ;RE-WRITE MENU
6157 070436  017737  112462  072272  435$:     MOV     @FREE,T38WR             ;SET UP WRITE BUFFER ADDRESS
6158 070444  012704  072270                    MOV     #T38PK4,R4              ;GET PACKET ADDRESS
6159 070450  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
6160 070454  004737  016250                    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
6161 070460  016501  000002                    MOV     TSSR(R5),R1             ;GET TSSR
6162 070464  012702  100206                    MOV     #SC!SSR!BIT1!BIT2,R2    ;SET UP EXPECTED
6163 070470  020102                            CMP     R1,R2                   ;ARE THEY EQUAL (CORRECT)
6164 070472  001404                            BEQ     440$                    ;BR, IF CORRECT STATUS
6168 070474                                    ERRHRD  ERRNO,T38WRT,PKTSSR     ;"TSSR INCORRECT AFTER WRITE COMMAND
     070474  104456                                                                    TRAP    C$ERHRD
     070476  001766                                                                    .WORD   1014
     070500  072450                                                                    .WORD   T38WRT
     070502  012046                                                                    .WORD   PKTSSR
6169 070504                            440$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     070504  104406                                                                    TRAP    C$CLP1
6170 070506  013701  071544                    MOV     T38BFR+6,R1             ;READ XSTO CONTENTS
6171 070512  010102                            MOV     R1,R2                   ;SET UPR EXPECTED
6172 070514  052702  004000                    BIS     #BIT11,R2               ;SET THE WRITE LOCK ERROR BIT (XSTO)
6173 070520  020102                            CMP     R1,R2                   ;WAS THE BIT SET
6174 070522  001404                            BEQ     450$                    ;BR, IF IT WAS (GOOD)
6178 070524                                    ERRHRD  ERRNO,T38WLE,EXPREC     ;"WRITE LOCK ERROR BIT NOT SET"
     070524  104456                                                                    TRAP    C$ERHRD
     070526  001767                                                                    .WORD   1015
     070530  072575                                                                    .WORD   T38WLE
     070532  015474                                                                    .WORD   EXPREC
6179 070534                            450$:   CKLOOP                          ;LOOP IF SELECTED
     070534  104406                                                                    TRAP    C$CLP1
6180 070536  000137  066752                    JMP     2$                      ;GO BACK TO MENU
6181
6182                                   ;*************************************************************************
6183                                   ;       SERVO EXERCISER NO LONGER USED
6184                                   ;*************************************************************************
6185 070542                            30$:
6186 070542                                    PRINTB  @T38MS3                 ;"EXE ANY OTHER MENU SELECTION TO STOP
     070542  012746  072355                                                             MOV     @T38MS3,-(SP)
     070546  012746  000001                                                             MOV     #1,-(SP)
     070552  010600                                                                     MOV     SP,R0
     070554  104414                                                                     TRAP    C$PNTB
     070556  062706  000004                                                             ADD     #4,SP
6187 070562  004737  074046                    JSR     PC,T38REST              ;SET PACKET TO INITIAL VALUES
6188 070566  004737  015774                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
6189 070572  103405                            BCS     500$                    ;BR IF SOFT INIT = OK
6193 070574  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6194 070576                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     070576  104455                                                                    TRAP    C$ERDF
     070600  001770                                                                    .WORD   1016
     070602  003652                                                                    .WORD   SFIERR
     070604  012034                                                                    .WORD   SFIMSG
6195 070606  013737  002202  072240  500$:     MOV     UNITN,T38DSW            ;SET UNIT NUMBER
6196 070614  012704  072220                    MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
6197 070620  004737  010662                    JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
6198 070624  103405                            BCS     510$                    ;BR, IF COMMAND ISSUED OK
6202 070626  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6203 070630                                    ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
```

```
        070630  104456                                                          TRAP    C$ERHRD
        070632  001771                                                          .WORD   1017
        070634  005056                                                          .WORD   WRTMSG
        070636  012034                                                          .WORD   SFIMSG
6204    070640                          510$:
6205    070640  112737  000000  071531          MOVB    #0,T38BS1               ;CLEAR BIT #4
6206    070646  112737  000020  071530          MOVB    #20,T38BSO              ;EXECUTE DRIVE TEST 22
6207    070654  012704  071520                  MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
6208    070660  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
6209    070664  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6210    070670  103405                          BCS     550$                    ;BR IF CARRY SET (GOOD RETURN)
6211    070672  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6215    070674                                  ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        070674  104455                                                          TRAP    C$ERDF
        070676  001772                                                          .WORD   1018
        070700  072716                                                          .WORD   T38SSR
        070702  012046                                                          .WORD   PKTSSR
6216    070704                          550$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        070704  104406                                                          TRAP    C$CLP1
6217    070706                                  SETPRI  #PRI06                  ;RAISE THE PRIORITY
        070706  012700  000300                                                  MOV     #PRI06,R0
        070712  104441                                                          TRAP    C$SPRI
6218    070714  005037  071504                  CLR     TTION2                  ;ASSUME INTERRUPTS ARE ENABLED
6219    070720  032737  000100  177560          BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
6220    070726  001005                          BNE     555$                    ;BRANCH IF YES
6221    070730  005237  071504                  INC     TTION2                  ;FLAG SET IF INTERRUPTS OFF
6222    070734  052737  000100  177560          BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
6223    070742  012701  000060          555$:   MOV     #TTIVEC,R1              ;START OF TTI VECTORS
6224    070746  011137  071506                  MOV     (R1),TVSAV2             ;SAVE THE CURRENT TTI VECTOR
6225    070752  012721  071000                  MOV     #590$,(R1)+             ;SET NEW INTERRUPT ROUTINE
6226    070756  011137  071510                  MOV     (R1),TPSAV2             ;SAVE THE VECTOR PRIORITY
6227    070762  012711  000300                  MOV     #PRI06,(R1)             ;USE PRIORITY SIX
6228    070766                                  SETPRI  #PRI00                  ;LOWER INTERRUPT BR LEVEL
        070766  012700  000000                                                  MOV     #PRI00,R0
        070772  104441                                                          TRAP    C$SPRI
6229    070774  000240          560$:   NOP                                     ;LOOP AWHILE
6230    070776  000776                  BR      560$                            ;STAY IN "TIGHT" LOOP
6231                                    ;+
6232                                    ;PROCESS CONSOLE INTERRUPTS
6233                                    ;-
6234
6235    071000  010046          590$:   MOV     R0,-(SP)                        ;SAVE WORK REGISTER
6236    071002  113700  177562          MOVB    @#TTIBFR,R0             ;GET THE OPERATOR INPUT
6237    071006  042700  000200          BIC     #200,R0                 ;STRIP OFF PARITY BIT
6238    071012  122700  000015          CMPB    #15,R0                  ;IS IT A CARRIAGE RETURN ?
6239    071016  001075                  BNE     591$                    ;JUST EXIT IF NOT
6240    071020  012766  066752  000002  MOV     #2$,2(SP)               ;RETURN TO MASTER MENU
6241    071026  005066  000004          CLR     4(SP)                   ;FORCE PRIORTY 0
6242    071032  013737  071506  000060  MOV     TVSAV2,@#TTIVEC ;RESTORE VECTOR
6243    071040  013737  071510  000062  MOV     TPSAV2,@#TTIVEC+2       ;RESTORE SUPER PRIORTY
6244    071046  112737  000025  071530  MOVB    #25,T38BSO              ;STOP DRIVE TEST 22
6245    071054  112737  000000  071531  MOVB    #0,T38BS1               ;CLEAR BS1
6246    071062  012704  071520          MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
6247    071066  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
6248    071072  012737  176750  071512  MOV     #65000.,T38DLY          ;SET UP DELAY COUNTER
6249    071100  004737  016250  592$:   JSR     PC,WAITF                ;DO A WAIT FOR SSR
6250    071104  016501  000002          MOV     TSSR(R5),R1             ;CONTENTS OF TSSR REGISTER
```

H2

TEST 10: MANUAL INTERVENTION SEQ 0227

```
6251 071110  032701  000200              BIT     #SSR,R1          ;CHECK FOR TSSR SET
6252 071114  001017                      BNE     595$             ;KEEP GOING IF NOT SET
6253 071116                              DELAY   250              ;CALL DELAY ROUTINE
     071116  012727  000250                                              MOV     #250,(PC)+
     071122  000000                                                       .WORD   0
     071124  013727  002116                                              MOV     L$DLY,(PC)+
     071130  000000                                                       .WORD   0
     071132  005367  177772                                              DEC     -6(PC)
     071136  001375                                                       BNE     .-4
     071140  005367  177756                                              DEC     -22(PC)
     071144  001367                                                       BNE     .-20
6254 071146  005337  071512              DEC     T38DLY           ;BUMP COUNTER DOWN
6255 071152  001352                      BNE     592$             ;BR, IF MORE TIME LEFT
6256 071154  004737  016336      595$:   JSR     PC,CHKTSSR       ;WAIT FOR SSR TO SET
6257 071160  103405                      BCS     580$             ;BR IF CARRY SET (GOOD RETURN)
6258 071162  010001                      MOV     R0,R1            ;SAVE CONTENTS OF TSSR
6262 071164                              ERRDF   ERRNO,T38SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
     071164  104455                                                       TRAP    C$ERDF
     071166  001773                                                       .WORD   1019
     071170  072716                                                       .WORD   T38SSR
     071172  012046                                                       .WORD   PKTSSR
6263 071174              580$:   CKLOOP                           ;LOOP ON ERROR, IF FLAG SET
     071174  104406                                                       TRAP    C$CLP1
6264 071176  005737  071504              TST     TTION2           ;ARE SUPER INTERRUPTS ENABLED
6265 071202  001403                      BEQ     591$             ;BR, IF YES
6266 071204  042737  000100  177560      BIC     #100,@#TTICSR    ;RESTORE REGISTER
6267 071212  012600              591$:   MOV     (SP)+,R0         ;RESTORE REGISTER
6268 071214  000002                      RTI                      ;RETURN
6269 071216              35$:
6270 071216  004737  074046              JSR     PC,T38REST       ;SET PACKET TO INITIAL VALUES
6271 071222  004737  015774              JSR     PC,SOFINIT       ;DO SOFT INIT OF CONTROLLER
6272 071226  103405                      BCS     600$             ;BR IF SOFT INIT = OK
6276 071230  010001                      MOV     R0,R1            ;SAVE CONTENTS OF TSSR
6277 071232                              ERRDF   ERRNO,SFIERR,SFIMSG  ;DEVICE FATAL ERROR DURING INIT
     071232  104455                                                       TRAP    C$ERDF
     071234  001774                                                       .WORD   1020
     071236  003652                                                       .WORD   SFIERR
     071240  012034                                                       .WORD   SFIMSG
6278 071242              600$:   CKLOOP                           ;LOOP IF SELECTED
     071242  104406                                                       TRAP    C$CLP1
6279 071244  012701  071536              MOV     #T38BFR,R1       ;ADDRESS OF MESSAGE BUFFER
6280 071250  012702  125252              MOV     #125252,R2       ;ALTERNATING 1'S AND 0'S
6281
6282 071254  010221              601$:   MOV     R2,(R1)+         ;CLEAR OUT THE MESSAGE BUFFER
6283 071256  022701  072212              CMP     #T38EB,R1        ;END OF BUFFER YET
6284 071262  001401                      BEQ     605$             ;BR, IF AT END OF BUFFER
6285 071264  000773                      BR      601$             ;NOT AT END KEEP GOING
6286 071266  013737  002202  072240 605$:   MOV     UNITN,T38DSW     ;SET UNIT NUMBER
6287 071274  012704  072220              MOV     #T38PK2,R4       ;SUBROUTINE NEEDS PACKET ADDRESS
6288 071300  004737  010662              JSR     PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
6289 071304  103405                      BCS     610$             ;BR, IF COMMAND ISSUED OK
6293 071306  010001                      MOV     R0,R1            ;SAVE CONTENTS OF TSSR
6294 071310                              ERRHRD  ERRNO,WRTMSG,SFIMSG  ;WRITE CHARACTERISTISC FAILED
     071310  104456                                                       TRAP    C$ERHRD
     071312  001775                                                       .WORD   1021
     071314  005056                                                       .WORD   WRTMSG
     071316  012034                                                       .WORD   SFIMSG
```

```
6295 071320                                610$:   CKLOOP                          ;LOOP IF SELECTED
     071320  104406                                                                        TRAP    C$CLP1
6296 071322  112737  000000  071531               MOVB    #0,T38BS1               ;CLEAR BIT #4
6297 071330  112737  000024  071530               MOVB    #24,T38BSO              ;READ EXTENDED DRIVE STATUS
6298 071336  012704  071520                       MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
6299 071342  010465  000000                       MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
6300 071346  012737  000144  071512               MOV     #100.,T38DLY            ;SET UP DELAY ROUTINE
6301 071354  004737  016250              620$:    JSR     PC,WAITF                ;WAIT AWHILE FOR SSR TO SET
6302 071360  016501  000002                       MOV     TSSR(R5),R1             ;SEE IF IT REALLY DID
6303 071364  032701  000200                       BIT     #SSR,R1                 ;JUST CHECK THAT BIT
6304 071370  001017                               BNE     630$                    ;BR, IF SSR IS SET
6305 071372                                       DELAY   250                     ;DELAY ABOUT .25 SEC
     071372  012727  000250                                                                MOV     #250.,(PC)+
     071376  000000                                                                .WORD   0
     071400  013727  002116                                                                MOV     L$DLY,(PC)+
     071404  000000                                                                .WORD   0
     071406  005367  177772                                                                DEC     -6(PC)
     071412  001375                                                                BNE     .-4
     071414  005367  177756                                                                DEC     -22(PC)
     071420  001367                                                                BNE     .-20
6306 071422  005337  071512                       DEC     T38DLY                  ;START DELAY COUNT DOWN
6307 071426  001352                               BNE     620$                    ;BR, IF COUNTER IS NOT AT DONE
6308 071430  004737  016336              630$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6309 071434  103405                               BCS     650$                    ;BR IF CARRY SET (GOOD RETURN)
6310 071436  010001                               MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6314 071440                                       ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     071440  104455                                                                        TRAP    C$ERDF
     071442  001776                                                                .WORD   1022
     071444  072716                                                                .WORD   T38SSR
     071446  012046                                                                .WORD   PKTSSR
6315 071450                                650$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     071450  104406                                                                        TRAP    C$CLP1
6316 071452  012700  071556                       MOV     #T38BFR+20,R0           ;MESSAGE BUFFER ADDRESS
6317 071456  005001                               CLR     R1                      ;NO HIGH ORDER ADDRESS BITS
6318 071460  005037  003134                       CLR     KTENABLE                ;NO KT11 STUFF EITHER
6319 071464  004737  074104                       JSR     PC,T38MBP               ;GO PRINT MESSAGE BUFFER CONTENTS
6320 071470  000137  066752                       JMP     2$                      ;GO BACK TO MENU
6321
6322
6323 071474  000137  000200              63$:     JMP     200                     ;REALLY RETURN TO THE SUPERVISOR
6324 071500                                64$:     EXIT    TST                     ;LEAVE TEST
     071500  104432                                                                        TRAP    C$EXIT
     071502  003054                                                                .WORD   L10075-.
6325
6326                                               ;+
6327                                               ;LOCAL TEXT MESSAGES FOR TEST
6328                                               ;-
6329
6330                                               ;LOCAL STORAGE FOR THIS TEST
6331                                               ;-
6332                                               ;+
6333                                               ;LOCAL STORAGE FOR THIS TEST
6334                                               ;-
6335
6336 071504  000000                               TTION2:         .WORD   0               ;WORD SET IF SUPERVISOR TTI INTER OFF
6337 071506  000000                               TVSAV2: .WORD   0               ;SAVE TTI VECTOR
6338 071510  000000                               TPSAV2: .WORD   0               ;SAVE TTI PRIORITY
```

J2

TEST 10: MANUAL INTERVENTION SEQ 0229

```
071512  000000      T38DLY: .WORD   0                       ;DELAY COUNTER FOR TEST
        071520              .=<.+10>&177770
071520                  T38PACKET:                          ;COMMAND PACKET FOR TEST
071520  140006              .WORD   140006                  ;WRITE SUBSYSTEM MEM. CMD.  ACK,CVC=1
071522  071530              .WORD   T38TAD                  ;ADDRESS OF CHARACTERISTICS BLOCK
071524  000000              .WORD   0
071526  000012              .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE
071530                  T38TAD:                         ;CHARACTERISTICS DATA BLOCK
071530     000          T38BS0: .BYTE   0                       ;BSEL0 BYTE
071531     000          T38BS1: .BYTE   0                       ;BSEL1 BYTE
071532  000000          T38BS2: .WORD   0                       ;BSEL1 WORD
071534  000000              .WORD   0                       ;DATA
071536                  T38BFR: .BLKW   150.                    ;MESSAGE BUFFER
072212  000000          T38EB:  .WORD                           ;END OF BUFFER ADDRESS


        072220              .=<.+10>&177770
072220                  T38PK2:                             ;COMMAND PACKET FOR TEST
072220  140004              .WORD   140004                  ;WRITE CHARA. MEM. CMND.,  ACK,CVC=1
072222  072230              .WORD   T38DTA                  ;ADDRESS OF SELECT DATA BLOCK
072224  000000              .WORD   0
072226  000012              .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE


072230                  T38DTA:                             ;SELECT DATA BLOCK
072230  071536              .WORD   T38BFR                  ;ADDRESS OF MESSAGE BUFFER
072232  000000              .WORD   0
072234  000400              .WORD   256.                    ;LENGTH OF MESSAGE BUFFER
072236  000000          T38EAI: .WORD   0                       ;EAI BIT WORD
072240  000000          T38DSW: .WORD   0                       ;DRIVE SELECT WORD ETC
        072250              .=<.+10>&177770
072250  140212          T38PK3: .WORD   140212                  ;MESSAGE BUFFER RELEASE COMMAND
072252  000000              .WORD   0                       ;NOT USED
072254  000000              .WORD   0                       ;NOT USED
072256  000000              .WORD   0                       ;NOT USED
072260  000000              .WORD   0                       ;NOT USED
                        ;
                        ;WRITE TAPE PACKET
                        ;
        072270              .=<.+10>&177770
072270  140005          T38PK4: .WORD   140005                  ;WRITE, ACK, CVC=1 COMMAND
072272  000000          T38WR:  .WORD   0                       ;ADDRESS OF WRITE BUFFER
072274  000000              .WORD   0                       ;MORE ADDRESS OF WRITE BUFFER
072276  000400          T38SIZ: .WORD   256.                    ;SIZE OF RECORD


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

TEST 10: MANUAL INTERVENTION

```
                                T38NE:  .ASCIZ  'Stand-alone Manual Intervention Not Executed'
                                T38MS3: .ASCIZ  '<NUL><NUL>A Type <RETURN> To Stop Servo Exerciser, Return To Menu'
                                T38WRT: .ASCIZ  'TSSR Not Correct After WRITE, With WRITE PROTECT On'
                                T38WRL: .ASCIZ  'WRITE LOCKED Bit Not Set In XSTO'
                                T38WLE: .ASCIZ  'WRITE LOCK ERROR Bit Not Set In XSTO'
                                T38NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
                                T38SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
                                T38INT: .ASCIZ  '<NUL><NUL>A Interrupt Received'
                                T38ONL: .ASCIZ  '<NUL><NUL>A Drive Is Now   ON-LINE'
                                T38OFL: .ASCIZ  '<NUL><NUL>A Drive Is Now  OFF-LINE'
                                T38SST: .ASCIZ  'Contents Of TSSR Incorrect After MESSAGE BUFFER RELEASE'
                                T38MS1: .ASCIZ  '<NUL><NUL>AToggle ON-LINE Switch to Generate ATTENTION Interrupts'
                                T38MS2: .ASCIZ  '<NUL><NUL>AType RETURN To Return To Menu<NUL>N'
                                T38MSG: .ASCIZ  'Is Write-Protected Tape Mounted'
                                T38ID:  .ASCIZ  'Manual Intervention'
                                        .EVEN
                                MIMENU: .WORD   1$,2$,3$,4$,5$,6$
                                        .WORD   8$,9$,10$,0

                                1$:     .ASCIZ  <12>'SELECT OPERATION FROM FOLLOWING OPTIONS:'
                                2$:     .ASCIZ  <12>'    0       Display This Menu'
                                3$:     .ASCIZ  '        1       Turn On All M7196 LED''s'
                                4$:     .ASCIZ  '        2       Turn Off All M7196 LED''s'
                                5$:     .ASCIZ  '        3       Offline/Online Attention'
                                6$:     .ASCIZ  '        4       Write Protect Test'
                                8$:     .ASCIZ  '        5       Print Extended Transport Status'
                                9$:     .ASCIZ  '        6       Return to Diagnostic Supervisor'
                                10$:    .ASCIZ  ''
                                        .EVEN

                                ;+
                                ;LOCAL STORAGE FOR THIS TEST
                                ;-

000000                          T38DAT: .WORD   0               ;LOGICAL RESPONSE TO QUESTION
                                T38REST:
                                        SAVREG                  ;SAVE THE REGISTERS
012701  071520                          MOV     #T38PACKET,R1   ;START OF THE PACKET
012721  140206                          MOV     #140206,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
012721  071530                          MOV     #T38TAD,(R1)+   ;ADDRESS OF DATA BLOCK
005021                                  CLR     (R1)+           ;EXTENDED ADDRESS
012721  000006                          MOV     #6.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
005021                                  CLR     (R1)+           ;CLEAR BSEL0 AND BSEL1
005021                                  CLR     (R1)+           ;CLEAR SEL2
005011                                  CLR     (R1)            ;CLEAR DATA AREA
000207                                  RTS     PC              ;RETURN


                                ;+
                                ;
                                ;THIS ROUTINE PRINTS THE CONTENTS OF
                                ;THE 256 BYTE MESSAGE BUFFER RETURNED BY THE
                                ;TSV-05.
                                ;
                                ;INPUT:
                                ;
                                ;       R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
```

TEST 10: MANUAL INTERVENTION

```
                                        ;       R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
                                        ;       NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
                                        ;
                                        ;
                                        ;-
                                        T38MBP:
                                                SAVREG                  ;SAVE THE REGISTERS
                                                MOV     R0,R5           ;SAVE LOW ORDER ADDRESS
                                                TST     KTENABLE        ;ADDRESS ABOVE 28K?
                                                BNE     910$            ;BR IF YES
                                                CLR     R1              ;SET HIGH ORDER ADDRESS TO 0
                                        910$:   MOV     R1,R3           ;SAVE HIGH ORDER ADDRESS
                                                ROL     R0              ;SHIFT BIT15 TO C BIT
                                                ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
                                                PRINTX  #T38AS0,R1,R5   ;PRINT MESSAGE BUFFER ADDRESS
074130  010546                                                                 MOV     R5,-(SP)
074132  010146                                                                 MOV     R1,-(SP)
074134  012746  074406                                                        MOV     #T38AS0,-(SP)
074140  012746  000003                                                        MOV     #3,-(SP)
074144  010600                                                                 MOV     SP,R0
074146  104415                                                                 TRAP    C$PNTX
074150  062706  000010                                                        ADD     #10,SP
                                                PRINTX  #T38AS1         ;PRINT HEADER FOR CONTENTS
074154  012746  074453                                                        MOV     #T38AS1,-(SP)
074160  012746  000001                                                        MOV     #1,-(SP)
074164  010600                                                                 MOV     SP,R0
074166  104415                                                                 TRAP    C$PNTX
074170  062706  000004                                                        ADD     #4,SP
                                                MOV     R5,R1           ;COPY LOW ORDER ADDRESS
                                                MOV     R3,R0           ;COPY HIGH ORDER ADDRESS
                                                BEQ     913$            ;BR IF NOT ABOVE 28K
                                                JSR     PC,SETMAP       ;SETUP PAR ADDRESS IN R0
                                                MOV     R0,R5           ;GET PAR FORMAT ADDRESS ABOVE 28K
                                        913$:   MOV     R5,T38CNT       ;HOLD ADDRESS
                                        911$:   MOV     (R5),R4         ;GET BUFFER ENTRY
                                                CMP     #125252,R4      ;CHECK FOR NO LOAD CONDITION
                                                BEQ     912$            ;BR, IF BUFFER WASN'T LOADED
                                                MOV     R4,R3           ;MAKE COPY
                                                BIC     #170377,R4      ;ONLY BITS 11,10,9 AND 8 ARE SAVED
                                                CLC                     ;CLEAR CARRY
                                                ROR     R4              ;11 TO 10 BIT POSITION
                                                ROR     R4              ;10 TO 9 BIT POSITION
                                                ROR     R4              ;9 TO 8 BIT POSITON
                                                ROR     R4              ;8 TO 7 BIT POSITION
                                                BIC     #177760,R3      ;ONLY BITS 3,2,1 AND 0 ARE SAVED
                                                ADD     R4,R3           ;"OR'EM TOGETHER
                                                MOV     R3,(R5)+        ;PUT BACK IN BUFFER
                                                CMP     R5,#T38EB       ;END OF BUFFER YET
                                                BNE     911$            ;BR, IF NOT AT END YET
                                        912$:   MOV     T38CNT,R5       ;PUT ADDRESS BACK
                                                MOV     #1,R4           ;START BYTE NUMBER AT ONE
                                        915$:   PRINTX  #T38ASN,R4,(R5)+        ;PRT MEM BUFFER W/NEWLINE
074272  012546                                                                 MOV     (R5)+,-(SP)
074274  010446                                                                 MOV     R4,-(SP)
074276  012746  074530                                                        MOV     #T38ASN,-(SP)
074302  012746  000003                                                        MOV     #3,-(SP)
```

TEST 10: MANUAL INTERVENTION

```
        074306  010600                                                          MOV     SP,R0
        074310  104415                                                          TRAP    C$PNTX
        074312  062706  000010                                                  ADD     #10,SP
  6502  074316  005037  074554                  CLR     T38CNT                  ;CLEAR COUNTER
  6503  074322  000412                          BR      921$                    ;SKIP OTHER PRINT
  6504  074324                          920$:   PRINTX  #T38ASC,R4,(R5)+        ;PRINT THE CONTENTS OF MEMORY BUFFER
        074324  012546                                                          MOV     (R5)+,-(SP)
        074326  010446                                                          MOV     R4,-(SP)
        074330  012746  074511                                                  MOV     #T38ASC,-(SP)
        074334  012746  000003                                                  MOV     #3,-(SP)
        074340  010600                                                          MOV     SP,R0
        074342  104415                                                          TRAP    C$PNTX
        074344  062706  000010                                                  ADD     #10,SP
  6505  074350  005237  074554          921$:   INC     T38CNT                  ;BUMP COUNTER
  6506  074354  005204                          INC     R4              ;NUMBER OF THE NEXT
  6507  074356  020427  000200                  CMP     R4,#128.                ;DONE ALL YET ?
  6508  074362  003010                          BGT     50$             ;BRANCH IF ALL DONE
  6509  074364  023727  074554  000004         CMP     T38CNT,#4               ;DONE FOUR YET
  6510  074372  001401                          BEQ     925$                    ;BR. IF THREE DONE
  6511  074374  000753                          BR      920$            ;KEEP GOING
  6512  074376  005037  074554          925$:   CLR     T38CNT                  ;CLEAR COUNTER
  6513  074402  000733                          BR      915$                    ;PRINT WITH NEW LINE
  6514  074404  000207                  50$:    RTS     PC              ;RETURN
  6515
  6516  074406     045     116     045  T38AS0: .ASCIZ  '<N><A Message Buffer Address = <01><05>'
  6517  074453     045     116     045  T38AS1: .ASCIZ  '<N><A Message Buffer Contents:'
  6518  074511     045     101     040  T38ASC: .ASCIZ  '<A  <D4><A: <03>'
  6519  074530     045     116     045  T38ASN: .ASCIZ  '<N><A Byte<D4><A: <03>'
  6520                                   .EVEN
  6521  074554  000000                  T38CNT: .WORD                           ;COUNTER FOR PRINT
  6522  074556                                  ENDTST
        074556                                                          L10075:
        074556  104401                                                          TRAP    C$ETST
  6523                                          .SBTTL  TEST 11: CONFIGURATION TYPEOUT
  6524
  6525                                  ;THIS IS A STANDALONE ROUTINE THAT PRINTS OUT ON THE CONSOLE TERMINAL
  6526                                  ;THE CONFIGURATION OF THE M7196 MODULE AND TSV05 SUBSYSTEM. SPECIFICALLY,
  6527                                  ;THE FOLLOWING INFORMATION IS PRESENTED:
  6528                                  ;
  6529                                  ;
  6530                                  ; 1.0 STATE OF THE EXTENDED FEATURES SWITCH ON THE M7196: ON (EXTENDED
  6531                                  ;     FEATURES ENABLED) OR OFF (EXTENDED FEATURES DISABLED).
  6532                                  ;
  6533                                  ; 2.0 STATE OF THE BUFFERING ENABLE SWITCH ON THE M7196: ON
  6534                                  ;     (BUFFERING ENABLED) OR OFF (BUFFERING DISABLED).
  6535                                  ;
  6536                                  ; 3.0 MICROCODE REVSION LEVEL OF THE M7196.
  6537                                  ;
  6538                                  ; 4.0 NUMBER OF TAPE TRANSPORTS CONNECTED TO THE CONTROLLER.
  6539                                  ;
  6540                                  ; 5.0 UNIT SELECT CODE AND STATE (ONLINE/OFFLINE, WRITE ENABLED/PROTECTED)
  6541                                  ;     OF EACH CONNECTED TRANSPORT. IN ADDITION, THE PROGRAM WILL INDICATE,
  6542                                  ;     FOR EACH ON-LINE TRANSPORT, WHETHER OR NOT IT IS EQUIPPED WITH THE
  6543                                  ;     EXTENDED TAPE STATUS READOUT FEATURE.
  6544                                  ;
  6545                                  ;
  6546                                  ;THE OPERATOR IS EXPECTED TO READ THE PRINTOUT AND VERIFY THAT IT MATCHES
```

```
                        ;THE ACTUAL CONFIGURATION AT HAND.  IF, FOR EXAMPLE, THE PROGRAM INDICATES
                        ;THAT IT "SEES" TWO TRANSPORTS CONNECTED WHEN IN FACT ONLY ONE IS PRESENT,
                        ;THE OPERATOR MUST INTERPRET THIS AS AN ERROR AND ATTEMPT TO FIND THE
                        ;CAUSE (BAD CABLE, FAULTY UNIT-SELECT DECODING IN THE TRANSPORT, ETC.).
                        ;[SINCE THE CONTROLLER CAN ONLY ACCESS UNIT 0 IF IT IS IN "STANDARD"
                        ;MODE, THE PROGRAM WILL FORCE THE MODULE INTO EXTENDED MODE VIA THE
                        ;WRITE SUBSYSTEM MEMORY COMMAND IN ORDER TO SCAN FOR CONNECTED TRANSPORTS.]
                        ;
                        ;
                        ;THIS ROUTINE, WHEN ITS ACTIONS ARE COMPLETED, WILL EXIT BACK TO THE
                        ;DIAGNOSTIC SUPERVISOR SO THAT IF ADDITIONAL UNITS (CONTROLLERS) ARE
                        ;SELECTED (E.G., FROM THE INITIAL STARTUP DIALOG), THE ROUTINE WILL BE
                        ;REENTERED SO THAT THEIR CONFIGURATIONS CAN BE PRINTED.
                        ;
                                BGNTST
                                                                        T11::
                                RFLAGS  R0                      ;GET OPERATOR FLAGS
                                                                        TRAP    C$RFLA
                                BEQ     10$                     ;BR, IF OK TO RUN
                                MOV     #T39NE,R0               ;"TEST NOT EXECUTED"
                                BR      11$                     ;JUMP OUT OF TEST IF NOT
                        10$:    MOV     #TST39ID,R0             ;TEST ID MESSAGE
                        11$:    JSR     PC,TSTSETUP             ;DO THE COMMON SETUP
                                JSR     PC,CHKMAN               ;IS MANUAL INTERVENTION ALLOWED?
                                BCS     20$                     ;BR, IF MANUAL INTERVENTION ALLOWED
                                JMP     64$                     ;JUMP TO OUT IF NOT
                        20$:
                                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
                                BCS     25$                     ;BR IF SOFT INIT = OK
                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                ERRDF   ERRNO,SFIERR SFIMSG     ;DEVICE FATAL ERROR DURING INIT
                                                                        TRAP    C$ERDF
                                                                        .WORD   1101
                                                                        .WORD   SFIERR
                                                                        .WORD   SFIMSG
                        25$:    CKLOOP                                  ;LOOP IF SELECTED
                                                                        TRAP    C$CLP1
                                MOV     UNITN,T39DSW            ;SET UNIT NUMBER
                                MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
                                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
                                BCS     50$                     ;BR, IF COMMAND ISSUED OK
                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
                                                                        TRAP    C$ERHRD
                                                                        .WORD   1102
                                                                        .WORD   WRTMSG
                                                                        .WORD   SFIMSG
                        50$:    CKLOOP                          ;LOOP IF SELECTED
                                                                        TRAP    C$CLP1
                                MOV     T39BFR+12,R1            ;GET XST2 STATUS FROM MESSAGE BUFFER
                                PRINTX  #T39SFS                 ;"STATE OF EXTENDED FEATURES SW ="
                                                                        MOV     #T39SFS,-(SP)
                                                                        MOV     #1,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #4,SP
                                BIT     #BIT7,R1                ;CHECK STATE OF E.F.S.
```

```
6597 074722 001011                      BNE     100$                ;BR, IF EXT. FEA. SW. IS ON
6598 074724                             PRINTX  #T390FF             ;" OFF"
     074724 012746 077545                                                   MOV     #T390FF,-(SP)
     074730 012746 000001                                                   MOV     #1,-(SP)
     074734 010600                                                          MOV     SP,R0
     074736 104415                                                          TRAP    C$PNTX
     074740 062706 000004                                                   ADD     #4,SP
6599 074744 000410                      BR      110$
6600 074746                     100$:   PRINTX  #T390N               ;SKIP OTHER PRINT STATEMENT
     074746 012746 077554                                           ;" ON "
     074752 012746 000001                                                   MOV     #T390N,-(SP)
     074756 010600                                                          MOV     #1,-(SP)
     074760 104415                                                          MOV     SP,R0
     074762 062706 000004                                                   TRAP    C$PNTX
6601 074766                     110$:   PRINTX  #T39SBS                     ADD     #4,SP
     074766 012746 077473                                           ;"STATE OF BUFFERING SWITCH ="
     074772 012746 000001                                                   MOV     #T39SBS,-(SP)
     074776 010600                                                          MOV     #1,-(SP)
     075000 104415                                                          MOV     SP,R0
     075002 062706 000004                                                   TRAP    C$PNTX
6602 075006 032701 000100               BIT     #BIT6,R1                    ADD     #4,SP
6603 075012 001011                      BNE     120$                ;CHECK STATE OF BUFFERING SW
6604 075014                             PRINTX  #T390FF             ;BR, IF BUFFERING IS ON
     075014 012746 077545                                           ;" OFF"
     075020 012746 000001                                                   MOV     #T390FF,-(SP)
     075024 010600                                                          MOV     #1,-(SP)
     075026 104415                                                          MOV     SP,R0
     075030 062706 000004                                                   TRAP    C$PNTX
6605 075034 000410                      BR      130$                        ADD     #4,SP
6606 075036                     120$:   PRINTX  #T390N               ;SKIP OTHER PRINT STATEMENT
     075036 012746 077554                                           ;" ON "
     075042 012746 000001                                                   MOV     #T390N,-(SP)
     075046 010600                                                          MOV     #1,-(SP)
     075050 104415                                                          MOV     SP,R0
     075052 062706 000004                                                   TRAP    C$PNTX
6607 075056 042701 177700       130$:   BIC     #177700,R1                  ADD     #4,SP
6608 075062 010137 077640               MOV     R1,T39RL            ;ONLY LEAVE MICROCODE REV LEVEL
6609 075066                             PRINTX  #T39MCL,T39RL       ;LOAD UP REV LEVEL
     075066 013746 077640                                           ;"MICROCODE REVISION LEVEL =000XXX"
     075072 012746 077563                                                   MOV     T39RL,-(SP)
     075076 012746 000002                                                   MOV     #T39MCL,-(SP)
     075102 010600                                                          MOV     #2,-(SP)
     075104 104415                                                          MOV     SP,R0
     075106 062706 000006                                                   TRAP    C$PNTX
6610 075112 004737 015774               JSR     PC,SOFINIT                  ADD     #6,SP
6611 075116 103405                      BCS     140$                ;DO SOFT INIT OF CONTROLLER
6615 075120 010001                      MOV     R0,R1               ;BR IF SOFT INIT = OK
6616 075122                             ERRDF   ERRNO,SFIERR,SFIMSG ;SAVE CONTENTS OF TSSR
     075122 104455                                                  ;DEVICE FATAL ERROR DURING INIT
     075124 002117                                                          TRAP    C$ERDF
     075126 003652                                                          .WORD   1103
     075130 012034                                                          .WORD   SFIERR
6617 075132                     140$:   CKLOOP                              .WORD   SFIMSG
     075132 104406                                                  ;LOOP IF SELECTED
6618 075134 013737 002202 076520        MOV     UNITN,T39DSW                TRAP    C$CLP1
6619 075142 012704 076500               MOV     #T39PK2,R4          ;SET UNIT NUMBER
6620 075146 004737 010662               JSR     PC,WRTCHR           ;SUBROUTINE NEEDS PACKET ADDRESS
                                                                    ;ISSUE WRITE CHARACTERISTICS
```

```
6621 075152  103405                        BCS     150$                    ;BR, IF COMMAND ISSUED OK
6625 075154  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6626 075156                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     075156  104456                                                               TRAP    C$ERHRD
     075160  002120                                                               .WORD   1104
     075162  005056                                                               .WORD   WRTMSG
     075164  012034                                                               .WORD   SFIMSG
6627 075166                         150$:  CKLOOP                          ;LOOP IF SELECTED
     075166  104406                                                               TRAP    C$CLP1
6628 075170  005737  002226                TST     EXTFEA                  ;CHECK FOR EXTENDED FEATURES SW SWITCH
6629 075174  001036                        BNE     174$                    ;BR IF SWITCH IS ON
6630 075176  112737  000200  076011        MOVB    #200,T39BS1             ;WRITE MISCELLANEOUS CONT/READ STATUS
6631 075204  112737  000010  076010        MOVB    #10,T39BSO              ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
6632 075212  012704  076000                MOV     #T39PACKET,R4           ;WRITE SUBSYS MEM PACKET
6633 075216  010465  000000                MOV     R4,TSD8(R5)             ;ISSUE COMMAND
6634 075222  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR
6635 075226  103405                        BCS     160$                    ;BR, IF NO ERROR
6636 075230  010001                        MOV     R0,R1                   ;ERROR, SAVE TSSR
6640 075232                                ERRHRD  ERRNO,T39NBA,PKTSSR     ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
     075232  104456                                                               TRAP    C$ERHRD
     075234  002121                                                               .WORD   1105
     075236  077255                                                               .WORD   T39NBA
     075240  012046                                                               .WORD   PKTSSR
6641 075242                         160$:  CKLOOP                          ;LOOP IF SELECTED
     075242  104406                                                               TRAP    C$CLP1
6642 075244  012704  076500                MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
6643                                       ;*************************************************************************
6644                                       ;
6645                                       ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
6646                                       ;
6647                                       ;*************************************************************************
6648
6649 075250  004737  010662                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
6650 075254  103405                        BCS     170$                    ;BR, IF COMMAND ISSUED OK
6654 075256  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6655 075260                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     075260  104456                                                               TRAP    C$ERHRD
     075262  002122                                                               .WORD   1106
     075264  005056                                                               .WORD   WRTMSG
     075266  012034                                                               .WORD   SFIMSG
6656 075270                         170$:  CKLOOP                          ;SCOPE LOOP
     075270  104406                                                               TRAP    C$CLP1
6657 075272  005037  002202        174$:  CLR     UNITN                   ;SET TO DRIVE 0
6658 075276  013737  002202  076520 175$:  MOV     UNITN,T39DSW            ;SET UNIT NUMBER
6659 075304  012704  076500                MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
6660 075310  004737  010662                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
6661 075314  103405                        BCS     180$                    ;BR, IF COMMAND ISSUED OK
6665 075316  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6666 075320                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     075320  104456                                                               TRAP    C$ERHRD
     075322  002123                                                               .WORD   1107
     075324  005056                                                               .WORD   WRTMSG
     075326  012034                                                               .WORD   SFIMSG
6667 075330                         180$:  CKLOOP                          ;LOOP IF SELECTED
     075330  104406                                                               TRAP    C$CLP1
6668
6669 075332  016501  000002        190$:  MOV     TSSR(R5),R1             ;GET TSSR STATUS
```

TEST 11: CONFIGURATION TYPEOUT

```
6670 075336  032701  000100                BIT     #OFL,R1                 ;CHECK FOR OFF-LINE
6671 075342  001414                        BEQ     200$                    ;BR, IF DRIVE IS ON-LINE
6672 075344                                PRINTX  #T390F2,UNITN           ;"DRIVE NUMBER XX IS OFF-LINE"
     075344  013746  002202                                                        MOV     UNITN,-(SP)
     075350  012746  077014                                                        MOV     #T390F2,-(SP)
     075354  012746  000002                                                        MOV     #2,-(SP)
     075360  010600                                                                MOV     SP,R0
     075362  104415                                                                TRAP    C$PNTX
     075364  062706  000006                                                        ADD     #6,SP
6673 075370  000137  075724                JMP     250$                    ;DO NOT TRY TO GET ANYMORE INFO.
6674 075374                        200$:   PRINTX  #T390N2,UNITN           ;"DRIVE NUMBER XX IS ON-LINE"
     075374  013746  002202                                                        MOV     UNITN,-(SP)
     075400  012746  077060                                                        MOV     #T390N2,-(SP)
     075404  012746  000002                                                        MOV     #2,-(SP)
     075410  010600                                                                MOV     SP,R0
     075412  104415                                                                TRAP    C$PNTX
     075414  062706  000006                                                        ADD     #6,SP
6675 075420  013701  076024                MOV     T39BFR+6,R1             ;READ EXTENDED STATUS (XST0)
6676 075424  032701  000004                BIT     #BIT2,R1                ;IS DRIVE WTITE PROTECTED
6677 075430  001013                        BNE     210$                    ;BR, IF WRITE PROTECTED
6678 075432                                PRINTX  #T39WPN,UNITN           ;"DRIVE NUMBER IS NOT WRT PRO"
     075432  013746  002202                                                        MOV     UNITN,-(SP)
     075436  012746  077176                                                        MOV     #T39WPN,-(SP)
     075442  012746  000002                                                        MOV     #2,-(SP)
     075446  010600                                                                MOV     SP,R0
     075450  104415                                                                TRAP    C$PNTX
     075452  062706  000006                                                        ADD     #6,SP
6679 075456  000412                        BR      220$                    ;SKIP OVER
6680 075460                        210$:   PRINTX  #T39WRT,UNITN           ;"DRIVE NUMBER XX IS WRT PRO"
     075460  013746  002202                                                        MOV     UNITN,-(SP)
     075464  012746  077123                                                        MOV     #T39WRT,-(SP)
     075470  012746  000002                                                        MOV     #2,-(SP)
     075474  010600                                                                MOV     SP,R0
     075476  104415                                                                TRAP    C$PNTX
     075500  062706  000006                                                        ADD     #6,SP
6681 075504  012737  125252  076116  220$:   MOV     #125252,T39BFR+100      ;SET 1 LOC TO KNOWN VALUE
6682 075512  112737  000000  076011          MOVB    #0,T39BS1               ;EXTENDED TAPE STATUS
6683 075520  112737  000024  076010          MOVB    #24,T39BS0              ;EXTENDED TAPE STATUS
6684 075526  012704  076000                MOV     #T39PACKET,R4           ;WRITE SUBSYS MEM PACKET
6685 075532  010465  000000                MOV     R4,TSDB(R5)             ;ISSUE COMMAND
6686 075536  012737  000144  075774        MOV     #100.,T39DLY            ;SET UP DELAY ROUTINE
6687 075544  004737  016250        222$:   JSR     PC,WAITF                ;WAIT AWHILE FOR SSR TO SET
6688 075550  016501  000002                MOV     TSSR(R5),R1             ;SEE IF IT REALLY DID
6689 075554  032701  000200                BIT     #SSR,R1                 ;JUST CHECK THAT BIT
6690 075560  001017                        BNE     225$                    ;BR, IF SSR IS SET
6691 075562                                DELAY   250                     ;DELAY ABOUT .25 SEC
     075562  012727  000250                                                        MOV     #250.,(PC)+
     075566  000000                                                                .WORD   0
     075570  013727  002116                                                        MOV     L$DLY,(PC)+
     075574  000000                                                                .WORD   0
     075576  005367  177772                                                        DEC     -6(PC)
     075602  001375                                                                BNE     .-4
     075604  005367  177756                                                        DEC     -22(PC)
     075610  001367                                                                BNE     .-20
6692 075612  005337  075774                DEC     T39DLY                  ;START DELAY COUNT DOWN
6693 075616  001352                        BNE     222$                    ;BR, IF COUNTER IS NOT AT DONE
6694 075620  004737  016336        225$:   JSR     PC,CHKTSSR              ;WAIT FOR SSR
```

TEST 11: CONFIGURATION TYPEOUT

```
6695 075624  103405                    BCS     230$                ;BR, IF NO ERROR
6696 075626  010001                    MOV     R0,R1               ;ERROR, SAVE TSSR
6700 075630                            ERRHRD  ERRNO,T39NBA,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
     075630  104456                                                        TRAP    C$ERHRD
     075632  002124                                                        .WORD   1108
     075634  077255                                                        .WORD   T39NBA
     075636  012046                                                        .WORD   PKTSSR
6701 075640                    230$:   CKLOOP                      ;LOOP IF SELECTED
     075640  104406                                                        TRAP    C$CLP1
6702 075642  023727  076116  125252    CMP     T39BFR+100,#125252  ;DID LOC GET OVER WRITTEN
6703 075650  001013                    BNE     240$                ;BR, IF IT DIDN'T GET ETC.
6704 075652                            PRINTX  #T39ETN,UNITN       ;"DRIVE DOESN'T HAVE EXT TAPE STATUS
     075652  013746  002202                                                MOV     UNITN,-(SP)
     075656  012746  076721                                                MOV     #T39ETN,-(SP)
     075662  012746  000002                                                MOV     #2,-(SP)
     075666  010600                                                        MOV     SP,R0
     075670  104415                                                        TRAP    C$PNTX
     075672  062706  000006                                                ADD     #6,SP
6705 075676  000412                    BR      250$                ;SKIP OVER
6706 075700                    240$:   PRINTX  #T39ETS,UNITN       ;"DRIVE HAS EXT TAPE STATUS"
     075700  013746  002202                                                MOV     UNITN,-(SP)
     075704  012746  076632                                                MOV     #T39ETS,-(SP)
     075710  012746  000002                                                MOV     #2,-(SP)
     075714  010600                                                        MOV     SP,R0
     075716  104415                                                        TRAP    C$PNTX
     075720  062706  000006                                                ADD     #6,SP
6707 075724  005237  002202    250$:   INC     UNITN               ;BUMP DRIVE NUMBER
6708 075730  023727  002202  000003    CMP     UNITN,#3            ;AT END OF DRIVES YET
6709 075736  001402                    BEQ     63$                 ;BR, IF NO MORE DRIVES
6710 075740  000137  075276            JMP     175$                ;DO NEXT DRIVE
6711 075744                    63$:    PRINTX  #T39NFL                     ;NEW LINE
     075744  012746  076550                                                MOV     #T39NFL,-(SP)
     075750  012746  000001                                                MOV     #1,-(SP)
     075754  010600                                                        MOV     SP,R0
     075756  104415                                                        TRAP    C$PNTX
     075760  062706  000004                                                ADD     #4,SP
6712 075764  000137  000200            JMP     200                 ;RETURN TO SUPERVISOR
6713 075770                    64$:    EXIT    TST                 ;EXIT THIS SECTION
     075770  104432                                                        TRAP    C$EXIT
     075772  001736                                                        .WORD   L10076-.
6714                                   ;+
6715                                   ;LOCAL TEXT MESSAGES FOR TEST
6716                                   ;-
6717
6718                                   ;LOCAL STORAGE FOR THIS TEST
6719                                   ;-
6720
6721 075774  000000            T39DLY: .WORD   0                           ;DELAY COUNTER FOR TEST
6723         076000                    .=<.+10>&177770
6725 076000                    T39PACKET:                          ;COMMAND PACKET FOR TEST
6726 076000  140006                    .WORD   140006              ;WRITE SUBSYSTEM MEM. CMD.  ACK,CVC=1
6727 076002  076010                    .WORD   T39TAD              ;ADDRESS OF CHARACTERISTICS BLOCK
6728 076004  000000                    .WORD   0
6729 076006  000012                    .WORD   10.                 ;STARTING VALUE OF BLOCK SIZE
6730 076010                    T39TAD:                     ;CHARACTERISTICS DATA BLOCK
6731 076010     000            T39BS0: .BYTE   0                   ;BSEL0 BYTE
6732 076011     000            T39BS1: .BYTE   0                   ;BSEL1 BYTE
```

```
6733 076012  000000                T39BS2: .WORD   0                       ;BSEL1 WORD
6734 076014  000000                        .WORD   0                       ;DATA
6735 076016                        T39BFR: .BLKW   150.                    ;MESSAGE BUFFER
6736
6737
6739         076500                        .=<.+10>&177770
6741 076500                        T39PK2:                                 ;COMMAND PACKET FOR TEST
6742 076500  140004                        .WORD   140004                  ;WRITE CHARA. MEM. CMND.,  ACK,CVC=1
6743 076502  076510                        .WORD   T39DTA                  ;ADDRESS OF SELECT DATA BLOCK
6744 076504  000000                        .WORD   0
6745 076506  000012                        .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE
6746
6747
6748 076510                        T39DTA:                                 ;SELECT DATA BLOCK
6749 076510  076016                        .WORD   T39BFR                  ;ADDRESS OF MESSAGE BUFFER
6750 076512  000000                        .WORD   0
6751 076514  000400                        .WORD   256.                    ;LENGTH OF MESSAGE BUFFER
6752 076516  000000                T39EAI: .WORD   0                       ;EAI BIT WORD
6753 076520  000000                T39DSW: .WORD   0                       ;DRIVE SELECT WORD ETC
6755         076530                        .=<.+10>&177770
6757 076530  140012                T39PK3: .WORD   140012                  ;MESSAGE BUFFER RELEASE COMMAND
6758 076532  000000                        .WORD   0                       ;NOT USED
6759                                   ;
6760                                   ;WRITE TAPE PACKET
6761                                   ;
6763         076540                        .=<.+10>&177770
6765 076540  140005                T39PK4: .WORD   140005                  ;WRITE, ACK, CVC=1 COMMAND
6766 076542  000000                T39WR:  .WORD   0                       ;ADDRESS OF WRITE BUFFER
6767 076544  000000                        .WORD   0                       ;MORE ADDRESS OF WRITE BUFFER
6768 076546  000400                T39SIZ: .WORD   256.                    ;SIZE OF RECORD
6769
6770
6771
6772
6773                                   ;+
6774                                   ;LOCAL TEXT MESSAGES FOR TEST
6775                                   ;-
6776
6777
6778
6779
6780 076550      045     116     000  T39NFL: .ASCIZ  '%N'
6781 076553      123     164     141  T39NE:  .ASCIZ  'Stand-alone Configuration Typeout Not Executed'
6782 076632      045     116     045  T39ETS: .ASCIZ  '%N%A Extended Tape Status Availiable, Drive Number %D2'
6783 076721      045     116     045  T39ETN: .ASCIZ  '%N%A Extended Tape Status NOT Availiable, Drive Number %D2'
6784 077014      045     116     045  T39OF2: .ASCIZ  '%N%A Drive Number %D2%A Is Off-Line'
6785 077060      045     116     045  T39ON2: .ASCIZ  '%N%A Drive Number %D2%A Is On-Line'
6786 077123      045     116     045  T39WRT: .ASCIZ  '%N%A Drive Number %D2%A Is Write Protected'
6787 077176      045     116     045  T39WPN: .ASCIZ  '%N%A Drive Number %D2%A Is NOT Write Protected'
6788 077255      127     122     111  T39NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
6789 077331      103     157     156  T39SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
6790
6791 077421      045     116     045  T39SFS: .ASCIZ  '%N%A State Of Extended Features Switch  ='
6792 077473      045     116     045  T39SBS: .ASCIZ  '%N%A State Of Buffering Switch          ='
6793 077545      045     101     040  T39OFF: .ASCIZ  '%A OFF'
6794 077554      045     101     040  T39ON:  .ASCIZ  '%A ON '
6795 077563      045     116     045  T39MCL: .ASCIZ  '%N%A M7196 Microcode Revision Level     =%02'
```

```
                                        .EVEN
077640  000000                  T39RL:  .WORD   0
                                        .EVEN
                                        .EVEN


                                ;+
                                ;LOCAL STORAGE FOR THIS TEST
                                ;-

077642  000000                  T39DAT: .WORD   0               ;LOGICAL RESPONSE TO QUESTION
077644                          T39REST:
                                        SAVREG                  ;SAVE THE REGISTERS
077650  012701  076000                  MOV     #T39PACKET,R1   ;START OF THE PACKET
077654  012721  140006                  MOV     #140006,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
077660  012721  076010                  MOV     #T39TAD,(R1)+   ;ADDRESS OF DATA BLOCK
077664  005021                          CLR     (R1)+           ;EXTENDED ADDRESS
077666  012721  000006                  MOV     #6.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
077672  005021                          CLR     (R1)+           ;CLEAR BSEL0 AND BSEL1
077674  005021                          CLR     (R1)+           ;CLEAR SEL2
077676  005011                          CLR     (R1)            ;CLEAR DATA AREA
077700  000207                          RTS     PC              ;RETURN

                                ;+
                                ;LOCAL TEXT MESSAGES FOR TEST
                                ;-

077702      103     157     156 TST39ID:        .ASCIZ  'Configuration Typeout'
                                        .EVEN
077730                                  ENDTST
077730
077730  104401                                                          L10076:
                                                                                TRAP    C$ETST

                        .SBTTL  TEST 12: SCOPE LOOPS


                ;+
                ;
                ;
                ;       THIS IS A STANDALONE ROUTINE PROVIDING A NUMBER OF TIGHT "SCOPE
                ;       LOOPS" USEFUL FOR DEBUGGING BASIC REGISTER ACCESS PROBLEMS WITH
                ;       THE M7196 MODULE.  THESE SCOPE LOOPS CAN BE USED WHEN THE NORMAL
                ;       "LOOP ON ERROR" OR "LOOP ON TEST (SUBTEST)" FACILITIES DON'T
                ;       SEEM TO ALLOW THE OPERATOR TO ZERO IN A PROBLEM IN THE EARLY
                ;       TESTS (I.E, THE HARDWARE MAY NOT BE RESPONDING TO A REGISTER
                ;       ACCESS, CAUSING A BUS ERROR TRAP, EVEN THOUGH THE DEVICE ADDRESS
                ;       SELECTED BY THE PROGRAM MATCHES THE CONFIGURATION SET UP IN THE
                ;       HARDWARE DIP SWITCHES).  THE FOLLOWING MENU OF SCOPE LOOPS ARE
                ;       AVAILABLE:
                ;
                ;               CODE    SCOPE LOOP
                ;
                ;                 0     HELP. PRINT THIS MENU.
                ;                 1     TSBA READ ACCESS
                ;                 2     TSSR READ ACCESS
                ;                 3     INITIALIZE (TSSR WRITE ACCESS)
                ;                 4     TSDB HIGH BYTE WRITE ACCESS
```

```
                        ;                5     TSDB LOW BYTE WRITE ACCESS
                        ;                6     TSDB MAINTENANCE-MODE WORD WRITE ACCESS
                        ;                7     TSDBX (TSSR HIGH BYTE) WRITE ACCESS
                        ;                        (EXTENDED FEATURES SWITCH MUST BE ON
                        ;                         TO USE SELECTION CODE 7)
                        ;                8     EXIT (RETURN TO SUPERVISOR)
                        ;
                        ;       FOR SCOPE LOOPS THAT WRITE INTO REGISTERS, THE PROGRAM PROMPTS
                        ;       THE OPERATOR FOR THE DATA TO BE WRITTEN.  TYPING <RETURN> CAUSES
                        ;       AN EXIT FROM THE SCOPE LOOP BACK TO MENU LEVEL.
                        ;
                        ;-


                                BGNTST
                                                                        T12::
                                RFLAGS  R0                      ;GET OPERATOR FLAGS
                                                                        TRAP    C$RFLA
                                BEQ     1$                      ;BR, IF OK TO RUN
                                MOV     #T40NE,R0               ;"TEST NOT EXECUTED"
                                BR      100$                    ;JUST EXIT IF NOT
                        1$:     MOV     #TST40ID,R0             ;TEST ID MESSAGE
                        100$:   JSR     PC,TSTSETUP             ;DO THE COMMON SETUP
                                JSR     PC,CHKMAN               ;SEE IF MANUAL INTERVENTION ALLOWED
                                BCS     2$                      ;CARRY SET IF INTERVENTION ALLOWED
                                JMP     64$                     ;EXIT IF NO MANUAL INTERVENTION
                        2$:     JSR     PC,SOFINIT              ;DO A SOFT INIT
                                BCS     5$                      ;BRANCH IF OK
                                MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
                                ERRDF   ERRNO,SFIERR,SFIMSG     ;REPORT FATAL ERROR
                                                                        TRAP    C$ERDF
                                                                        .WORD   1201
                                                                        .WORD   SFIERR
                                                                        .WORD   SFIMSG
                        5$:     MOV     #SCMENU,R0              ;MENU OF SCOPE LOOP SELECTIONS
                                MOV     #8.,R1                  ;MAXIMUM ALLOWED SELECTION
                                JSR     PC,GETSEL               ;GO GET THE OPERATORS SELECTION
                                TST     R0                      ;WAS ZERO SPECIFIED ?
                                BEQ     2$                      ;REPEAT MENU IF YES.
                                CMP     R0,#7                   ;EXTENDED TSSR ?
                                BNE     3$                      ;BRANCH IF NOT
                                TST     EXTFEA                  ;CHECK FOR EXTENDED FEATURES SET
                                BNE     3$                      ;BR, IF IT IS ON
                                PRINTF  #EXFMSG                 ;WARN OPERATOR EXTENDED FEATURES CLEAR
                                                                        MOV     #EXFMSG,-(SP)
                                                                        MOV     #1,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTF
                                                                        ADD     #4,SP
                                JMP     2$                      ;GO BACK TO BASIC MENU
                        3$:     MOV     R0,R4                   ;SAVE THE MENU SELECTION
                                SETPRI  #PRI06                  ;RAISE THE PRIORITY
                                                                        MOV     #PRI06,R0
                                                                        TRAP    C$SPRI
                                CLR     TTION                   ;ASSUME INTERRUPTS ARE ENABLED
                                BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
                                BNE     4$                      ;BRANCH IF YES
```

```
 6902 100112  005237  100456                 INC     TTION                ;FLAG SET IF INTERRUPTS OFF
 6903 100116  052737  000100  177560         BIS     @#100,@#TTICSR       ;ENABLE INTERRUPTS
 6904 100124  012701  000060          4$:    MOV     #TTIVEC,R1           ;START OF TTI VECTORS
 6905 100130  011137  100460                 MOV     (R1),TVECSAV         ;SAVE THE CURRENT TTI VECTOR
 6906 100134  012721  100360                 MOV     #60$,(R1)+           ;SET NEW INTERRUPT ROUTINE
 6907 100140  011137  100462                 MOV     (R1),TPRISAV         ;SAVE THE VECTOR PRIORITY
 6908 100144  012711  000300                 MOV     #PRI06,(R1)          ;USE PRIORITY SIX
 6909 100150                                 SETPRI  #PRI00               ;LOWER INTERRUPT BR LEVEL
      100150  012700  000000                                                      MOV     #PRI00,R0
      100154  104441                                                              TRAP    C$SPRI
 6910 100156  006304                         ASL     R4                   ;CONVERT TO WORD OFFSET
 6911 100160  000174  100164                 JMP     @6$(R4)              ;JUMP TO PROPER LOOP
 6912 100164  077766          6$:    .WORD   2$                           ;RETYPE THE MENU
 6913 100166  100206                 .WORD   10$                          ;TSBA READ ACCESS
 6914 100170  100216                 .WORD   15$                          ;TSSR READ ACCESS
 6915 100172  100230                 .WORD   20$                          ;TSSR WRITE ACCESS
 6916 100174  100250                 .WORD   25$                          ;TSDB HIGH BYTE WRITE ACCESS
 6917 100176  100274                 .WORD   30$                          ;TSDB LOW BYTE WRITE ACCESS
 6918 100200  100320                 .WORD   35$                          ;TSDB MAINTENANCE MODE
 6919 100202  100340                 .WORD   40$                          ;TSDBX WRITE ACCESS
 6920 100204  100452                 .WORD   65$                          ;LEAVE THE TEST
 6921
 6922
 6923 100206  105065  000000    10$:    CLRB    TSDB(R5)                 ;ENTER MAINTENANCE MODE
 6924 100212  011500    12$:    MOV     (R5),R0                          ;READ TSBA REGISTER
 6925 100214  000776            BR      12$                              ;LOOP UNTIL HALTED
 6926
 6927
 6928 100216  012703  000002    15$:    MOV     #TSSR,R3                 ;ADDRESS OF TSSR REGISTER
 6929 100222  060503                    ADD     R5,R3                    ;POINT TO TSV05'S REGISTERS
 6930 100224  011300    18$:    MOV     (R3),R0                          ;READ TSSR REGISTER
 6931 100226  000776            BR      18$                              ;LOOP UNTIL STOPPED
 6932
 6933 100230  004737  020174    20$:    JSR     PC,GETPAT                ;READ THE DATA PATTERN
 6934 100234  010001                    MOV     R0,R1                    ;DATA PATTERN FOR LOOP
 6935 100236  012703  000002            MOV     #TSSR,R3                 ;ADDRESS OF TSSR
 6936 100242  060503                    ADD     R5,R3                    ;POINT TO TSV05'S REGISTERS
 6937 100244  010113    22$:    MOV     R1,(R3)                          ;WRITE DATA TO TSSR
 6938 100246  000776            BR      22$                              ;LOOP
 6939
 6940
 6941 100250  105065  000000    25$:    CLRB    TSDB(R5)                 ;ENTER MAINTENANCE MODE
 6942 100254  004737  020174            JSR     PC,GETPAT                ;READ THE DATA PATTERN
 6943 100260  010001                    MOV     R0,R1                    ;DATA PATTERN FOR LOOP
 6944 100262  012703  000001            MOV     #TSDBH,R3                ;ADDRESS OF HIGH BYTE OF TSDB
 6945 100266  060503                    ADD     R5,R3                    ;POINT TO TSV05'S REGISTERS
 6946 100270  110113    27$:    MOVB    R1,(R3)                          ;WRITE THE DATA TO TSDB, HIGH BYTE
 6947 100272  000776            BR      27$                              ;LOOP UNTIL STOPPED
 6948
 6949
 6950 100274  105065  000000    30$:    CLRB    TSDB(R5)                 ;ENTER MAINTENANCE MODE
 6951 100300  004737  020174            JSR     PC,GETPAT                ;READ THE DATA PATTERN
 6952 100304  010001                    MOV     R0,R1                    ;DATA PATTERN FOR LOOP
 6953 100306  012703  000000            MOV     #TSDB,R3                 ;ADDRESS OF TSSR
 6954 100312  060503                    ADD     R5,R3                    ;POINT TO TSV05'S REGISTERS
 6955 100314  110113    32$:    MOVB    R1,(R3)                          ;WRITE DATA TO TSSR, LOW BYTE
 6956 100316  000776            BR      32$                              ;LOOP UNTIL HALTED BY OPERATOR
```

TEST 12: SCOPE LOOPS

```
6957
6958 100320  004737  020174          35$:    JSR     PC,GETPAT               ;READ THE DATA PATTERN
6959 100324  010001                          MOV     R0,R1                   ;DATA PATTERN FOR LOOP
6960 100326  012703  000000                  MOV     #TSDB,R3                ;SELECT TSDB
6961 100332  060503                          ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
6962 100334  010113                  37$:    MOV     R1,(R3)                 ;WRITE THE DATA PATTERN
6963
6964 100336  000776                          BR      37$                     ;LOOP UNTIL HALTED
6965
6966 100340  004737  020174          40$:    JSR     PC,GETPAT               ;READ THE DATA PATTERN
6967 100344  010001                          MOV     R0,R1                   ;SAVE THE DATA PATTERN
6968 100346  012703  000003                  MOV     #TSSRH,R3               ;BYTE ADDRESS OF TSSR, HIGH BYTE
6969 100352  060503                          ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
6970 100354  110113                  42$:    MOVB    R1,(R3)                 ;WRITE THE DATA TO REGISTER
6971 100356  000776                          BR      42$                     ;LOOP UNTIL HALTED
6972
6973
6974                                         ;+
6975                                         ;PROCESS CONSOLE INTERRUPTS
6976                                         ;-
6977
6978 100360  010046                  60$:    MOV     R0,-(SP)                ;SAVE WORK REGISTER
6979 100362  113700  177562                  MOVB    @#TTIBFR,R0             ;GET THE OPERATOR INPUT
6980 100366  042700  000200                  BIC     #200,R0                 ;STRIP OFF PARITY BIT
6981 100372  122700  000015                  CMPB    #15,R0                  ;IS IT A CARRIAGE RETURN ?
6982 100376  001021                          BNE     61$                     ;JUST EXIT IF NOT
6983 100400  012766  077766  000002          MOV     #2$,2(SP)               ;RETURN TO MASTER MENU
6984 100406  005066  000004                  CLR     4(SP)                   ;FORCE PRIORITY ZERO
6985 100412  013737  100460  000060          MOV     TVECSAV,@#TTIVEC        ;RESTORE SUPERVISOR VECTOR
6986 100420  013737  100462  000062          MOV     TPRISAV,@#TTIVEC+2      ;RESTORE SUPERVISOR PRIORITY
6987 100426  005737  100456                  TST     TTION                   ;ARE SUPERVISOR INTERRUPTS ENABLED ?
6988 100432  001403                          BEQ     61$                     ;BRANCH IF YES
6989 100434  042737  000100  177560          BIC     #100,@#TTICSR           ;TURN OFF TTI INTERRUPTS
6990 100442  012600                  61$:    MOV     (SP)+,R0                ;RESTORE REGISTER
6991 100444  000002                          RTI                             ;RETURN FROM INTERUPT
6992
6993 100446                          64$:
6994 100446                          63$:    EXIT    TST                     ;EXIT THE TEST
                                                                                     TRAP    C$EXIT
     100446  104432                                                                  .WORD   L10077-.
     100450  000736
6995 100452  000137  000200          65$:    JMP     200                     ;RETURN TO SUPERVISOR
6996
6997                                         ;+
6998                                         ;LOCAL STORAGE FOR THIS TEST
6999                                         ;-
7000
7001 100456  000000                  TTION:          .WORD   0               ;WORD SET IF SUPERVISOR TTI INTER OFF
7002 100460  000000                  TVECSAV:        .WORD   0               ;SAVE TTI VECTOR
7003 100462  000000                  TPRISAV:        .WORD   0               ;SAVE TTI PRIORITY
7004
7005
7006                                         ;+
7007                                         ;MENU FOR OPERATOR INPUT FOR SCOPE LOOPS
7008                                         ;-
7009
7010                                 .EVEN
7011 100464  100516  100571  100617  SCMENU: .WORD   1$,2$,3$,4$,5$,6$
```

K3

SEQ 0243

TEST 12: SCOPE LOOPS

```
7012 100500  100770  101026  101074          .WORD   7$,8$,9$,10$,11$,12$,0
7013
7014
7015 100516     012     123     105  1$:     .ASCIZ  <12>'SELECT SCOPE LOOP FROM FOLLOWING OPTIONS:'
7016 100571     012     011     060  2$:     .ASCIZ  <12>'    0       Display This Menu'
7017 100617     011     061     011  3$:     .ASCIZ  '        1       TSBA Read Access'
7018 100643     011     062     011  4$:     .ASCIZ  '        2       TSSR Read Access'
7019 100667     011     063     011  5$:     .ASCIZ  '        3       Initialize (TSSR Write Access)'
7020 100731     011     064     011  6$:     .ASCIZ  '        4       TSDB High Byte Write Access'
7021 100770     011     065     011  7$:     .ASCIZ  '        5       TSDB Low Byte Write Access'
7022 101026     011     066     011  8$:     .ASCIZ  '        6       TSDB Maintenance Mode Write Access'
7023 101074     011     067     011  9$:     .ASCIZ  '        7       TSDBX (TSSR High Byte) Write Access'
7024 101143     011     070     011  10$:    .ASCIZ  '        8       Return to Diagnostic Supervisor'
7025 101206     000             011  11$:    .ASCIZ  ''
7026 101207     124     171     160  12$:    .ASCIZ  'Type RETURN To Stop Scope Loops'
7027 101247     045     116     045  EXFMSG: .ASCIZ  '%N%A *** Extended Features Switch Not On *** '
7028 101325     123     164     141  T40NE:  .ASCIZ  'Stand-alone Scope Loops Not Executed'
7029 101372     123     143     157  TST40ID:        .ASCIZ  'Scope Loops'
7030                                  .EVEN
7031 101406                          ENDTST
     101406                                                          L10077:
     101406  104401                                                          TRAP    C$ETST
7032 101410                          ENDMOD
```

TEST 12: SCOPE LOOPS

```
                                        .TITLE  TSV6 - PARAMETER CODING


                                        BGNMOD  TSV6
                                TSV6::

                                        .SBTTL  HARDWARE PARAMETER CODING SECTION

                                ;++
                                ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--
                                        BGNHRD
101410  000010                        .WORD L10100-L$HARD/2
101412                          L$HARD::

                                        GPRMA   HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
101412  000031                        .WORD   T$CODE
101414  101432                        .WORD   HPM1
101416  160010                        .WORD   T$LOLIM
101420  177776                        .WORD   T$HILIM
                                        GPRMA   HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
101422  001031                        .WORD   T$CODE
101424  101466                        .WORD   HPM2
101426  000000                        .WORD   T$LOLIM
101430  000776                        .WORD   T$HILIM
                                        ;GPRMD  HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
                                        ENDHRD
                                        .EVEN
101432                          L10100:
101432      104     105     126 HPM1:   .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
101466      111     116     124 HPM2:   .ASCIZ  'INTERRUPT VECTOR    '
101512      111     116     124 HPM3:   .ASCIZ  'INTERRUPT PRIORITY      '
                                        .EVEN
```

SOFTWARE PARAMETER CODING SECTION

```
                                        .SBTTL  SOFTWARE PARAMETER CODING SECTION

                                ;++
                                ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--
101542                                  BGNSFT
101542  000003                          .WORD L10101-L$SOFT/2
101544                          L$SOFT::
                                ;       GPRML   SPM1,0,-1,YES                   ; GET TRANSPORT TEST FLAG.
101544                                  GPRML   SPM4,2,-1,YES                   ; GET ITERATION CONTROL.
101544  001130                          .WORD   T$CODE
101546  101602                          .WORD   SPM4
101550  177777                          .WORD   -1
                                ;       GPRMD   SPM6,4,D,7777,0,7777,YES        ; GET LOCAL ERROR LIMIT
                                ;       GPRMD   SPM7,6,D,7777,0,7777,YES        ; GET GLOBAL ERROR LIMIT
101552                                  ENDSFT
                                        .EVEN
101552                          L10101:

101552  105  116  101  SPM1:   .ASCIZ  'ENABLE TRANSPORT TESTS '
101602  111  116  110  SPM4:   .ASCIZ  'INHIBIT ITERATIONS     '
101632  120  105  122  SPM6:   .ASCIZ  'PER TEST ERROR LIMIT   '
101662  120  105  122  SPM7:   .ASCIZ  'PER UNIT ERROR LIMIT   '
                                        .SBTTL  PATCH AREA

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

101712                          PATCH::

101712                                  .BLKW   32.

                                        .=.!377+1
1024)0                                  LASTAD          ;SET LAST USED ADDRESS.
                                        .EVEN
102400  000000                          .WORD   0
102402  000000                          .WORD   0
102404                          L$LAST::
102404                                  ENDMOD
000001                                  .END
```

N3

SEQ 0246

SYMBOL TABLE

```
ADDSSR  012126 G
ADR     = 000020 G
AMBTSS  006635
ASSEMB= 000010
A1716   = 000003
BADDAT  003156 G
BADSSR  015700 G
BDVPCR= 177520 G
BENBSW  002230 G
BIE     = 040000
BIT0    = 000001 G
BIT00   = 000001 G
BIT01   = 000002 G
BIT02   = 000004 G
BIT03   = 000010 G
BIT04   = 000020 G
BIT05   = 000040 G
BIT06   = 000100 G
BIT07   = 000200 G
BIT08   = 000400 G
BIT09   = 001000 G
BIT1    = 000002 G
BIT10   = 002000 G
BIT11   = 004000 G
BIT12   = 010000 G
BIT13   = 020000 G
BIT14   = 040000 G
BIT15   = 100000 G
BIT2    = 000004 G
BIT3    = 000010 G
BIT4    = 000020 G
BIT5    = 000040 G
BIT6    = 000100 G
BIT7    = 000200 G
BIT8    = 000400 G
BIT9    = 001000 G
BOE     = 000400 G
BRINIT  004457
BSEL0   = 000000
BSEL1   = 000001
CHKAMB  016044
CHKMAN  020500 G
CHKTSS  016336
CKDROP  017202
CKEMAX  017102
CKMSG   011360 G
CKMSG2  011500 G
CKRAM   011114 G
CKRAM2  011224 G
CMDPKT  021260 G
CMPMEM  017660
CONFIG  017250
COUNT   002310 G
CSRADD  002206 G
CTAB    003164 G
CTABE   003176 G
CTABM   003164 G
C$AU    = 000052
C$AUTO= 000061
C$BRK   = 000022
C$BSEG= 000004
C$BSUB= 000002
C$CEFG= 000045
C$CLCK= 000062
C$CLEA= 000012
C$CLOS= 000035
C$CLP1= 000006
C$CVEC= 000036
C$DCLN= 000044
C$DODU= 000051
C$DRPT= 000024
C$DU    = 000053
C$EDIT= 000003
C$ERDF= 000055
C$ERHR= 000056
C$ERRO= 000060
C$ERSF= 000054
C$ERSO= 000057
C$ESCA= 000010
C$ESEG= 000005
C$ESUB= 000003
C$ETST= 000001
C$EXIT= 000032
C$GETB= 000026
C$GETW= 000027
C$GMAN= 000043
C$GPHR= 000042
C$GPLO= 000030
C$GPRI= 000040
C$INIT= 000011
C$INLP= 000020
C$MANI= 000050
C$MEM   = 000031
C$MSG   = 000023
C$OPEN= 000034
C$PNTB= 000014
C$PNTF= 000017
C$PNTS= 000016
C$PNTX= 000015
C$QIO   = 000377
C$RDBU= 000007
C$REFG= 000047
C$RESE= 000033
C$REVI= 000003
C$RFLA= 000021
C$RPT   = 000025
C$SEFG= 000046
C$SPRI= 000041
C$SVEC= 000037
C$TPRI= 000013
DATA    002312 G
DATASC  020232
DEBUGM  011632
DEVCNT  002220 G
DEVDRO  023424
DEVNRD  023343
DEVNXR  023261
DEVONL  023211
DEVSUM  023154
DFPTBL  002156 G
DIAGMC= 000000
DICEB   = 000001
DSBINT  016204
DUAD12  004643
DUFLG   003112 G
DUMMY   003062
EF.CON= 000036 G
EF.NEW= 000035 G
EF.PWR= 000034 G
EF.RES= 000037 G
EF.STA= 000040 G
EMAXDU  016777
EN      = 000000
ENAINT  016152
ENVIRN  020630
EPRTSW  002200 G
EPRT1   006360
EPRT2   006360
ERCM    011733
ERRHI   002236 G
ERRK    016756
ERRLO   002240 G
ERRNO   = 002261
ERRVEC= 000004 G
ERTABE  003376
ERTABL  003176
ESUM    016760
EVL     = 000004 G
EXBCNT= 000010
EXFMSG  101247
EXPBRE  015502 G
EXPD    002232 G
EXPGOT  004533
EXPGT2  004567
EXPMSG  002322 G
EXPREC  015474 G
EXTA    005772
EXTEND  005770
EXTFEA  002226 G
E$END   = 002100
E$LOAD= 000035
FATERR= 000060
FATFLG  002222 G
FERCM   011722
FIFEXP  012170 G
FIF1MS  012242
FIF2MS  012311
FILLME  017422
FNOINT  004215
FORCER  002176 G
FREE    003124 G
FREEHI  003130
FRESIZ  003126 G
FUSI    004117
F$AU    = 000015
F$AUTO= 000020
F$BGN   = 000040
F$CLEA= 000007
F$DU    = 000016
F$END   = 000041
F$HARD= 000004
F$HW    = 000013
F$INIT= 000006
F$JMP   = 000050
F$MOD   = 000000
F$MSG   = 000011
F$PROT= 000021
F$PWR   = 000017
F$RPT   = 000012
F$SEG   = 000003
F$SOFT= 000005
F$SRV   = 000010
F$SUB   = 000002
F$SW    = 000014
F$TEST= 000001
GDDAT   003160 G
GERRMA  002174 G
GETPAT  020174 G
GETSEL  020256 G
G$CNTO= 000200
G$DELM= 000372
G$DISP= 000003
G$EXCP= 000400
G$HILI= 000002
G$LOLI= 000001
G$NO    = 000000
G$OFFS= 000400
G$OFSI= 000376
G$PRMA= 000001
G$PRMD= 000002
G$PRML= 000000
G$RADA= 000140
G$RADB= 000000
G$RADD= 000040
G$RADL= 000120
G$RADO= 000020
G$XFER= 000004
G$YES   = 000010
HIADDR= 001400
HOE     = 100000 G
HPM1    101432
HPM2    101466
HPM3    101512
IBE     = 010000 G
IDU     = 000040 G
IER     = 020000 G
IFAULT  004256
INCERK  017044
INTCPC  016150
INTFLA  016145
INTMAS  016144
INTR    016216 G
INTREC  002224 G
INTVEC  016146
INTX    004300
INVERT  021206 G
IOKCKI= 000200
IOKSTP= 000001
IPRI    002212 G
ISR     = 000100 G
IVEC    002210 G
IXE     = 004000 G
I$AU    = 000041
I$AUTO= 000041
I$CLN   = 000041
I$DU    = 000041
I$HRD   = 000041
I$INIT= 000041
I$MOD   = 000041
I$MSG   = 000041
I$PROT= 000040
I$PTAB= 000041
I$PWR   = 000041
I$RPT   = 000041
I$SEG   = 000041
I$SETU= 000041
I$SFT   = 000041
I$SRV   = 000041
I$SUB   = 000041
I$TST   = 000041
J$JMP   = 000167
KIPAR0= 172340
KIPAR1= 172342
KIPAR2= 172344
KIPAR3= 172346
KIPAR4= 172350
KIPAR5= 172352
KIPAR6= 172354
KIPAR7= 172356
KIPDR0= 172300
KIPDR1= 172302
KIPDR2= 172304
KIPDR3= 172306
KIPDR4= 172310
KIPDR5= 172312
KIPDR6= 172314
KIPDR7= 172316
KTENAB  003134 G
KTFLG   003132 G
KTINIT  021054
KTOFF   017274
KTON    017256
LERRMA  002172 G
LERRNO= 000000
LISTAL= 000001
```

SYMBOL TABLE

SEQ 0247

```
LOE   = 040000 G    L$UNIT  002012 G    L10071  056560      OFL   = 000100      PRMNO   002320 G
LOOPCN  002216 G    L10000  002164      L10072  060024      ONEFIL= 000000      PRMSGE  014552 G
LOOPCO  013126      L10001  002176      L10073  066714      O$APTS= 000000      PRMSGO  014732
LOOPFL  003162 G    L10002  005766      L10074  065054      O$AU  = 000001      PRMSG1  014777
LOT   = 000010 G    L10003  012044      L10075  074556      O$BGNR= 000001      PRMSG2  015035
L$ACP   002110 G    L10004  012062      L10076  077730      O$BGNS= 000001      PROASC  014420
L$APT   002036 G    L10005  012100      L10077  101406      O$DU  = 000001      PR1ASC  014465
L$AU    022400 G    L10006  012106      L10100  101432      O$ERRT= 000000      PST32W  003152 G
L$AUT   002070 G    L10007  012124      L10101  101552      O$GNSW= 000001      PUNIT   022332
L$AUTO  022604 G    L10010  012142      MEMADD  013754 G    O$POIN= 000001      PW.D11= 000021
L$CCP   002106 G    L10011  012166      MEMCK   021276 G    O$SETU= 000000      PW.D13= 000022
L$CLEA  022664 G    L10012  012240      MENASC  020447      PASRPT  022102      PW.D22= 000020
L$CO    002032 G    L10013  012410      MENERR  020374      PATCH   101712 G    PW.NOP= 000000
L$DEPO  002011 G    L10014  013124      MENRES  020476      PATDAT  020230      PW.NO1= 000023
L$DESC  003410 G    L10015  013752      MIMENU  073442      PC.ERA= 002400      PW.RDE= 000024
L$DESP  002076 G    L10016  013774      MMVEC = 000250      PC.IER= 002000      PW.RDR= 000001
L$DEVP  002060 G    L10017  015500      MSA.FR= 000006      PC.NOO= 001000      PW.RDS= 000005
L$DISP  002124 G    L10020  015506      MSA.NO= 000000      PC.REL= 000000      PW.RFI= 000003
L$DLY   002116 G    L10021  015514      MSA.NR= 000004      PC.REW= 000400      PW.WCT= 000006
L$DTP   002040 G    L10022  015526      MSA.VO= 000002      PKBCNT= 000006      PW.WFI= 000004
L$DTYP  002034 G    L10023  015550      MSGEXP  012144 G    PKHI  = 000004      PW.WFM= 000007
L$DU    022476 G    L10024  015576      MSGLOO  013064 G    PKLOW = 000002      PW.WMI= 000010
L$DUT   002072 G    L10025  015736      MSGSTA  012350 G    PKTADD  007554      PW.WNP= 000011
L$DVTY  003402 G    L10026  016246      MSGSUB  013742 G    PKTFRM  007516      PW.WTR= 000002
L$EF    002052 G    L10030  022330      MS.ATT= 000006      PKTGET  012064 G    P.ACK = 100000
L$ENVI  002044 G    L10031  022474      MS.EXT= 000200      PKTMES  012110 G    P.CMD = 000037
L$ETP   002102 G    L10032  022602      MS.RSD= 000001      PKTRAM  004745 G    P.CONT= 000012
L$EXP1  002046 G    L10033  022662      MS.RSF= 000020      PKTSSR  012046 G    P.CVC = 040000
L$EXP4  002064 G    L10034  022710      MS.RST= 000010      PNT   = 001000 G    P.FMT = 000140
L$EXP5  002066 G    L10035  023152      M8186   005554      PRAMPK  013776      P.FORM= 000011
L$HARD  101412 G    L10036  024454      M8189   005645      PRASC   014523      P.GETS= 000017
L$HIME  002120 G    L10037  026446      NBA   = 002000      PRBEXP  015470      P.IE  = 000200
L$HPCP  002016 G    L10040  024730      NEWPAS  022036      PRBMSG  015336      P.INIT= 000013
L$HPTP  002022 G    L10041  025174      NODEV   003114 G    PRBREC  015472      P.MODE= 007400
L$HW    002156 G    L10042  032042      NOEXTF  030236      PRBTOT  015423      P.OPP = 020000
L$ICP   002104 G    L10043  027022      NOINIT  004335      PRBYTE  015122 G    P.POSI= 000010
L$INIT  021556 G    L10044  027312      NOINTR  004221      PRI   = 002000 G    P.READ= 000001
L$LADP  002026 G    L10045  027610      NOITS   002170 G    PRIADD  010160      P.SWB = 010000
L$LAST  102404 G    L10046  030242      NOMAN   020534      PRIAO   010230      P.WRIT= 000005
L$LOAD  002100 G    L10047  034632      NOMEM   005460      PRIBXO  007612 G    P.WRTC= 000004
L$LUN   002074 G    L10050  032376      NP.IR = 000200      PRIEQU  010060      P.WRTS= 000006
L$MREV  002050 G    L10051  032770      NP.LOO= 000040      PRIPKT  007370 G    QVP     002204 G
L$NAME  002000 G    L10052  033376      NP.OUT= 000100      PRIRAM  010066      RAMASC  014156
L$PRIO  002042 G    L10053  040424      NP.WRP= 000020      PRITAD  010274      RAMDAT  002242 G
L$PROT  021546 G    L10054  035716      NSI     004152      PRITSS  006024      RAMERR  015510 G
L$PRT   002112 G    L10055  036650      NSINIT  004407      PRITO   010356      RAMEXP  015530 G
L$REPP  002062 G    L10056  050536      NUL     004527      PRIT1   010421      RAMFOR  010116
L$REV   002010 G    L10057  040730      NULCR   004530      PRIXOR  007742 G    RAMSIZ  002302 G
L$RPT   022712 G    L10060  042140      NXM   = 004000      PRIO0 = 000000 G    RAMTAD  015516 G
L$SOFT  101544 G    L10061  043500      NXMFLG  003136 G    PRIO1 = 000040 G    RCVHIA  002304 G
L$SPC   002056 G    L10062  044056      NXMHI   003142 G    PRIO2 = 000100 G    RCVLOA  002306 G
L$SPCP  002020 G    L10063  045330      NXMLO   003140 G    PRIO3 = 000140 G    RDERR   005206
L$SPTP  002024 G    L10064  046374      NXMTST  021452      PRIO4 = 000200 G    RECMSG  002466 G
L$STA   002030 G    L10065  052016      NXR     003740      PRIO5 = 000240 G    RECV    002234 G
L$SW    002166 G    L10066  062644      NXRERR  005736 G    PRIO6 = 000300 G    REGSAV  020140
L$TEST  002114 G    L10067  054022      NXRX    003777      PRIO7 = 000340 G    RETERR  005372
L$TIML  002014 G    L10070  055310      NXTU    022050      PRMESS  014242      REWIND  011014 G
```

SYMBOL TABLE

```
RMCHBE= 000167          S1.IID= 004000          TST40I  101372          T10     066716 G        T158FR  033442
RMCHEN= 000200          S1.I1R= 020000          TSV2    002000 G        T11     074560 G        T158F2  034130
RMMSGB= 000215          S1.I2R= 040000          TSV3    002176 G        T12     077732 G        T158S0  034130
RMMSGE= 000234          S1.PAR= 100000          TSV4    021546 G        T128FR  030332          T158S1  034131
RMPKTB= 000201          S2.ATI= 000010          TSV5    023474 G        T128KG  031007          T15DAT  033430
RMPKTE= 000210          S2.BTI= 000004          TSV6    101410 G        T128LK  030364          T15LOO  032074
RMR   = 010000          S2.DIM= 000200          TTIBFR= 177562 G        T12CHA  032000          T15PAC  033420
RWPACK  011110          S2.ILW= 000100          TTICSR= 177560 G        T12CKR  031560 G        T15PK2  034120
SC    = 100000          S2.INR= 000020          TTION   100456          T12CON  031366          T15RES  034522
SCE   = 020000          S2.OUT= 000040          TTION2  071504          T12DAT  030320          T15RT2  034574
SCHERR  005300          S2.UND= 000003          TTIVEC= 000060 G        T12DPR  031166          T15SSR  034136
SCME    005013          TBLEND= 003062 G        TVECSA  100460          T12GET  030546          T15S2   034132
SCMENU  100464          TCOASC  006476          TVSAV2  071506          T12HIA  030352          T15S3   034134
SDELAY  010660          TCOCOD  006676          T$ARGC= 000001          T12KT   030360          T16BEN  040310
SELASC  020442          TEMP1   003116 G        T$CODE= 001130          T12LOA  030354          T16BFR  040262
SELDAT= 000004          TEMP2   003120 G        T$ERRN= 002261          T12LOO  026476          T16BFS  040302
SEL2  = 000002          TERCLS= 000016          T$EXCP= 000000          T12MSG  030711          T16CLR  040126
SETMAP  017316          TESTNO= 000014          T$FLAG= 000040          T12NIN  031075          T16DAT  040250
SETU    022134          TEXASC  006435          T$GMAN= 000000          T12NXM  031257          T16DT2  040320
SFFMSG  012102 G        TFCASC  006537          T$HILI= 000776          T12PAC  030310          T16D01  036702
SFHERR  003705          TIMEXP  015552 G        T$LAST= 000001          T12PAR  030356          T16D28  036704
SFIERR  003652          TIMSGO  015600          T$LOLI= 000000          T12SET  031736          T16D53  036706
SFIMSG  012034 G        TINERR  012021          T$LSYM= 010000          T12SWR  031670          T16D78  036710
SFPTBL  002166 G        TMPBFR  002632 G        T$LTNO= 000014          T12TBE  030516          T16LEN  037232
SIFLAG  003154 G        TNAM    016704          T$NEST= 177777          T12WRT  030622          T16LOO  034652
SIMSG   011766          TPRISA  100462          T$NS0 = 000000          T121LO  026610          T16PAC  040240
SKIPT   003400          TPSAV2  071510          T$NS1 = 000005          T122LO  027144          T16PK2  040310
SOFINI  015774 G        TRANST  002166 G        T$NS2 = 000002          T123LO  027364          T16REJ  037344
SPACE   010466 G        TSBA  = 000000 G        T$NS3 = 000003          T124LO  030006          T16RES  040060
SPM1    101552          TSBAH = 000001 G        T$PTNU= 000000          T124TS  030362          T16SEX  040220
SPM4    101602          TSDB  = 000000 G        T$SAVL= 177777          T138FR  024112          T16SRD  040152
SPM6    101632          TSDBH = 000001 G        T$SEGL= 177777          T13DAT  024100          T16SSR  036762
SPM7    101662          TSFCOD  007236          T$SEKO= 010000          T13LOO  023512          T16TSB  037130
SR0   = 177572          TSREJ = 000006          T$SUBN= 000000          T13MEM  024205          T16T01  037461
SR1   = 177574          TSSDEF  006606          T$TAGL= 177777          T13NBA  024244          T16T28  037560
SR2   = 177576          TSSR  = 000002 G        T$TAGN= 010102          T13PAC  024070          T16T53  037660
SR3   = 172516          TSSRBI  003502 G        T$TEMP= 000000          T13RES  024406          T16T78  037760
SSR   = 000200          TSSRFO  006415          T$TEST= 000014          T13SSR  024317          T16WMI  040172
STATCO  012412          TSSRH = 000003 G        T$TSTM= 177777          T14BFR  025226          T162SS  037017
SVCGBL= 000000          TSSX    004020          T$TSTS= 000001          T14BSO  025220          T163SS  037063
SVCINS= 000000          TSTBLK  002752 G        T$$AU = 010031          T14BS1  025221          T17BEN  050412
SVCSUB= 000001          TSTCNT  002214 G        T$$AUT= 010033          T14BS2  025222          T17BFR  050272
SVCTAG= 000000          TSTEND  016720          T$$CLE= 010034          T14DAT  025220          T17BFS  050312
SVCTST= 000001          TSTFLA  002314 G        T$$DU = 010032          T14DTA  025640          T17CLE  050176
S$LSYM= 010000          TSTLOO  016456 G        T$$HAR= 010100          T14LOO  024474          T17CLR  050010
SO.IDB= 000010          TSTPTR  002316 G        T$$HW = 010000          T14NBA  025652          T17DAT  050260
SO.IFB= 000002          TSTSET  016510 G        T$$INI= 010030          T14NIN  026113          T17DT2  050430
SO.IFP= 000001          TST12I  030520          T$$MSG= 010025          T14PAC  025210          T17EXE  046554
SO.ILD= 000020          TST13I  024132          T$$PRO= 010027          T14PK2  025630          T17EXP  046432
SO.ION= 000040          TST14I  026271          T$$RPT= 010035          T14RES  026336          T17EXS  046452
SO.IRD= 000100          TST15I  034503          T$$SEG= 010000          T14RST  026374          T17LOO  040454
SO.IRW= 000004          TST16I  036712          T$$SOF= 010101          T14SSR  026023          T17MSK  046426
SO.ISP= 000200          TST17I  046656          T$$SRV= 010026          T14TSB  026205          T17PAC  050250
S1.ICE= 002000          TST18I  051246          T$$SUB= 010074          T142RE  025726          T17PK2  050420
S1.IEO= 010000          TST19I  060232          T$$SW = 010001          T15AM2  034212          T17RFI  050156
S1.IFM= 001000          TST20I  065232          T$$TES= 010077          T15AM3  034313          T17RSF  050054
S1.IME= 000400          TST39I  077702          T1      023474 G        T15AM4  034415          T17SET  050216
```

SYMBOL TABLE

```
T17SNP  050076        T19SSR  060273        T3      026450 G      T39NBA  077255        WF.IED= 000010
T17SRD  050034        T19WCT  062172        T38FLG  003150 G      T39NE   076553        WF.IER= 000004
T17SSR  046675        T19WFI  062136        T3.1    026476        T39NFL  076550        WF.IMI= 000200
T17WFD  046554        T19WFM  062212        T3.2    027036        T39OFF  077545        WF.IRE= 000040
T17WFI  050122        T19WST  061573        T3.3    027326        T39OF2  077014        WF.IWF= 000020
T171CM  047215        T191CM  060730        T3.4    027624        T39ON   077554        WF.IWR= 000100
T172CM  047277        T192CM  061012        T38ASC  074511        T39ON2  077060        WF.I3R= 000002
T172SS  046732        T192SS  060330        T38ASN  074530        T39PAC  076000        WF.I4R= 000001
T173CM  047373        T193CM  061106        T38ASO  074406        T39PK2  076500        WRTCHR  010662 G
T173SS  046776        T193SS  060374        T38AS1  074453        T39PK3  076530        WRTERR  005113
T174CM  047457        T194SS  060441        T38BFR  071536        T39PK4  076540        WRTMSG  005056
T174SS  047043        T195CM  061174        T38BSO  071530        T39RES  077644        WSMBK   021270 G
T175CM  047542        T195SS  060504        T38BS1  071531        T39RL   077640        XFERAS  015740
T175SS  047106        T196CM  061250        T38BS2  071532        T39SBS  077473        XNXM    016376
T176CM  047616        T196SS  060550        T38CNT  074554        T39SFS  077421        XORBFO  007674
T176SS  047152        T197CM  061333        T38DAT  074044        T39SIZ  076546        XORFOR  010012
T177CM  047724        T197SS  060613        T38DLY  071512        T39SSR  077331        XST0  = 000006 G
T18BFR  051752        T198CM  061421        T38DSW  072240        T39TAD  076010        XST1  = 000010 G
T18CMP  051453        T198SS  060662        T38DTA  072230        T39WPN  077176        XST2  = 000012 G
T18DAT  051740        T199CM  061510        T38EAI  072236        T39WR   076542        XST3  = 000014 G
T18DT2  052010        T2      024456 G      T38EB   072212        T39WRT  077123        XST4  = 000016 G
T18EXP  051216        T2.1    024474        T38ID   073415        T4      032044 G      XSOBOT= 000002
T18LOO  050556        T2.2    024744        T38INT  073006        T4.1    032074        XSOEOT= 000001
T18MSK  051212        T208EN  066572        T38MBP  074104        T4.2    032400        XSOIE = 000040
T18PAC  051730        T208FR  066452        T38MSG  073355        T4.3    032772        XSOILA= 000400
T18PK2  052000        T208FS  066472        T38MS1  073216        T40NE   101325        XSOILC= 001000
T18SET  051620        T20CLE  066364        T38MS2  073311        T5      034634 G      XSOLET= 020000
T18SMI  051576        T20CLR  066152        T38MS3  072355        T5.1    034652        XSOMOT= 000200
T18SRD  051556        T20CWP  066057        T38NBA  072642        T5.2    035732        XSONEF= 002000
T18SSR  051305        T20DAT  066440        T38NE   072300        T6      040426 G      XSOONL= 000100
T18XS   051242        T20DT2  066610        T38OFL  073072        T6.1    040454        XSOPED= 000010
T182SS  051342        T20EXE  065232        T38ONL  073036        T6.2    040744        XSORLL= 010000
T183SS  051406        T20EXP  065112        T38PAC  071520        T6.3    042154        XSORLS= 040000
T19BEN  062522        T20EXS  065132        T38PK2  072220        T6.4    043514        XSOTMK= 100000
T19BFC  060046        T20LOO  062664        T38PK3  072250        T6.5    044072        XSOVCK= 000020
T19BFR  062402        T20MSK  065106        T38PK4  072270        T6.6    045344        XSOWLE= 004000
T19BFS  062422        T20PAC  066430        T38RES  074046        T7      050540 G      XSOWLK= 000004
T19CLE  062250        T20PK2  066600        T38SIZ  072276        T8      052020 G      XXCOMM  003122 G
T19CLR  062004        T20RFI  066236        T38SSR  072716        T8.1    052036        X$ALWA= 000000
T19CNV  062314        T20RSF  065756        T38SST  073126        T8.2    054036        X$FALS= 000040
T19DAT  062370        T20SET  066404        T38TAD  071530        T8.3    055324        X$OFFS= 000400
T19DT2  062540        T20SEX  066346        T38WLE  072575        T8.4    056574        X$TRUE= 000020
T19EXE  060232        T20SRD  066176        T38WR   072272        T9      062646 G      X1.COR= 020000
T19EXP  060112        T20SSR  065261        T38WRL  072534        T9.1    062664        X1.DLT= 100000
T19EXS  060132        T20SWP  065647        T38WRT  072450        UAM   = 000200 G      X1.MBZ= 017375
T19LOO  052036        T20WFI  066256        T39BFR  076016        UNITN   002202 G      X1.RBP= 000400
T19MSK  060106        T20WFM  066312        T39BSO  076010        UNREC = 000006        X1.SPA= 040000
T19PAC  062360        T20WMI  066332        T39BS1  076011        USI     004123        X1.UNC= 000002
T19PK2  062530        T20WNP  066216        T39BS2  076012        WAITF   016250 G      X2.BUF= 000100
T19PRE  060044        T202SS  065316        T39DAT  077642        WC.IFA= 000200        X2.EXT= 000200
T19RFI  062116        T203SS  065362        T39DLY  075774        WC.IFE= 000002        X2.OPM= 100000
T19RSF  062050        T204SS  065427        T39DSW  076520        WC.IGO= 000001        X2.RCE= 040000
T19RST  061700        T205SS  065472        T39DTA  076510        WC.IRE= 000010        X2.REV= 000077
T19SET  062270        T206SS  065536        T39EAI  076516        WC.IRW= 000004        X2.SPA= 035400
T19SEX  062232        T208SS  065601        T39ETN  076721        WC.IOT= 000100        X2.UNI= 000007
T19SNP  062072        T23A    003144 G      T39ETS  076632        WC.I1T= 000040        X2.WCF= 002000
T19SRD  062030        T23B    003146 G      T39MCL  077563        WC.I5R= 000020        X3.DCK= 000010
```

```
TSV6 - PARAMETER CODING MACRO M1200  20-MAR-84 08:26  PAGE 30-5                                                    SEQ 0250

SYMBOL TABLE

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

. ABS.  102404     000
        000000     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  32264 WORDS  ( 127 PAGES)
DYNAMIC MEMORY:  20060 WORDS  ( 77 PAGES)
ELAPSED TIME:  00:08:06
CNTSBAO.BIC,CNTSBAO.SEQ/-SP=SVC34/ML,TSV1B,TSV22B,TSV3B,TSV4,TSV55B,TSV6
.
```