TSV05

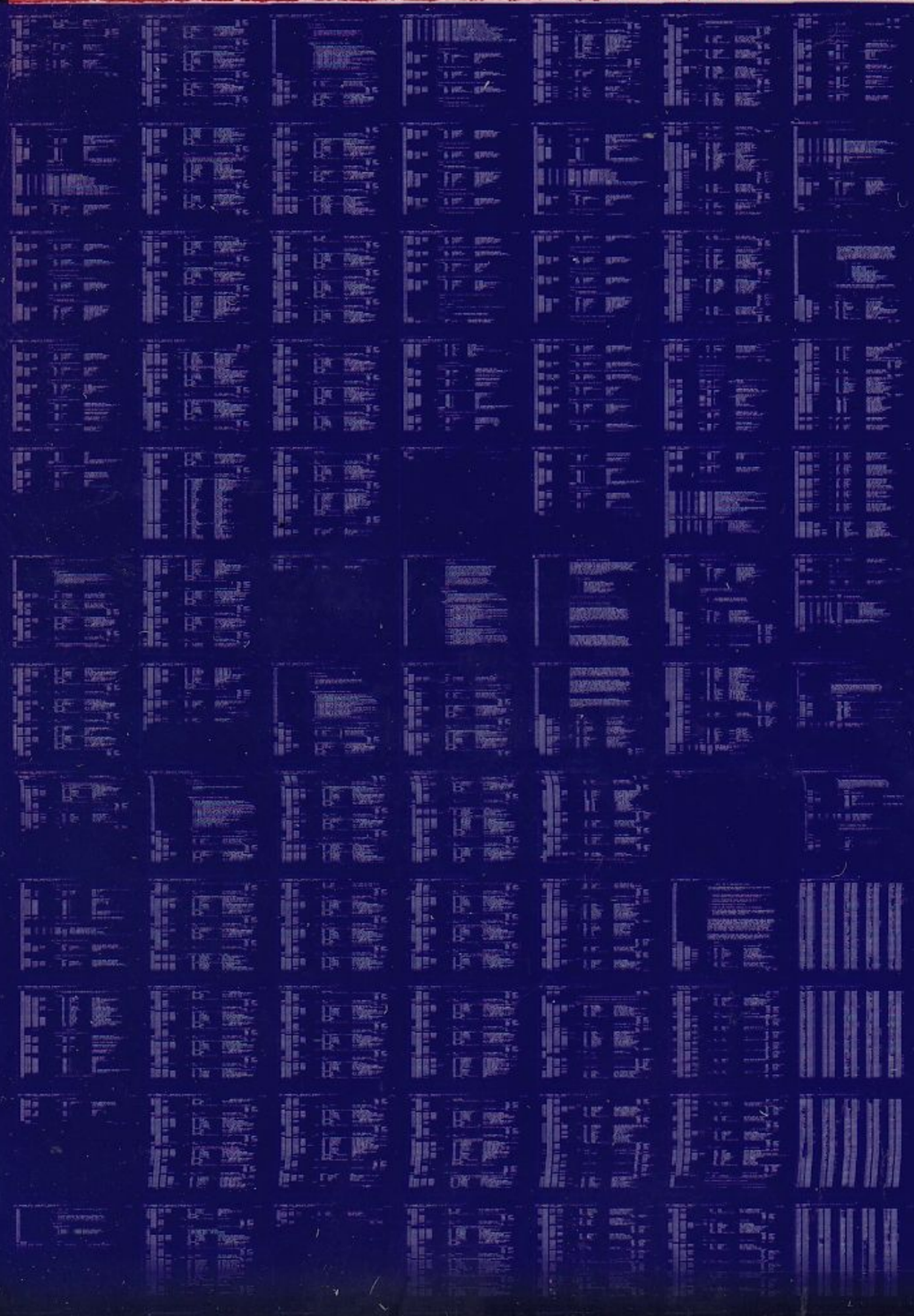
TSV05 CTRL LT2
CVTSBA0

AH-T096A-MC
FICHE 1 OF 2

TSV05

TSV05 CTRL LT2
CVTSBAO

AH-T096A-MC
FICHE 2 OF 2



.REM_

IDENTIFICATION

ID: AC-T095A-MC

PRODUCT TITLE: CVTSBA0 TSV05 CTRL LT2

AUTHOR: DICK GORDON

MAINTAINER: SCOTT SNOWDON

DATE: MARCH 08, 1982

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

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

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

1.2 SYSTEM REQUIREMENTS

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

1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

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

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

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

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

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

FUNCTIONAL PDP-11/23 CENTRAL PROCESSOR AND MEMORY
FUNCTIONAL CONSOLE TERMINAL
FUNCTIONAL STANDALONE DIAGNOSTIC SUPERVISOR

FUNCTIONAL DIAGNOSTIC LOADER/MONITOR (XXDP+)

1.5 ASSUMPTIONS

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

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

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

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

2.1.1 OPERATOR COMMANDS

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

BOOT THE DIAGNOSTIC MEDIA

```
.R VTSB??
DIAG. RUN-TIME SERVICES REV D. APR 79
CVTSB-A-0
```

```
****TSV05 LOGIC DIAGNOSTIC****
UNIT IS TSV05
>DR
```

2.2 SWITCHES

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

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

EXAMPLE OF SWITCH USAGE:

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

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

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

	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 = 172520, VECTOR = 224

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

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

UNIT 0

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

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

THE ADDRESS AND VECTOR QUESTIONS WILL BE ASKED FOR EACH OF THE NUMBER OF UNITS (CONTROLLERS) SPECIFIED IN THE "# UNITS?" QUESTION. LOGICAL UNIT NUMBERS ARE ASSIGNED IN ORDER, BEGINNING AT 0. UP TO FOUR UNITS CAN BE SELECTED FOR TESTING AS FOLLOWS:
UP TO 4 TSV05 CONTROLLERS PER 11/23 AND UP TO 2 DRIVES PER CONTROLLER

2.5 SOFTWARE QUESTIONS

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

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

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

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

2.6 EXTENDED P-TABLE DIALOGUE

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

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

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

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

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

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

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

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

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

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

UNIT 8
CSR ADDRESS (O)  160000<CR>
```

```
        SUB-DEVICE # (0) ?  7<CR>
        Q-FACTOR (0) 1 ?  <CR>
```

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

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

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

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

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

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

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

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

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

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

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

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

2.7 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

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

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

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

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

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

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

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

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

3.2 SPECIFIC ERROR MESSAGES

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

ERROR MESSAGE EXAMPLE 1

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

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

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

ERROR MESSAGE EXAMPLE 2

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

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

ERROR MESSAGE EXAMPLE 3

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

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

4.0 PERFORMANCE AND PROGRESS REPORTS

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

SUCCESSFUL RUN EXAMPLE (PDP-11/23)

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

 UNITS (D) ?  1

 UNIT 0

 DEVICE ADDRESS  (O) 172520 ? <CR>

 VECTOR (O) 224 ? <CR>

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

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

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

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

0 ERRORS

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

PROGRAM RUN TIMES

THE AVERAGE RUN TIMES OF THE PROGRAM ARE LISTED BELOW.THESE FIGURES ARE TO BE USED AS A GUIDE. THE TIMING WAS DONE ON A PDP-11/23 PROCESSOR WITH A LA-34 CONSOLE.

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

TEST NUMBER	N/I SECS.	ITER SECS	DEF SECS.
1	15	50	35
2	1	6	5
3	1	1	0
4	110	540	430
5	1	10	9
6	10	120	110
7	1	3	2
8	15	15	12
9	17	17	13

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

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

5.0 DEVICE INFORMATION TABLES

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

CHANGE HW (L) ?

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

UNIT 0

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

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

THE ADDRESS AND VECTOR QUESTIONS WILL BE ASKED FOR EACH OF THE NUMBER OF

UNITS (CONTROLLERS) SPECIFIED IN THE "# UNITS?" QUESTION. LOGICAL UNIT NUMBERS ARE ASSIGNED IN ORDER, BEGINNING AT 0. UP TO FOUR UNITS CAN BE SELECTED FOR TESTING.

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

CHANGE SW (L) ?

6.0 TEST SUMMARIES

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

TEST DESCRIPTION:

THIS TEST VERIFIES THAT A HARDWARE INITIALIZE COMMAND INVOKED AFTER A WRITE CHARACTERISTICS COMMAND SETS UP THE COMMAND, MESSAGE AND CHARACTERISTIC IMAGE BLOCKS IN THE CONTROLLER RAM CORRECTLY.

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

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

TEST 3: DMA MEMORY ADDRESSING

THIS TEST VERIFIES THAT THE CONTROLLER CAN PROPERLY ADDRESS AND ACCESS ALL AVAILABLE CPU MEMORY (OTHER THAN THAT OCCUPIED BY THE DIAGNOSTIC AND DIAGNOSTIC SUPERVISOR CODE) FOR BOTH READING (DATI) AND WRITING (DATO). VERIFIED ARE THE LSI-11 BUS DRIVERS FOR ALL AVAILABLE ADDRESS LINES. UP TO THIS POINT ONLY 16 BITS HAVE BEEN USED FOR DMA TRANSFERS.

****

CAUTION

THE LSI BUS DRIVERS FOR ALL AVAILABLE ADDRESS LINES ARE ONLY CHECKED WHEN RUNNING ON A 11/23B SYSTEM WITH MORE THAN 128K WORDS OF MEMORY!

****

TEST 4: RAM EXERCISER TEST

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

TEST 5: EXTENDED FEATURES SWITCH AND TIMERS A,B

TEST DESCRIPTION:

THIS TEST VERIFIES THE INVERT EXTENDED FEATURES FUNCTION
CAN LOGICALLY INVERT THE EXTENDED FEATURES SWITCH AND
THAT THE INTERNAL TIMERS A AND B OPERATE CORRECTLY.

TEST 6: FIFO EXERCISER

TEST DESCRIPTION:

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

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

TEST DESCRIPTION:

WRITE TO TSSR REGISTER TO SOFT INITIALIZE THE CONTROLLER
DO WRITE CHARACTERISTICS TO CHECK FOR EXTENDED FEATURES SWITCH
IF EXTENDED FEATURES HARDWARE SWITCH CLEAR THEN:
  DO WRITE SUBSYSTEM WRITE MISCELLANEOUS TO SET EXTENDED FEATURES.
DO WRITE CHARACTERISTICS TO SELECT RESERVED UNIT 7
DO A WRITE SUBSYSTEM READ STATUS
IF ANY TRANSPORT INTERFACE SIGNALS ARE ASSERTED THEN PRINT ERROR

TEST 8: TRANSPORT BUS INTERFACE LOOPBACK TEST

TEST DESCRIPTION:

THIS TEST VERIFIES THE CONTROLLER'S TRANSPORT BUS
DRIVERS, RECEIVERS, AND SIGNAL LOOPBACK LOGIC. NOTE
THAT THE STATIC TRANSPORT BUS TEST MUST HAVE RUN
CORRECTLY FOR THIS TEST TO PROVIDE MEANINGFUL RESULTS.

TEST 9: READ/WRITE DATA PARITY TEST

TEST DESCRIPTION:

THIS TEST VERIFIES THAT THE WRITE DATA PARITY GENERATOR
AND THE READ DATA PARITY CHECKER OPERATE PROPERLY. THE
TRANSPORT BUS SIGNAL LOOPBACK MODE IS ENABLED AND A
SET WRONG PARITY FUNCTION IS EXECUTED. THEN VARIOUS
WRITE SUBSYSTEM MEMORY FUNCTIONS ARE PERFORMED TO
WRITE DATA TO AND FROM THE FIFO IN LOOPBACK MODE.
THE PROGRAM THEN CHECKS TO INSURE A READ DATA PARITY
ERROR OCCURRED.
A RESET FIFO IS DONE AND THE READ DATA PARITY

ERROR BIT IS AGAIN TESTED TO INSURE IT CLEARED.
FINALLY A CLEAR WRONG PARITY FUNCTION IS DONE
AND IT IS VERIFIED THE DATA WORD CAN PASS IN LOOPBACK
MODE WITHOUT SETTING READ DATA PARITY ERROR.

TEST 10: MANUAL INTERVENTION

THE MANUAL INTERVENTION TEST IS A STANDALONE ROUTINE (NOT REALLY A "TEST")
THAT ALLOWS THE OPERATOR TO CHECK OUT VARIOUS ELEMENTS AND FUNCTIONS OF
THE SUBSYSTEM THAT CANNOT BE MANIPULATED BY THE PROGRAM ALONE. WHEN
THIS ROUTINE IS STARTED, IT FIRST PRINTS OUT A MENU OF SELECTABLE
SUBTESTS AND THEN WAITS FOR THE OPERATOR TO TYPE IN A SELECTION CODE.
THE ONLY WAYS TO EXIT THIS ROUTINE AND RETURN TO THE DIAGNOSTIC SUPERVISOR
ARE BY TYPING <CTRL-C> OR SELECTING CODE 6.
SELECTION CODES AND SUBROUTINES ARE:

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

TEST 11: CONFIGURATION TYPEOUT

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

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

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

3.0 MICROCODE REVSION LEVEL OF THE M7196,

4.0 NUMBER OF TAPE TRANSPORTS CONNECTED TO THE CONTROLLER,

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

TEST 12: SCOPE LOOPS

THIS IS A STANDALONE ROUTINE PROVIDING A NUMBER OF TIGHT "SCOPE
LOOPS" USEFUL FOR DEBUGGING BASIC REGISTER ACCESS PROBLEMS WITH
THE M7196 MODULE. THESE SCOPE LOOPS CAN BE USED WHEN THE NORMAL
"LOOP ON ERROR" OR "LOOP ON TEST (SUBTEST)" FACILITIES DON'T

SEEM TO ALLOW THE OPERATOR TO ZERO IN A PROBLEM IN THE EARLY TESTS (I.E, THE HARDWARE MAY NOT BE RESPONDING TO A REGISTER ACCESS, CAUSING A BUS ERROR TRAP, EVEN THOUGH THE DEVICE ADDRESS SELECTED BY THE PROGRAM MATCHES THE CONFIGURATION SET UP IN THE HARDWARE DIP SWITCHES). THE FOLLOWING MENU OF SCOPE LOOPS ARE AVAILABLE:

CODE	SCOPE LOOP
0	HELP. PRINT THIS MENU.
1	TSBA READ ACCESS
2	TSSR READ ACCESS
3	INITIALIZE (TSSR WRITE ACCESS)
4	TSDB HIGH BYTE WRITE ACCESS
5	TSDB LOW BYTE WRITE ACCESS
6	TSDB MAINTENANCE-MODE WORD WRITE ACCESS
7	TSDBX (TSSR HIGH BYTE) WRITE ACCESS (EXTENDED FEATURES SWITCH MUST BE ON TO USE SELECTION CODE 7)
8	EXIT (RETURN TO SUPERVISOR)

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

7.0 MAINTENANCE HISTORY

REVISION A - MARCH 1982

-

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

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

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


                                        POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
002000                                  HEADER  CVTSB,A,0,655.,0
002000                          L$NAME::                ;DIAGNOSTIC NAME
002000      103                         .ASCII /C/
002001      126                         .ASCII /V/
002002      124                         .ASCII /T/
002003      123                         .ASCII /S/
002004      102                         .ASCII /B/
002005      000                         .BYTE   0
002006      000                         .BYTE   0
002007      000                         .BYTE   0
002010                          L$REV::                 ;REVISION LEVEL
002010      101                         .ASCII  /A/
002011                          L$DEPO::                ;0
002011      060                         .ASCII  /0/
002012                          L$UNIT::                ;NUMBER OF UNITS
002012  000000                          .WORD   0
002014                          L$TIML::                ;LONGEST TEST TIME
002014  001217                          .WORD   655.
002016                          L$HPCP::                ;POINTER TO H.W. QUES.
002016  101342                          .WORD   L$HARD
002020                          L$SPCP::                ;POINTER TO S.W. QUES.
002020  101474                          .WORD   L$SOFT
002022                          L$HPTP::                ;PTR. TO DEF. H.W. PTABLE
002022  002156                          .WORD   L$HW
002024                          L$SPTP::                ;PTR. TO S.W. PTABLE
002024  002166                          .WORD   L$SW
002026                          L$LADP::                ;DIAG. END ADDRESS
002026  102004                          .WORD   L$LAST
002030                          L$STA::                 ;RESERVED FOR APT STATS
002030  000000                          .WORD   0
002032                          L$CO::
002032  000000                          .WORD   0
002034                          L$DTYP::                ;DIAGNOSTIC TYPE
002034  000000                          .WORD   0
002036                          L$APT::                 ;APT EXPANSION
002036  000000                          .WORD   0
002040                          L$DTP::                 ;PTR. TO DISPATCH TABLE
002040  002124                          .WORD   L$DISPATCH
```

```
002042
002042  000000                  L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
                                .WORD   0
002044
002044  000000                  L$ENVI::                ;FLAGS DESCRIBE HOW IT WAS SETUP
                                .WORD   0
002046
002046  000000                  L$EXP1::                ;EXPANSION WORD
                                .WORD   0
002050
002050  003                     L$MREV::                ;SVC REV AND EDIT #
                                .BYTE   C$REVISION
002051  003                     .BYTE   C$EDIT
002052
002052  000000                  L$EF::                  ;DIAG. EVENT FLAGS
                                .WORD   0
002054  000000                  .WORD   0
002056
002056  000000                  L$SPC::
                                .WORD   0
002060
002060  003402                  L$DEVP::                ; POINTER TO DEVICE TYPE LIST
                                .WORD   L$DVTYP
002062
002062  022664                  L$REPP::                ;PTR. TO REPORT CODE
                                .WORD   L$RPT
002064
002064  000000                  L$EXP4::
                                .WORD   0
002066
002066  000000                  L$EXP5::
                                .WORD   0
002070
002070  022352                  L$AUT::                 ;PTR. TO ADD UNIT CODE
                                .WORD   L$AU
002072
002072  022450                  L$DUT::                 ;PTR. TO DROP UNIT CODE
                                .WORD   L$DU
002074
002074  000000                  L$LUN::                 ;LUN FOR EXERCISERS TO FILL
                                .WORD   0
002076
002076  003410                  L$DESP::                ;POINTER TO DIAG. DESCRIPTION
                                .WORD   L$DESC
002100
002100  104035                  L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
                                EMT     E$LOAD
002102
002102  000000                  L$ETP::                 ;POINTER TO ERRTBL
                                .WORD   0
002104
002104  021556                  L$ICP::                 ;PTR. TO INIT CODE
                                .WORD   L$INIT
002106
002106  022636                  L$CCP::                 ;PTR. TO CLEAN-UP CODE
                                .WORD   L$CLEAN
002110
002110  022556                  L$ACP::                 ;PTR. TO AUTO CODE
                                .WORD   L$AUTO
002112
002112  021546                  L$PRT::                 ;PTR. TO PROTECT TABLE
                                .WORD   L$PROT
002114
002114  000000                  L$TEST::                ;TEST NUMBER
                                .WORD   0
002116
002116  000000                  L$DLY::                 ;DELAY COUNT
                                .WORD   0
002120
002120  000000                  L$HIME::                ;PTR. TO HIGH MEM
                                .WORD   0
```

```
                        .SBTTL  DISPATCH TABLE

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

002122                  DISPATCH 12
002122  000014          .WORD   12
002124          L$DISPATCH::
002124  023446          .WORD   T1
002126  024426          .WORD   T2
002130  026420          .WORD   T3
002132  031774          .WORD   T4
002134  034564          .WORD   T5
002136  040356          .WORD   T6
002140  050470          .WORD   T7
002142  051750          .WORD   T8
002144  062576          .WORD   T9
002146  066646          .WORD   T10
002150  074510          .WORD   T11
002152  077662          .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  172520                  .WORD   172520          ; 1ST (OF 2) REGISTERS.
002160  000224                  .WORD   224             ; INTERRUPT VECTOR
002162  000200                  .WORD   PRI04           ; INTERRUPT PRIORITY.
002164                          ENDHW
002164                  L10000:
```

```
                                .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
002176
```

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

002176                          BGNMOD  TSV3
002176                  TSV3::

                                .SBTTL  GLOBAL EQUATES SECTION

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

002176                          EQUALS          ; GET STANDARD EQUATES.
                        ;
                        ; BIT DIFINITIONS
                        ;
100000                  BIT15== 100000
040000                  BIT14== 40000
020000                  BIT13== 20000
010000                  BIT12== 10000
004000                  BIT11== 4000
002000                  BIT10== 2000
001000                  BIT09== 1000
000400                  BIT08== 400
000200                  BIT07== 200
000100                  BIT06== 100
000040                  BIT05== 40
000020                  BIT04== 20
000010                  BIT03== 10
000004                  BIT02== 4
000002                  BIT01== 2
000001                  BIT00== 1
                        ;
001000                  BIT9==  BIT09
000400                  BIT8==  BIT08
000200                  BIT7==  BIT07
000100                  BIT6==  BIT06
000040                  BIT5==  BIT05
000020                  BIT4==  BIT04
000010                  BIT3==  BIT03
000004                  BIT2==  BIT02
000002                  BIT1==  BIT01
000001                  BIT0==  BIT00
                        ;
                        ; EVENT FLAG DEFINITIONS
                        ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                        ;
000040                  EF.START==      32.                     ; START COMMAND WAS ISSUED
000037                  EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
000036                  EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
000035                  EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
000034                  EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
                        ;
                        ;
```

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

34
35 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
```

```
                        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                  HIADDR= BIT9!BIT8       ;EXTENDED ADDRESS BITS
000200                  SSR=    BIT7            ;SUB SYSTEM READY
000100                  OFL=    BIT6            ;OFF LINE BIT
000060                  FATERR= BIT4!BIT5       ;FATAL TERMINATION ERROR CODES
000016                  TERCLS= BIT3!BIT2!BIT1  ;TERMINATION CODES


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

100000                  XS0TMK= BIT15           ;TAPE MARK DETECTED
040000                  XS0RLS= BIT14           ;RECORD LENGTH SHORT
020000                  XS0LET= BIT13           ;LOGICAL END OF TAPE
010000                  XS0RLL= BIT12           ;RECORD LENGTH LONG
004000                  XS0WLE= BIT11           ;WRITE LOCK ERROR
002000                  XS0NEF= BIT10           ;NON EXECUTABLE FUNCTION
001000                  XS0ILC= BIT9            ;ILLEGAL COMMAND
000400                  XS0ILA= BIT8            ;ILLEGAL ADDRESS
000200                  XS0MOT= BIT7            ;TAPE IN MOTION
000100                  XS0ONL= BIT6            ;TRANSPORT ON LINE
000040                  XS0IE=  BIT5            ;INTERRUPT ENABLE
000020                  XS0VCK= BIT4            ;VOLUME CHECK BIT
000010                  XS0PED= BIT3            ;PHASE ENCODED DRIVE
000004                  XS0WLK= BIT2            ;WRITE LOCKED
000002                  XS0BOT= BIT1            ;BEGINNING OF TAPE
000001                  XS0EOT= BIT0            ;END OF TAPE


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

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

                        ;+
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
                        ;(XST2)
                        ;-
100000                  X2.OPM  = BIT15          ;OPERATION IN PROGRESS (TAPE MOVING)
040000                  X2.RCE  = BIT14          ;RAM CHECKSUM ERROR
035400                  X2.SPARE= BIT13+BIT12+BIT11+BIT9+BIT8   ;NOT USED BY TSV05 (ALWAYS=0)
002000                  X2.WCF  = BIT10          ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
000200                  X2.EXTF = BIT7           ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
000100                  X2.BUFE = BIT6           ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
000077                  X2.REV  = 000077         ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
000007                  X2.UNIT = BIT2+BIT1+BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.

                        ;+
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
                        ;(XST3)
                        ;-
177400                  X3.MDE  = 177400         ;MICRO-DIAGNOSTIC ERROR CODE
000200                  X3.SPARE= BIT7           ;NOT USED BY TSV05
000100                  X3.OPI  = BIT6           ;OPERATION INCOMPLETE
000040                  X3.REV  = BIT5           ;REVERSE
000020                  X3.TRF  = BIT4           ;TRANSPORT RESPONSE FAILURE
000010                  X3.DCK  = BIT3           ;DENSITY CHECK
000006                  X3.MBZ  =BIT2+BIT1       ;NOT USED ALWAYS 0
000001                  X3.RIB  = BIT0           ;REVERSE INTO BOT

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


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

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

                        ;+
                        ;
                        ;DEVICE REGISTER OFFSETS
```

```
                        ;
                        ;-

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

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
000002          WC.IFEN         = BIT1          ;IFEN   - FORMATTER ENABLE
000001          WC.IGO          = BIT0          ;GO

                ;+
                ;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
```

```
                000400          S1.IHER         = BIT8          ;                       IHER H
                000200          S0.ISPEED       = BIT7          ;WORD #8 BYTE 0 ISPEED H
                000100          S0.IRDY         = BIT6          ;                       IRDY L
                000040          S0.IONL         = BIT5          ;                       IONL L
                000020          S0.ILDP         = BIT4          ;                       ILDP L
                000010          S0.IDBY         = BIT3          ;                       IDBY L
                000004          S0.IRWD         = BIT2          ;                       IRWD L
                000002          S0.IFBY         = BIT1          ;                       IFBY L
                000001          S0.IFPT         = BIT0          ;                       IFPT L
```

```
                        .SBTTL  SPECIAL MACROS AND OPDEFS.


                ;+
                ;SAVE GENERAL REGS 1 TO 5
                ;-

                        .MACRO  SAVREG
                        JSR     R5,REGSAV
                        .ENDM

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

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

                ;+
```

```
                                ;MACRO TO PERFORM XOR
                                ;-

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

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

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

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   PR104   ;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
002304  000000          RCVHIADD::      .WORD   0       ;RECEIVED BUFFER HIGH ADDRESS
002306  000000          RCVLOADD::      .WORD   0       ;RECEIVED BUFFER LOW ADDRESS
002310  000000          COUNT::         .WORD   0       ;TEST COUNT PATTERN
002312  000000          DATA::          .WORD   0       ;TEST DATA
002314  000000          TSTFLAG::       .WORD   0       ;TEST FLAG WORD
002316  000000          TSTPTR::        .WORD   0       ;TSTBLK POINTER
002320  000000          PRMNO::         .WORD   0       ;PRINT ROUTINE TEMP
002322                  EXPMSG::        .BLKB   100.    ;EXPECTED MESSAGE BUFFER DATA
002466                  RECMSG::        .BLKB   100.    ;RECEIVED MESSAGE BUFFER DATA
002632                  TMPBFR::        .BLKB   80.     ;TEMPORARY STORAGE FOR PRINT
```

```
                                .SBTTL  TSTBLK  - TEST DATA TABLE

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

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

```
                                        .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.
003376  000000                  ERTABE:         .WORD   0

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

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



                                ;+
                                ;NAMES OF DEVICES SUPPORTED
                                ;-

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/
```

```
644 004060  045  101  040            .ASCIZ  /%A  TSBA,TSSR REC'D: %06%A,%06/
645 004117  045  116  045  FUSI:     .ASCII  /%N%A/
646 004123  040  040  125  USI:      .ASCIZ  /  UNEXPECTED INTERRUPT/
647 004152  040  040  111  NSI:      .ASCIZ  /  INTERRUPT EXPECTED, NOT RECEIVED/
648 004215  045  116  045  FNOINTR:          .ASCII  /%N%A/
649 004221  040  040  116  NOINTR:   .ASCIZ  /  NO INTERRUPT WAS GENERATED/
650 004256  040  040  111  IFAULT:   .ASCIZ  /  INTERRUPT FAULT/
651 004300  045  101  040  INTX:     .ASCIZ  /%A  CPU PC: %06%A  TSBA: %06/
652 004335  040  040  042  NOINIT:   .ASCIZ  /  "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
653 004407  040  040  042  NSINIT:   .ASCIZ  /  "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
654 004457  040  040  042  BRINIT:   .ASCIZ  /  "BUS-RESET" DIDN'T INITIALIZE THE DPU/
655
656 004527  000            NUL:      .ASCIZ  //
657 004530  045  116  000  NULCR:    .ASCIZ  /%N/
658 004533  045  101  040  EXPGOT:   .ASCIZ  /%A EXP'D: %06%A, REC'D: %06/
659 004567  045  116  045  EXPGT2:   .ASCIZ  /%N%A EXP'D: %06%A, %06%N%A REC'D: %0%A, %06/
660 004643  045  101  040  DUAD12:   .ASCIZ  /%A REG(W) WRITTEN TO: %06%A REG(R) READ; EXP'D: %06%A, REC'D: %06/
661 004745  122  101  115  PKTRAM::          .ASCIZ  'RAM Contents Do Not Match Packet Sent'
662 005013  040  040  103  SCME:     .ASCIZ  /  CONFIG DOESN'T MATCH MFG. MASTER/
663 005056  127  122  111  WRTMSG:   .ASCIZ  'WRITE CHARACTERISTICS Failed'
664 005113  124  123  123  WRTERR:   .ASCIZ  'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
665 005206  124  123  123  RDERR:    .ASCIZ  'TSSR Incorrect After READ Command, More Bits Set Than SSR'
666 005300  106  101  124  SCHERR:   .ASCIZ  'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
667 005372  105  122  122  RETERR:   .ASCIZ  'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
668 005460  045  116  045  NOMEM:    .ASCIZ  '%N%A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****%N'
669 005554  045  116  045  M8186:    .ASCIZ  '%N%A ********** 11/23A SYSTEM **********%N'
670 005645  045  116  045  M8189:    .ASCIZ  '%N%A ********** 11/23B SYSTEM **********%N'
671                                    .EVEN
672
673
674
```

```
                        .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$PNTX
005756  062706  000006          ADD     #6,SP
005762  004737  005770          JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
005766                      ENDMSG
                        L10002:
005766  104423                TRAP    C$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$:     PRINTX  #NULCR          ; PRINT A BLANK LINE
006002  012746  004530          MOV     #NULCR,-(SP)
006006  012746  000001          MOV     #1,-(SP)
006012  010600                  MOV     SP,R0
006014  104415                  TRAP    C$PNTX
006016  062706  000004          ADD     #4,SP
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)
```

```
                006170  012746  000002                  MOV     #2,-(SP)
                006174  010600                          MOV     SP,R0
                006176  104415                          TRAP    C$PNTX
                006200  062706  000006                  ADD     #6,SP
    746
    747         006204  010403              20$:        MOV     R4,R3           ;GET THE TSSR CONTENTS
    748         006206  042703  177761                  BIC     #^CTERCLS,R3    ;CLEAR ALL BUT TERMINATION
    749         006212  016303  006676                  MOV     TCOCOD(R3),R3   ;GET THE TERMINATION CODE MEANING
    750         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$PNTX
                006234  062706  000006                  ADD     #6,SP
    751         006240  010403                          MOV     R4,R3           ;TSSR CONTENTS AGAIN
    752         006242  042703  177717                  BIC     #^CFATERR,R3    ;CLEAR ALL BUT FATAL TERMINATION
    753         006246  001416                          BEQ     25$             ;DON'T PRINT IF ZERO
    754         006250  006203                          ASR     R3
    755         006252  006203                          ASR     R3
    756         006254  006203                          ASR     R3              ;ALINE TERMINATION CODE FOR INDEX
    757         006256  016303  007236                  MOV     TSFCOD(R3),R3   ;GET THE FATAL TERMINATION CODE
    758         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$PNTX
                006300  062706  000006                  ADD     #6,SP
    759         006304  042704  176377          25$:    BIC     #^CHIADDR,R4    ;CLEAR ALL BUT EXTENDED ADDRESS
    760         006310  001411                          BEQ     30$             ;DON'T PRINT IF ZERO
    761         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$PNTX
                006330  062706  000006                  ADD     #6,SP
    762         006334  013703  002200          30$:    MOV     EPRTSW,R3               ;PRINT MEAASGE BUFFER ADDRESS
    763         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
    764         006356  000207                          RTS     PC              ;RETURN TO CALLER
    765
    771         006360                      EPRT2:
    772         006360      045     116     045 EPRT1:  .ASCIZ   '%N%A *****REPLACE M7196*****'
    773
    783         006415      045     116     045 TSSRFOR:        .ASCIZ  '%N%A TSSR = %O6'
    784         006435      045     116     045 TEXASC:         .ASCIZ  '%N%A Extended Address Bits = %O6'
    785         006476      045     116     045 TCOASC:         .ASCIZ  '%N%A Termination Class Code = %T'
    786         006537      045     116     045 TFCASC:         .ASCIZ  '%N%A Fatal Termination Class Code = %T'
    787         006606      045     116     045 TSSDEF:         .ASCIZ  '%N%A TSSR Bits Set: %T'
    788         006635      045     116     045 AMBTSSR:        .ASCIZ  '%N%A TSSR Contents Are Ambiguous'
    789                                             .EVEN
```

```
790 006676  006716  006741  006767  TCOCOD: .WORD   1$,2$,3$,4$,5$,6$,7$,8$
791 006716    116     157     162  1$:     .ASCIZ  'Normal Termination'
792 006741    124     145     162  2$:     .ASCIZ  'Termination Condition'
793 006767    124     141     160  3$:     .ASCIZ  'Tape Status Alert'
794 007011    106     165     156  4$:     .ASCIZ  'Function Reject'
795 007031    122     145     143  5$:     .ASCIZ  'Recoverable Error - Tape Position One Record Down'
796 007113    122     145     143  6$:     .ASCIZ  'Recoverable Error - Tape Was Not Moved'
797 007162    125     156     162  7$:     .ASCIZ  'Unrecoverable Error'
798 007206    106     141     164  8$:     .ASCIZ  'Fatal Controller Error'
799                                          .EVEN
800
801 007236  007246  007302  007313  TSFCOD: .WORD   1$,2$,3$,4$
802 007246    111     156     164  1$:     .ASCIZ  'Internal Diagnostic Failure'
803 007302    122     145     163  2$:     .ASCIZ  'Reserved'
804 007313    102     165     163  3$:     .ASCIZ  'Bus Interface or Sanity Check Error'
805 007357    122     145     163  4$:     .ASCIZ  'Reserved'
806                                          .EVEN
```

```
                        .SBTTL  PRIPKT  - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

                        ;+
                        ;THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
                        ;THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
                        ;
                        ;INPUT:
                        ;
                        ;       R0      NUMBER OF WORDS IN PACKET
                        ;       R3      HIGH ORDER COMMAND PACKET ADDRESS
                        ;       R4      ADDRESS OF COMMAND PACKET
                        ;
                        ;       NOTE:   R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
                        ;-

007370                  PRIPKT::
007370                          SAVREG                          ;SAVE THE REGISTERS
007374  010005                  MOV     R0,R5                   ;SAVE NO. OF WORDS IN PACKET
007376  005737  003134          TST     KTENABLE                ;ABOVE 28K UNDER TEST?
007402  001001                  BNE     10$                     ;BR IF YES
007404  005003                  CLR     R3                      ;SET HIGH ORDER ADDRESS TO 0
007406  010301          10$:    MOV     R3,R1                   ;COPY HIGH ORDER ADDRESS
007410  010400                  MOV     R4,R0                   ;GET LOWER ADDRESS
007412  006100                  ROL     R0                      ;SHIFT BIT 15 INTO C BIT
007414  006101                  ROL     R1                      ;AND INTO HIGH ORDER.
007416                          PRINTB  #PKTADD,R1,R4           ;PRINT PACKET ADDRESS
007416  010446                  MOV     R4,-(SP)
007420  010146                  MOV     R1,-(SP)
007422  012746  007554          MOV     #PKTADD,-(SP)
007426  012746  000003          MOV     #3,-(SP)
007432  010600                  MOV     SP,R0
007434  104414                  TRAP    C$PNTB
007436  062706  000010          ADD     #10,SP
007442  010300          15$:    MOV     R3,R0                   ;GET HIGH ORDER ADDRESS
007444  001404                  BEQ     20$                     ;BR IF NOT ABOVE 28K.
007446  010401                  MOV     R4,R1                   ;GET LOW ORDER ADDRESS
007450  004737  017316          JSR     PC,SETMAP               ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
007454  010004                  MOV     R0,R4                   ;GET RETURNED PAR6 ADDRESS BIAS
007456  005001          20$:    CLR     R1                      ;SAVE WORD NUMBER
007460  012402          25$:    MOV     (R4)+,R2                ;GET PACKET CONTENTS
007462                          PRINTB  #PKTFRM,R1,R2           ;PRINT THE DATA
007462  010246                  MOV     R2,-(SP)
007464  010146                  MOV     R1,-(SP)
007466  012746  007516          MOV     #PKTFRM,-(SP)
007472  012746  000003          MOV     #3,-(SP)
007476  010600                  MOV     SP,R0
007500  104414                  TRAP    C$PNTB
007502  062706  000010          ADD     #10,SP
007506  005201                  INC     R1                      ;NEXT WORD NUMBER
007510  020105                  CMP     R1,R5                   ;DONE ALL PACKET WORDS?
007512  002762                  BLT     25$                     ;LOOP TILL ALL DONE
007514  000207                  RTS     PC                      ;RETURN

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

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

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

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

007674      045     116     045  XORBFOR:       .ASCIZ  '%N%A EXPD: %03%A RECV: %03%A XOR: %03'
                                        .EVEN
```

```
                        .SBTTL  PRIXOR  - PRINT EXPD, RECV AND XOR

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

007742          PRIXOR::
007742                  SAVREG                          ;SAVE THE REGISTERS
007746  010203          MOV     R2,R3                   ;EXPECTED DATA
007750                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007760                  PRINTB  #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
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.
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                                ;
                                ;INPUTS:
                                ;
                                ;       R4      RAM ADDRESS
                                ;
                                ;-
010066                          PRIRAM:
010066                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
010072                                  PRINTB  #RAMFOR,R4              ;PRINT RAM ADDRESS IN ERROR
010072  010446                          MOV     R4,-(SP)
010074  012746  010116                  MOV     #RAMFOR,-(SP)
010100  012746  000002                  MOV     #2,-(SP)
010104  010600                          MOV     SP,R0
010106  104414                          TRAP    C$PNTB
010110  062706  000006                  ADD     #6,SP
010114  000207                          RTS     PC                      ;RETURN

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


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

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

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


                              .SBTTL  PRITADD - PRINT MEMORY TEST ADDRESS
                        ;+
                        ;
                        ;PRINT MEMORY ADDRESS
                        ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                        ;
                        ; IMPLICIT INPUTS
                        ;
                        ;       ERRHI   - HIGH ORDER ADDRESS
                        ;       ERRLO   - LOW ORDER ADDRESS
                        ;
                        ;-
010274                  PRITADD:
010274                          SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
010300  013702  002236          MOV     ERRHI,R2                ;GET HIGH ADDRESSS
010304  013701  002240          MOV     ERRLO,R1                ;GET LOW ADDRESS
                                ;MOV    R1,R2                   ;COPY LOW ADDRESS
                                ;ROL    R1                      ;SHIFT BIT 15 TO C BIT
                                ;ROL    R0                      ;SHIFT INTO HIGH ORDER
010310                          PRINTB  #PRIT0,R1               ;PRINT MEMORY ADDRESS LOW IN ERROR
010310  010146                  MOV     R1,-(SP)
010312  012746  010356          MOV     #PRIT0,-(SP)
010316  012746  000002          MOV     #2,-(SP)
010322  010600                  MOV     SP,R0
010324  104414                  TRAP    C$PNTB
010326  062706  000006          ADD     #6,SP
010332                          PRINTB  #PRIT1,R2               ;PRINT MEMORY ADDRESS HIGH IN ERROR
010332  010246                  MOV     R2,-(SP)
010334  012746  010421          MOV     #PRIT1,-(SP)
010340  012746  000002          MOV     #2,-(SP)
010344  010600                  MOV     SP,R0
010346  104414                  TRAP    C$PNTB
010350  062706  000006          ADD     #6,SP
010354  000207                  RTS     PC                      ;RETURN
```

```
 1004
 1005 010356     045     116     045  PRIT0:  .ASCIZ  '%N%A MEMORY TEST ADDRESS LOW = %06'
 1006 010421     045     116     045  PRIT1:  .ASCIZ  '%N%A MEMORY TEST ADDRESS HIGH = %06'
 1007                                         .EVEN
 1008
 1009
 1010
```

```
                        .SBTTL  SPACE   - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

                        ;+
                        ;
                        ;ROUTINE TO ISSUE A SPACE RECORDS
                        ;COMMAND (FORWARD OR REVERSE)
                        ;
                        ;INPUT:
                        ;
                        ;       R3      NUMBER OF RECORDS TO BE SPACED OVER
                        ;               BIT15 CONTROLS DIRECTION
                        ;               BIT15 = 0 IS FORWARD
                        ;               BIT15 = 1 IS REVERSE
                        ;       R5      FIRST DEVICE UNIBUS ADDRESS
                        ;
                        ;       REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
                        ;
                        ;OUTPUT:
                        ;
                        ;       CARRY   SET - SPACE RECORDS COMMAND OK
                        ;               CLR - SPACE RECORDS FAILED
                        ;
                        ;
                        ;       R0      THE CONTENTS OF R4 IS MOVED TO R0
                        ;
                        ;IMPLICIT OUTPUT:
                        ;
                        ;       TAPE HAS BEEN MOVED
                        ;
                        ;SIDE EFFECTS:
                        ;
                        ;
                        ;-
                        ;
010466                  SPACE::
010466                          SAVREG                          ;SAVE THE GENERAL REGISTERS
010472  012737  000764  010660          MOV     #500.,SDELAY    ;SET UP DELAY
010500  012737  140010  010650          MOV     #140010,80$     ;SET UP COMMAND, SPACE FORWARD
010506  005703                          TST     R3              ;CHECK FOR DIRECTION
010510  100403                          BMI     5$              ;BR, IF REVERSE INDICATED
010512  010337  010652                  MOV     R3,90$          ;LOAD UP NUMBER OF RECORDS TO SPACE
010516  000407                          BR      10$             ;GO DO COMMAND
010520  042703  100000          5$:     BIC     #BIT15,R3       ;CLEAR DIRECTION BIT
010524  010337  010652                  MOV     R3,90$          ;LOAD UP NUMBER OF RECORDS TO SPACE
010530  052737  000400  010650          BIS     #BIT8,80$       ;SET REVERSE BIT IN COMMAND PACKET
010536  012704  010650          10$:    MOV     #80$,R4         ;SET UP R4 WITH PACKET ADDRESS
010542  010465  000000                  MOV     R4,TSDB(R5)     ;SEND OUT COMMAND
010546  004737  016250          15$:    JSR     PC,WAITF        ;WAIT FOR SSR
010552  103420                          BCS     20$             ;BR, IF SSR IS SET AND OK
010554                                  DELAY   250     ;DELAY ABOUT .25 SECONDS
010554  012727  000250                  MOV     #250,(PC)+
010560  000000                          .WORD   0
010562  013727  002116                  MOV     L$DLY,(PC)+
010566  000000                          .WORD   0
010570  005367  177772                  DEC     -6(PC)
010574  001375                          BNE     .-4
```

```
        010576  005367  177756                  DEC     -22(PC)
        010602  001367                          BNE     .-20
  1063  010604  005337  010660                  DEC     $DELAY          ;BUMP DELAY COUNTER DOWN
  1064  010610  001356                          BNE     15$             ;BR, IF MORE DELAY
  1065  010612  000411                          BR      60$             ;BR IF TROUBLE CARRY = CLEAR
  1066  010614  016501  000002          20$:    MOV     TSSR(R5),R1     ;READ TSSR
  1067  010620  012702  000200                  MOV     #SSR,R2         ;SET UP EXPECTED
  1068  010624  020201                  25$:    CMP     R2,R1           ;ARE THEY OK
  1069  010626  001401                          BEQ     40$             ;BR, IF EQUAL = OK
  1070  010630  000402                          BR      60$             ;TROUBLE EXIT
  1071  010632  000261                  40$:    SEC                     ;SET CARRY NO TROUBLE
  1072  010634  000401                          BR      70$             ;EXIT
  1073  010636  000241                  60$:    CLC                     ;CARRY CLEAR = ERROR
  1074  010640                          70$:
  1075  010640  010400                          MOV     R4,R0           ;PASS PACKET ADDRESS
  1076  010642  000207                          RTS     PC              ;RETURN
```

```
 1078                                   ;
 1079                                   ;
 1080                                   ;
 1081                                   ;PACKET FOR SPACE COMMAND
 1082                                   ;
 1084            010650                         .=<.+10>&177770
 1086                                   ;
 1087                                   ;COMMAND WORD
 1088 010650    000000                 80$:    .WORD
 1089                                   ;NUMBER OF RECORDS TO BE SPACED OVER WORD
 1090 010652    000000                 90$:    .WORD
 1091 010654    000000                         .WORD
 1092 010656    000000                         .WORD
 1093 010660    000000                 SDELAY: .WORD   0                       ;DELAY COUNTER
 1094                                           .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
010666  005037  002230          CLR     BENBSW          ;CLEAR BUFFER ENABLE SWITCH
010672  005037  002226          CLR     EXTFEA                  ;CLEAR EXTENDED FEATURES SW SWITCH
010676  010465  000000  10$:    MOV     R4,TSDB(R5)             ;SEND OUT COMMAND
010702  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR
010706  103401                  BCS     20$                     ;BR, IF SSR IS SET AND OK
010710  000435                  BR      60$                     ;BR IF TROUBLE CARRY = CLEAR
010712  016501  000002  20$:    MOV     TSSR(R5),R1             ;READ TSSR
010716  012702  000200          MOV     #SSR,R2                 ;SET UP EXPECTED
010722  032701  000100          BIT     #OFL,R1                 ;WAS OFF LINE SET IN TSSR
010726  001402                  BEQ     25$                     ;BR, IF NO OFL SET
010730  052702  000100          BIS     #OFL,R2                 ;MAKE THEM LOOK ALIKE
010734  020201          25$:    CMP     R2,R1                   ;ARE THEY OK
010736  001401                  BEQ     40$                     ;BR, IF EQUAL = OK
010740  000421                  BR      60$                     ;TROUBLE EXIT
010742  062704  000010  40$:    ADD     #8.,R4                  ;POINT TO WRT CHARA DATA PACKET
010746  011403                  MOV     (R4),R3                 ;GET ADDRESS OF MESSAGE BUFFER
010750  032763  000200  000012  BIT     #X2.EXTF,XST2(R3)       ;EXTENDED FEATURES BIT SET?
010756  001402                  BEQ     45$                     ;BR IF NO
010760  005237  002226          INC     EXTFEA                  ;SET EXTENDED FEATURES SW SWITCH
010764                  45$:
010764  032763  000100  000012  BIT     #X2.BUFE,XST2(R3)                       ;BUFFER ENABLE SWITCH SET
010772  001402                  BEQ     50$                     ;BR, IF SWITCH NOT SET
010774  005237  002230          INC     BENBSW                  ;SET SOFTWARE SWITCH FOR ENABLED
```

```
1153 011000                          50$:
1154 011000  000261                          SEC                     ;SET CARRY NO TROUBLE
1155 011002  000401                          BR      70$             ;EXIT
1156 011004  000241                  60$:    CLC                     ;CARRY CLEAR = ERROR
1157 011006  016500  000002          70$:    MOV     TSSR(R5),R0     ;RETURN TSSR CONTENTS
1158 011012  000207                          RTS     PC              ;RETURN
1159
1160
```

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

                ;+
                ;
                ;THIS ROUTINE WILL REWIND THE SELECTED TAPE.
                ;
                ;       CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
                ;                TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
                ;                SSR TO SET IN THE TSSR
                ;
                ;
                ;CALLING SEQUENCE:
                ;
                ;       DO A SOFT INIT
                ;       DO A WRITE CHARACTERISTICS
                ;       JSR     PC,REWIND
                ;
                ;INPUT:
                ;
                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                ;
                ;
                ;OUTPUT
                ;
                ;       R0      THE CONTENTS OF R4 IS PASSED TO R0
                ;
                ;
                ;-
                ;
011014          REWIND::
011014                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011020  012704  011110          MOV     #RWPACK,R4      ;GET PACKET ADDRESS
011024  010465  000000          MOV     R4,TSDB(R5)     ;SEND PACKET ADDRESS TO EXECUTE
011030  012703  000550          MOV     #360.,R3        ;ENOUGH TIME FOR 2400' REEL TO REWIND
011034  004737  016250  10$:    JSR     PC,WAITF        ;WAIT FOR SSR TO SET
011040  103417                  BCS     20$             ;LEAVE WHEN SSR IS SET
011042                          DELAY   250.            ;WAIT FOR .25 SECONDS
011042  012727  000372          MOV     #250.,(PC)+
011046  000000                  .WORD   0
011050  013727  002116          MOV     L$DLY,(PC)+
011054  000000                  .WORD   0
011056  005367  177772          DEC     -6(PC)
011062  001375                  BNE     .-4
011064  005367  177756          DEC     -22(PC)
011070  001367                  BNE     .-20
011072  005303                  DEC     R3              ;BUMP COUNTER DOWN
011074  001357                  BNE     10$             ;KEEP GOING
011076  000241                  CLC                     ;CLEAR CARRY TO SET ERROR
011100  010400          20$:    MOV     R4,R0           ;PASS THE PACKET ADDRESS
011102  000207                  RTS     PC              ;RETURN


                        .=<.+10>&177770
011110          RWPACK:
011110  102010                  .WORD   102010          ;POSTION COMMAND (REWIND)
011112  000000                  .WORD   0               ;NOT USED
```

```
1213
1214
1215
```

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

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

011114                  CKRAM::
011114                          SAVREG                          ;SAVE THE GENERAL REGISTERS
011120  012701  002242          MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
011124  012702  000201          MOV     #RMPKTBEG,R2            ;BYTE ADDRESS OF FIRST RAM DATA
011130  005003                  CLR     R3                      ;CLEAR THE ERROR FLAG
011132  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR
011136  112765  000000  000000  MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
011144  004737  016336  10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011150  010265  000000          MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
011154  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011160  116511  000000          MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
011164  122124                  CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
011166  001401                  BEQ     20$                     ;BRANCH IF OK
011170  005203                  INC     R3                      ;SET ERROR FLAG
011172  005202          20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
011174  020227  000210          CMP     R2,#RMPKTEND            ;REACHED END YET ?
011200  003761                  BLE     10$                     ;BRANCH TILL ALL READ
011202  005703                  TST     R3                      ;WAS AN ERROR FOUND ?
011204  001402                  BEQ     30$                     ;BRANCH IF NOT
011206  000241                  CLC                             ;CLEAR CARRY TO SHOW ERROR
011210  000401                  BR      50$                     ;AND EXIT
011212  000261          30$:    SEC                             ;SHOW GOOD COMPARE
011214  012737  000010  002302  50$:    MOV     #8.,RAMSIZ      ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
011222  000207                  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
                ;
                ;-

                CKRAM2::
                        SAVREG                          ;SAVE THE GENERAL REGISTERS
                        MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
                        MOV     #RMCHBEG,R2             ;BYTE ADDRESS OF FIRST RAM DATA
                        CLR     R3                      ;CLEAR THE ERROR FLAG
                        JSR     PC,CHKTSSR              ;WAIT FOR SSR
                        MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
                10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
                        MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
                        JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
                        MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
                        CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
                        BEQ     20$                     ;BRANCH IF OK
                        INC     R3                      ;SET ERROR FLAG
                20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
                        MOV     #8.,RAMSIZ              ;ASSUME EXTFEA NOT SET
                        TST     EXTFEA                  ;IS THE SOFTWARE EXTENDED FEATURES SET
                        BEQ     25$                     ;BR, IF NOT SET
                        MOV     #10.,RAMSIZ             ;SET RAMSIZ FOR EXTEND FEATURES
                        CMP     R2,#RMCHEND             ;AT END OF EXTENDED BUFFER
                        BLE     10$                     ;BR, IF NOT AT END YET
                        BR      27$                     ;AT END BRANCH
                25$:    CMP     R2,#RMCHEND-2           ;REACHED END YET ?
                        BLE     10$                     ;BRANCH TILL ALL READ
                27$:    TST     R3                      ;WAS AN ERROR FOUND ?
                        BEQ     30$                     ;BRANCH IF NOT
                        CLC                             ;CLEAR CARRY TO SHOW ERROR
                        BR      50$                     ;AND EXIT
                30$:    SEC                             ;SHOW GOOD COMPARE
```

```
 1329 011356  000207                50$:    RTS     PC                      ;RETURN
 1330
```

```
                                        .SBTTL  CKMSG   - COMPARE WRITE CHAR. MESSAGE BUFFERS
                                ;+
                                ;
                                ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
                                ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
                                ;ERROR PRINT ROUTINES.
                                ;
                                ;INPUT:
                                ;
                                ;       R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
                                ;       R2      EXPD MESSAGE BUFFER ADDRESS
                                ;OUTPUT:
                                ;
                                ;       CARRY   SET - MESSAGE BUFFERS MATCH
                                ;               CLR -MESSAGE BUFFERS DON'T MATCH
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       EXPMSG          BUFFER IS SET TO EXPD DATA
                                ;       RECMSG          BUFFER IS SET TO RECV DATA
                                ;       RCVHIADD        SET TO HIGH ORDER ADDRESS OF RECV
                                ;       RCVLOADD        SET TO LOW ORDER ADDRESS OF RECV
                                ;
                                ;-
011360                          CKMSG::
011360                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011364  010037  002304                  MOV     R0,RCVHIADD             ;SAVE RECV HIGH ADDRESS
011370  010137  002306                  MOV     R1,RCVLOAD              ;SAVE RECV LOW ADDRESS
011374  005737  003134                  TST     KTENABLE                ;TESTING ABOVE 28K?
011400  001403                          BEQ     10$                     ;BR IF NO
011402  004737  017316                  JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
011406  010001                          MOV     R0,R1                   ;GET RETURNED ADDRESS BIASED TO PAR6
011410  005004                  10$:    CLR     R4                      ;WORD IN BUFFER
011412  005003                          CLR     R3                      ;CLEAR ERROR SEEN FLAG
011414  010205                          MOV     R2,R5                   ;GET EXPD BUFFER ADDRESS
011416  011264  002322          15$:    MOV     (R2),EXPMSG(R4)         ;SAVE EXPD FOR ERROR REPORT
011422  011164  002466                  MOV     (R1),RECMSG(R4)         ;SAVE RECV FOR ERROR REPORT
011426  022221                          CMP     (R2)+,(R1)+             ;EXPD EQUAL RECV?
011430  001401                          BEQ     25$                     ;BR IF YES
011432  005203                          INC     R3                      ;SET ERROR SEEN FLAG
011434  062704  000002          25$:    ADD     #2,R4                   ;POINT TO NEXT WORD ADDRESS
011440  020427  000014                  CMP     R4,#14                  ;DONE FIRST 7 WORDS?
011444  003764                          BLE     15$                     ;BR IF NO
011446  032765  000200  000012          BIT     #X2.EXTF,XST2(R5);IS EXTENDED FEATURES SET IN EXPD?
011454  001403                          BEQ     50$                     ;BR IF NO
011456  020427  000016                  CMP     R4,#16                  ;DONE EXTENDED FEATURES WORD?
011462  003755                          BLE     15$                     ;BR IF NO
011464  005703                  50$:    TST     R3                      ;ANY ERRORS SEEN?
011466  001402                          BEQ     55$                     ;BR IF NO
011470  000241                          CLC                             ;SET FAILURE
011472  000401                          BR      60$                     ;
011474  000261                  55$:    SEC                             ;SET SUCCESS
011476  000207                  60$:    RTS     PC                      ;RETURN
```

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

```
1440 011622  000241                        CLC                     ;SET FAILURE
1441 011624  000401                        BR      60$             ;
1442 011626  000261                55$:     SEC                     ;SET SUCCESS
1443 011630  000207                60$:     RTS     PC              ;RETURN
1444
1445 011632     120     122     117  DEBUGMSG:       .ASCIZ  'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-';@@D
1446 011722     045     116     045  FERCM:  .ASCII  /%N%A  ***/
1447 011733     040     040     124  ERCM:   .ASCIZ  /  TSSR ERROR CODE REC'D = /
1448 011766     056     056     056  SIMSG:  .ASCIZ  /.... AFTER DOING SOFT INIT/
1449 012021     124     105     123  TINERR: .ASCIZ  /TEST: .../
1450                                  .EVEN
```

```
1452
1453
1454                                    ;+
1455                                    ;
1456                                    ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
1457                                    ;
1458                                    ;INPUT:
1459                                    ;
1460                                    ;       R1      CONTENTS OF TSSR AT ERROR
1461                                    ;
1462                                    ;SIDE EFFECTS:
1463                                    ;
1464                                    ;       EXECUTES DROP UNIT TO CEASE TESTING
1465                                    ;
1466                                    ;-
1467
1468 012034                                        BGNMSG  SFIMSG
     012034                             SFIMSG::
1469 012034  004737  006024                        JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1470 012040  004737  017202                        JSR     PC,CKDROP       ;DROP UNIT, IF ALLOWED
1471 012044                                        ENDMSG
     012044                             L10003:
     012044  104423                             TRAP    C$MSG
1472
1473                                    ;+
1474                                    ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1475                                    ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
1476                                    ;
1477                                    ;INPUTS:
1478                                    ;
1479                                    ;       R1      TSSR CONTENTS
1480                                    ;       R4      ADDRESS OF COMMAND PACKET
1481                                    ;
1482                                    ;-
1483
1484 012046                                        BGNMSG  PKTSSR
     012046                             PKTSSR::
1485 012046  004737  006024                        JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1486 012052  012700  000004                        MOV     #4,R0           ;NO. OF WORDS IN PACKET
1487 012056  004737  007370                        JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
1488 012062                                        ENDMSG
     012062                             L10004:
     012062  104423                             TRAP    C$MSG
1489
1490                                    ;+
1491                                    ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1492                                    ;TSSR AND A GET STATUS COMMAND PACKET.
1493                                    ;
1494                                    ;INPUTS:
1495                                    ;
1496                                    ;       R1      TSSR CONTENTS
1497                                    ;       R4      ADDRESS OF COMMAND PACKET
1498                                    ;
1499                                    ;-
1500
1501 012064                                        BGNMSG  PKTGETS
     012064                             PKTGETS::
```

```
1502 012064  004737  006024                    JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1503 012070  012700  000002                    MOV     #2,R0           ;NO. OF WORDS IN GET STATUS PACKET
1504 012074  004737  007370                    JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
1505 012100                                    ENDMSG
     012100                              L10005:
     012100  104423                            TRAP    C$MSG
1506
1507
1508                                        ;+
1509                                        ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
1510                                        ;
1511                                        ;INPUTS:
1512                                        ;
1513                                        ;       R1      TSSR CONTENTS
1514                                        ;       R4      ADDRESS OF COMMAND PACKET
1515                                        ;-
1516
1517 012102                                    BGNMSG  SFFMSG
     012102                              SFFMSG::
1518 012102  004737  006024                    JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1519 012106                                    ENDMSG
     012106                              L10006:
     012106  104423                            TRAP    C$MSG
1520
1521
1522                                            .SBTTL  PKTMES  - PRINT TSSR AND MESSAGE BUFFER
1523                                        ;+
1524                                        ;
1525                                        ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
1526                                        ;BUFFER FOR ERROR REPORTS
1527                                        ;
1528                                        ;INPUTS:
1529                                        ;
1530                                        ;       R1      CONTENTS OF TSSR
1531                                        ;       R2      LOW ORDER MESSAGE BUFFER
1532                                        ;       R3      HIGH ORDER MESSAGE BUFFER ADDRESS
1533                                        ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
1534                                        ;-
1535 012110                                    BGNMSG  PKTMES
     012110                              PKTMES::
1536 012110  004737  006024                    JSR     PC,PRITSSR              ;PRINT CONTENTS OF TSSR
1537 012114  010200                            MOV     R2,R0                   ;LOW ORDER ADDRESS
1538 012116  010301                            MOV     R3,R1                   ;HIGH ORDER ADDRESS
1539 012120  004737  014242                    JSR     PC,PRMESS               ;PRINT THE MESSAGE BUFFER
1540 012124                                    ENDMSG
     012124                              L10007:
     012124  104423                            TRAP    C$MSG
1541
```

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

                                        .SBTTL  MSGEXP - PRINT WRITE CHAR. EXPD-RECV MESSAGE BUFFERS
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
                                        BGNMSG  MSGEXP
012144                          MSGEXP::
012144  012700  000007                  MOV     #7,R0           ;ASSUME NO EXT FEATURES
012150  005737  002226                  TST     EXTFEA          ;EXT FEATURES SET?
012154  001402                          BEQ     5$              ;BR IF NO
012156  012700  000010                  MOV     #8.,R0          ;EXT FEATURE BUFFER IS 8 WORDS
012162  004737  014552          5$:     JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012166                                  ENDMSG
012166                          L10011:
012166  104423                          TRAP    C$MSG
```

```
                                        .SBTTL  FIFEXP  - PRINT FIFO EXP/RECV DATA
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT FIFO EXP/RECV DATA
                                ;
                                ;       R1      - BYTE COUNT
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
                                ;-
                                        BGNMSG  FIFEXP
                                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
                                        ENDMSG
                                L10012:
                                        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
                                ;-
012350                                  BGNMSG  MSGSTAT
012350                          MSGSTAT::
012350  012701  012412                   MOV     #STATCOD,R1     ;ASCII ADDRESS TABLE
012354  012100                  10$:     MOV     (R1)+,R0        ;DONE ALL MSG LINES?
012356  001410                           BEQ     20$             ;BR IF YES
012360                                   PRINTX  R0              ;PRINT STATUS BIT NAMES
012360  010046                           MOV     R0,-(SP)
012362  012746  000001                   MOV     #1,-(SP)
012366  010600                           MOV     SP,R0
012370  104415                           TRAP    C$PNTX
012372  062706  000004                   ADD     #4,SP
012376  000766                           BR      10$             ;DO ANOTHER MSG LINE
012400  012700  000012          20$:     MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
012404  004737  014552                   JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012410                                   ENDMSG
012410                          L10013:
012410  104423                           TRAP    C$MSG

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


                                        .SBTTL  MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
013064                                  BGNMSG  MSGLOOP
013064                          MSGLOOP::
013064  012701  013126                   MOV     #LOOPCOD,R1     ;ASCII ADDRESS TABLE
```

```
1655 013070  012100                  10$:   MOV     (R1)+,R0        ;DONE ALL MSG LINES?
1656 013072  001410                         BEQ     20$             ;BR IF YES
1657 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$PNTX
     013106  062706  000004                 ADD     #4,SP
1658 013112  000766                         BR      10$             ;DO ANOTHER MSG LINE
1659 013114  012700  000012          20$:   MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
1660 013120  004737  014552                 JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
1661 013124                                 ENDMSG
     013124                          L10014:
     013124  104423                         TRAP    C$MSG
1662
1663 013126  013146  013221  013320  LOOPCOD:       .WORD   1$,2$,3$,4$,5$,6$,7$,0
1664 013146     045     116     045  1$:.ASCIZ  '%N%A Tape Bus Loopback Signals in Word #8:'
1665 013221     045     116     045  2$:.ASCIZ  '%N%A      PARERR<15>          IRESV2<14>          IRESV1<13>'
1666 013320     045     116     045  3$:.ASCIZ  '%N%A IHISP=>IEOT<12>    IWRT=>IIDENT<11>    IREV  =>ICER  <10>'
1667 013417     045     116     045  4$:.ASCIZ  '%N%A IWFM =>IFMK<09>    IEDIT=>IHER <08>    IFAD  =>ISPEED<07>'
1668 013516     045     116     045  5$:.ASCIZ  '%N%A ITAD0=>IRDY<06>    ITAD1=>IONL <05>    IERASE=>ILDP  <04>'
1669 013615     045     116     045  6$:.ASCIZ  '%N%A IREW =>IDBY<03>    IRWU =>IRWD <02>    IFEN  =>IFBY  <01>'
1670 013714     045     116     045  7$:.ASCIZ  '%N%A IGO  =>IFPT<00>'
1671                                        .EVEN
1672
```

```
                                        .SBTTL  MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
                                ;
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
                                        BGNMSG  MSGSUB
013742                          MSGSUB::
013742
013742  012700  000012                    MOV     #10.,R0         ;SIZE OF WRITE SUBSYSTEM BUFFER
013746  004737  014552                    JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
013752                                  ENDMSG
013752                          L10015:
013752  104423                          TRAP    C$MSG


                                        .SBTTL  MEMADD - PRINT MEMORY ADDRESS DATA ERROR
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       ERRHI   - MEMORY ERROR HIGH ORDER ADDRESS
                                ;       ERRLO   - MEMORY ERROR LOW ORDER ADDRESS
                                ;       EXP     - EXPECTED DATA
                                ;       RECV    - RECEIVED DATA
                                ;-
                                        BGNMSG  MEMADD
013754                          MEMADD::
013754
013754  004737  010160                    JSR     PC,PRIADD       ;PRINT MEMORY ADDRESS IN ERROR
013760  013701  002232                    MOV     EXPD,R1         ;GET EXPD DATA
013764  013702  002234                    MOV     RECV,R2         ;GET RECEIVED DATA
013770  004737  007742                    JSR     PC,PRIXOR       ;PRINT EXPD/RECV
013774                                  ENDMSG
013774                          L10016:
013774  104423                          TRAP    C$MSG

```

```
                                .SBTTL  PRAMPKT - PRINT RAM AND PACKET DATA
                        ;+
                        ;
                        ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
                        ;WHEN THE RAM DATA DOES NOT MATCH.
                        ;
                        ;INPUTS:
                        ;
                        ;       R4      POINTER TO COMMAND PACKET
                        ;
                        ;IMPLICIT INPUTS:
                        ;
                        ;       RAMDATA         DATA AS READ FROM THE RAM
                        ;       RAMSIZ          NUMBER OF BYTES IN PACKET
                        ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
                        ;
                        ;IMPLICIT OUTPUTS:
                        ;
                        ;       RAMSIZ  SET TO 0
                        ;-
                        PRAMPKT:
                                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$,NOTSSR
014024  000436                  BR      10$                     ;@@D
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'
```

```
1764                                    .EVEN
1765
1766
```

```
                                .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:
                                SAVREG                          ;SAVE THE REGISTERS
014246  010005                  MOV     R0,R5                   ;SAVE LOW ORDER ADDRESS
014250  005737  003134          TST     KTENABLE                ;ADDRESS ABOVE 28K?
014254  001001                  BNE     10$                     ;BR IF YES
014256  005001                  CLR     R1                      ;SET HIGH ORDER ADDRESS TO 0
014260  010103          10$:    MOV     R1,R3                   ;SAVE HIGH ORDER ADDRESS
014262  006100                  ROL     R0                      ;SHIFT BIT15 TO C BIT
014264  006101                  ROL     R1                      ;SHIFT TO HIGH ORDER FOR PRINTOUT
014266                          PRINTX  #PROASC,R1,R5           ;PRINT MESSAGE BUFFER ADDRESS
014266  010546                  MOV     R5,-(SP)
014270  010146                  MOV     R1,-(SP)
014272  012746  014420          MOV     #PROASC,-(SP)
014276  012746  000003          MOV     #3,-(SP)
014302  010600                  MOV     SP,R0
014304  104415                  TRAP    C$PNTX
014306  062706  000010          ADD     #10,SP
014312                          PRINTX  #PR1ASC                 ;PRINT HEADER FOR CONTENTS
014312  012746  014465          MOV     #PR1ASC,-(SP)
014316  012746  000001          MOV     #1,-(SP)
014322  010600                  MOV     SP,R0
014324  104415                  TRAP    C$PNTX
014326  062706  000004          ADD     #4,SP
014332  005004                  CLR     R4                      ;NUMBER OF THE NEXT WORD
014334  010501                  MOV     R5,R1                   ;COPY LOW ORDER ADDRESS
014336  010300                  MOV     R3,R0                   ;COPY HIGH ORDER ADDRESS
014340  001403                  BEQ     20$                     ;BR IF NOT ABOVE 28K
014342  004737  017316          JSR     PC,SETMAP               ;SETUP PAR ADDRESS IN R0
014346  010005                  MOV     R0,R5                   ;GET PAR FORMAT ADDRESS ABOVE 28K
014350                  20$:    PRINTX  #PRASC,R4,(R5)+         ;PRINT THE CONTENTS OF MEMORY BUFFER
014350  012546                  MOV     (R5)+,-(SP)
014352  010446                  MOV     R4,-(SP)
014354  012746  014523          MOV     #PRASC,-(SP)
014360  012746  000003          MOV     #3,-(SP)
014364  010600                  MOV     SP,R0
014366  104415                  TRAP    C$PNTX
014370  062706  000010          ADD     #10,SP
014374  005204                  INC     R4                      ;NUMBER OF THE NEXT
014376  020427  000007          CMP     R4,#7                   ;DONE ALL YET ?
014402  003005                  BGT     50$                     ;BRANCH IF ALL DONE
```

```
1806 014404  002761                        BLT     20$             ;PRINT FIRST 7 WORDS
1807 014406  032763  000200  000012        BIT     #X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
1808 014414  001355                        BNE     20$             ;PRINT EXTENDED STATUS WORD
1809 014416  000207                 50$:    RTS     PC              ;RETURN
1810
1811 014420     045     116     045  PR0ASC: .ASCIZ  '%N%A Message Buffer Address = %01%05'
1812 014465     045     116     045  PR1ASC: .ASCIZ  '%N%A Message Buffer Contents:'
1813 014523     045     116     045  PRASC:  .ASCIZ  '%N%A    Word%D1%A: %O'
1814                                        .EVEN
```

```
                        .SBTTL  PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
                ;+
                ;
                ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
                ;
                ;       R0      - NUMBER OF WORDS IN BUFFER
                ;
                ;IMPLICIT INPUTS:
                ;
                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                ;-
014552          PRMSGEXP::
014552                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
014556  010005          MOV     R0,R5                   ;SAVE NUMBER OF WORDS
014560  013700  002306  MOV     RCVLOADD,R0             ;GET RECV LOW ADDRESS
014564  010004          MOV     R0,R4                   ;COPY LOW ADDRESS
014566  013701  002304  MOV     RCVHIADD,R1             ;GET RECV HIGH ADDRESS
014572  006100          ROL     R0                      ;SHIFT BIT15 TO C BIT
014574  006101          ROL     R1                      ;SHIFT TO HIGH ORDER FOR PRINTOUT
014576                  PRINTX  #PRMSG0,R1,R4           ;PRINT MESSAGE BUFFER ADDRESS
014576  010446          MOV     R4,-(SP)
014600  010146          MOV     R1,-(SP)
014602  012746  014732  MOV     #PRMSG0,-(SP)
014606  012746  000003  MOV     #3,-(SP)
014612  010600          MOV     SP,R0
014614  104415          TRAP    C$PNTX
014616  062706  000010  ADD     #10,SP
014622                  PRINTX  #PRMSG1                 ;PRINT HEADER FOR CONTENTS
014622  012746  014777  MOV     #PRMSG1,-(SP)
014626  012746  000001  MOV     #1,-(SP)
014632  010600          MOV     SP,R0
014634  104415          TRAP    C$PNTX
014636  062706  000004  ADD     #4,SP
014642  005004          CLR     R4                      ;NUMBER OF THE CURRENT WORD
014644  012701  002322  MOV     #EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
014650  012702  002466  MOV     #RECMSG,R2              ;GET RECV BUFFER ADDRESS
014654  011100  20$:    MOV     (R1),R0                 ;GET EXPD
014656  011203          MOV     (R2),R3                 ;GET RECV
014660                  XOR     R0,R3                   ;XOR EXPD/RECV
014670                  PRINTX  #PRMSG2,R4,(R1)+,(R2)+,R3
014670  010346          MOV     R3,-(SP)
014672  012246          MOV     (R2)+,-(SP)
014674  012146          MOV     (R1)+,-(SP)
014676  010446          MOV     R4,-(SP)
014700  012746  015035  MOV     #PRMSG2,-(SP)
014704  012746  000005  MOV     #5,-(SP)
014710  010600          MOV     SP,R0
014712  104415          TRAP    C$PNTX
014714  062706  000014  ADD     #14,SP
014720  005204          INC     R4                      ;NUMBER OF THE NEXT
014722  020405          CMP     R4,R5                   ;DONE ALL YET?
014724  002001          BGE     50$                     ;BR IF YES
014726  000752          BR      20$                     ;DO ANOTHER
014730  000207  50$:    RTS     PC                      ;RETURN
```

```
1852
1853 014732  045  116  045  PRMSG0: .ASCIZ  '%N%A Message Buffer Address = %01%05'
1854 014777  045  116  045  PRMSG1: .ASCIZ  '%N%A Message Buffer Contents:'
1855 015035  045  116  045  PRMSG2: .ASCIZ  '%N%A   WORD #%D2%A  EXPD: %06%A  RECV: %06%A  XOR: %06'
1856                         .EVEN
1857
```

```
                        .SBTTL  PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
                ;+
                ;
                ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
                ;       ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
                ;
                ;       R0      - NUMBER OF BYTES IN BUFFER
                ;
                ;IMPLICIT INPUTS:
                ;
                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                ;-
015122          PRBYTEXP::
015122                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
015126  010005          MOV     R0,R5                   ;SAVE NUMBER OF BYTES
015130  005037  002320  CLR     PRMNO                   ;INIT ERROR COUNT
015134  005004          CLR     R4                      ;NUMBER OF THE CURRENT BYTE
015136  012701  002322  MOV     #EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
015142  012702  002466  MOV     #RECMSG,R2              ;GET RECV BUFFER ADDRESS
015146  111100          20$:    MOVB    (R1),R0         ;GET EXPD BYTE
015150  042700  177400  BIC     #^C<377>,R0             ;CLEAR UPPER BYTE
015154  110037  015470  MOVB    R0,PRBEXP               ;SAVE FOR ERROR REPORT
015160  111203          MOVB    (R2),R3                 ;GET RECV BYTE
015162  042703  177400  BIC     #^C<377>,R3             ;CLEAR UPPER BYTE
015166  110337  015472  MOVB    R3,PRBREC               ;FOR ERROR REPORT
015172                  XOR     R0,R3                   ;XOR EXPD/RECV
015202  122122          CMPB    (R1)+,(R2)+             ;EXPD = RECV?
015204  001431          BEQ     30$                     ;BR IF YES
015206  005237  002320  INC     PRMNO                   ;UPDATE ERROR COUNT
015212  023727  002320  000010  CMP     PRMNO,#8.       ;PRINTED 8?
015220  101023          BHI     30$                     ;BR IF YES
015222                  27$:    PRINTX  #PRBMSG,R4,PRBEXP,PRBREC,R3
015222  010346          MOV     R3,-(SP)
015224  013746  015472  MOV     PRBREC,-(SP)
015230  013746  015470  MOV     PRBEXP,-(SP)
015234  010446          MOV     R4,-(SP)
015236  012746  015336  MOV     #PRBMSG,-(SP)
015242  012746  000005  MOV     #5,-(SP)
015246  010600          MOV     SP,R0
015250  104415          TRAP    C$PNTX
015252  062706  000014  ADD     #14,SP
015256                  FORCEXIT        50$             ;@@D
015266  000404          BR      35$                     ;@@D
015270                  30$:
015270                  FORCERROR  27$,NOTSSR           ;@@D
015300                  35$:                            ;@@D
015300  005204          INC     R4                      ;NUMBER OF THE NEXT
015302  020405          CMP     R4,R5                   ;DONE ALL YET?
015304  002001          BGE     50$                     ;BR IF YES
015306  000717          BR      20$                     ;DO ANOTHER
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
1902    015334  000207                          RTS     PC              ;RETURN
1903
1904    015336      045     116     045 PRBMSG: .ASCIZ  '%N%A   BYTE #%D2%A  EXPD: %O3%A  RECV: %O3%A  XOR: %O3'
1905    015423      045     116     045 PRBTOT: .ASCIZ  '%N%A NUMBER OF BYTES IN ERROR = %D2'
1906                                    .EVEN
1907    015470  000000                  PRBEXP: .WORD   0               ;EXPD
1908    015472  000000                  PRBREC: .WORD   0               ;RECV
1909
```

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

015474                          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
                                EXPBREC::
015502  004737  007612                  JSR     PC,PRIBXOR              ;PRINT THE DATA
                                        ENDMSG
                                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::
                                JSR     PC,PRITADD      ;PRINT TEST ADDRESS
                                JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
                                ENDMSG
                        L10022:
                                TRAP    C$MSG
                        ;
                        ;
                                .SBTTL  RAMEXP  - PRINT RAM EXPD/RECV DATA
                        ;+
                        ;
                        ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                        ;
                        ;INPUTS:
                        ;
                        ;       R1      RECEIVED DATA
                        ;       R2      EXPECTED DATA
                        ;       R4      CONTROLLER RAM ADDRESS
                        ;-
                        ;
                                BGNMSG  RAMEXP
                        RAMEXP::
                                BIC     #^C<377>,R1             ;SAVE EXPD RAM DATA BYTE
                                BIC     #^C<377>,R2             ;SAVE EXPD RAM DATA BYTE
                                JSR     PC,PRIRAM               ;PRINT THE RAM ADDRESS
                                JSR     PC,PRIXOR               ;PRINT THE DATA
                                ENDMSG
                        L10023:
                                TRAP    C$MSG
                        ;
                        ;
                                .SBTTL  TIMEXP  - PRINT TIMER A,B AND EXP/REC
                        ;+
                        ;
                        ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                        ;AND TIMER A,B HEADER MESSAGE
                        ;
                        ;INPUTS:
                        ;
                        ;       R1      RECEIVED DATA
                        ;       R2      EXPECTED DATA
```

```
                                        ;-
                                        
015552                                          BGNMSG  TIMEXP
015552                                  TIMEXP::
015552                                          PRINTX  #TIMSG0                 ;PRINT HEADER
015552  012746  015600                          MOV     #TIMSG0,-(SP)
015556  012746  000001                          MOV     #1,-(SP)
015562  010600                                  MOV     SP,R0
015564  104415                                  TRAP    C$PNTX
015566  062706  000004                          ADD     #4,SP
015572  004737  007742                          JSR     PC,PRIXOR               ;PRINT THE DATA
015576                                          ENDMSG
015576                                  L10024:
015576  104423                                  TRAP    C$MSG

                                        
015600    045    116    045  TIMSG0: .ASCIZ  '%N%A TIMER A STATUS IS IN BIT 3%N%A TIMER B STATUS IS IN BIT 2'
                                        .EVEN
```

```
                                        .SBTTL  BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

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

015700                                  BGNMSG  BADSSR
015700                          BADSSR::
015700  010246                          MOV     R2,-(SP)                ;SAVE DATA TRANSFERRED
015702  042702  177400                  BIC     #177400,R2              ;GET JUST ONE BYTE
015706                                  PRINTB  #XFERASC,R2
015706  010246                          MOV     R2,-(SP)
015710  012746  015740                  MOV     #XFERASC,-(SP)
015714  012746  000002                  MOV     #2,-(SP)
015720  010600                          MOV     SP,R0
015722  104414                          TRAP    C$PNTB
015724  062706  000006                  ADD     #6,SP
015730  012602                          MOV     (SP)+,R2                ;RESTORE R2
015732  004737  006024                  JSR     PC,PRITSSR              ;DECODE TSSR CONTENTS
015736                                  ENDMSG
015736                          L10025:
015736  104423                          TRAP    C$MSG
015740    045     116     045   XFERASC:        .ASCIZ  '%N%A Data Transferred = %03'
```

```
                                .SBTTL  GLOBAL SUBROUTINES SECTION

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

```
                        .SBTTL  SOFINIT - SOFT INITIALIZE OF CONTROLLER

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

015774          SOFINIT::
015774                  SAVREG                          ; SAVE THE REGISTERS
016000  012765  000000  000002          MOV     #0,TSSR(R5)     ; DO THE INIT.
016006  004737  016250                  JSR     PC,WAITF        ; WAIT FOR SSR
016012  016500  000002                  MOV     TSSR(R5),R0     ;GET THE TSSR REGISTER
016016  010004                          MOV     R0,R4           ;TSSR CONTENTS
016020  042704  176277                  BIC     #^C<HIADDR!OFL>,R4
016024  052704  002200                  BIS     #SSR!NBA,R4     ;R4 HAS EXPECTED CONTENTS
016030  020400                          CMP     R4,R0           ;ONLY EXPECTED BITS SET ?
016032  001402                          BEQ     5$              ;BRANCH IF OKAY
016034  000241                          CLC                     ;CLEAR THE CARRY FOR ERROR
016036  000401                          BR      10$             ;GO TO EXIT
016040  000261                  5$:     SEC                     ;SET THE CARRY BIT
016042  000207                  10$:    RTS     PC              ;RETURN TO CALLER
```

```
                        .SBTTL  CHKAMB  - CHECK TSSR FOR AMBIGUITY

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

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

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

                ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
                ;
        000200          IOKCKIN=BIT7    ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
        000001          IOKSTP=BIT0     ; EXPECT "STOP" INTERRUPT.
                ;
                ;INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
016144  000     INTMASK:        .BYTE   0
                ;INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
016145  000     INTFLAG:        .BYTE   0

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

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

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

```
                                                .SBTTL  INTR    - INTERRUPT HANDLERS

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

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

```
                                .SBTTL  WAITF   - WAIT FOR SUBSYSTEM READY
                        ;
                        ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
                        ;
                        ;INPUTS:
                        ;
                        ;       R5      ADDRESS OF FIRST DEVICE REGISTER
                        ;
                        ;OUTPUTS:
                        ;
                        ;       R0      CONTENTS OF LAST TSSR READ
                        ;       CARRY   SET - READY BIT SET
                        ;               CLR - TIMEOUT WAITING FOR READY
                        ;
016250  000401          WAITF:: BR      1$              ;NOP WHEN SUPER FIXED
016252                          BREAK                   ; DO A SUPVSR BREAK FIRST.
016252  104422                  TRAP    C$BRK
016254  012746  003000  1$:     MOV     #3000,-(SP)     ;300 MSEC TIMER
016260  016500  000002  2$:     MOV     TSSR(R5),R0     ;READ THE TSSR REGISTER
016264  105700                  TSTB    R0              ;TEST FOR READY BIT SET
                        ;
016266  100420                  BMI     3$              ; EXIT ON STOP FLAG.
016270                          DELAY   1               ; WAIT 100 USEC
016270  012727  000001          MOV     #1,(PC)+
016274  000000                  .WORD   0
016276  013727  002116          MOV     L$DLY,(PC)+
016302  000000                  .WORD   0
016304  005367  177772          DEC     -6(PC)
016310  001375                  BNE     .-4
016312  005367  177756          DEC     -22(PC)
016316  001367                  BNE     .-20
016320  005316                  DEC     (SP)            ;REDUCE DELAY COUNT
016322  001356                  BNE     2$              ;RETRY UNTIL TIMER EXPIRES
016324  000241                  CLC                     ; C = 0, CONTROLLER STILL RUNNING...
016326  000401                  BR      4$              ;...OR HUNG-UP AFTER 300 MSEC.
016330  000261          3$:     SEC                     ; C = 1, CONTROLLER IS STOPPED.
016332  005326          4$:     DEC     (SP)+           ;RESTORE STACK WITHOUT CHANGING CARRY BIT
016334  000207                  RTS     PC
```

```
                                .SBTTL  CHKTSSR - CHECK TSSR FOR READY

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

016336                  CHKTSSR:
016336  004737  016250          JSR     PC,WAITF                ;WAIT FOR READY
016342  103014                  BCC     20$                     ;BRANCH IF TIME OUT
016344  004737  016044          JSR     PC,CHKAMB               ;TSSR AMBIGOUS?
016350  103006                  BCC     10$                     ;BR IF YES
016352  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
                ;       MOV ADR2,R2
                ;       JSR PC,NXM
                ;       RETURN          ;TEST "C" AND PROCEED.
                ;
016376  012737  016430  000004  XNXM:   MOV     #2$,@#4         ; SET BUSERR VECTOR.
016404  012737  000200  000006          MOV     #PRI04,@#6
016412  005003                          CLR     R3              ;FLAG.
016414  005711                  1$:     TST     (R1)            ;TEST THE ADDRESS(ES).
                                                                ;IF ANY TRAP, CONTINUE AT 2$.
016416  020102                          CMP     R1,R2           ;OTHERWISE, CONTINUE HERE.
016420  001407                          BEQ     3$              ;BR IF FINISHED (NO NEXM'S).
016422  062701  000002                  ADD     #2,R1           ;SET NEXT ADDRESS...
016426  000772                          BR      1$              ;...AND CONTINUE.

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


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

```
                                        .SBTTL  TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
                                ;+
                                ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
                                ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
                                ; IN THE CURRENT RUN SEQUENCE.
                                ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
                                ;
                                ;INPUT:
                                ;
                                ;       R0      POINTER TO TEST ID ASCIZ STRING
                                ;
                                ;OUTPUT:
                                ;
                                ;       R5      ADDRESS OF FIRST DEVICE REGISTER
                                ;
                                ;IMPLICIT OUTPUTS:
                                ;
                                ;       TSTCNT  UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;       INTERRUPT LEVEL IS RASIED TO LEVEL OF
                                ;       THE DEVICE UNDER TEST
                                ;
                                ;-
                                ;
016510                          TSTSETUP::
016510  010046                          MOV     R0,-(SP)        ;SAVE THE TEST ID MESSAGE
016512  005037  003154                   CLR     SIFLAG          ; CLEAR "SOFT INIT" FLAG
016516  005037  016756                   CLR     ERRK            ; CLEAR LOCAL ERROR COUNTER.
016522  005037  005772                   CLR     EXTA            ; CLEAR ERROR EXTENSION FLAG.
016526  105037  016144                   CLRB    INTMASK         ; CLEAR INTERRUPT MASK (CHECK ERROR)
016532  013700  002202                   MOV     UNITN,R0        ; GET THE UNIT NUMBER,
016536  006300                           ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
016540  005737  003114                   TST     NODEV           ; DID STARTUP FIND THE DEVICE?
016544  001430                           BEQ     4$              ; BR IF YES
016546  100010                           BPL     3$              ; BR IF NOT IDLE
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$DCLN
2378            016624  000423                          BR      5$
2379
2380            016626                          4$:     RFLAGS  R0              ; GET THE OPERATOR FLAGS.
                016626  104421                          TRAP    C$RFLA
2381            016630  032700  001000                  BIT     #PNT,R0         ; PRINT THE TEST NUMBERS?
2382            016634  001412                          BEQ     1$              ; BR IF NO
2383            016636  011600                          MOV     (SP),R0         ;GET THE ID MESSAGE
2384            016640                                  PRINTF  #TNAM,R0                ;DISPLAY THE TEST ID
                016640  010046                          MOV     R0,-(SP)
                016642  012746  016704                  MOV     #TNAM,-(SP)
                016646  012746  000002                  MOV     #2,-(SP)
                016652  010600                          MOV     SP,R0
                016654  104417                          TRAP    C$PNTF
                016656  062706  000006                  ADD     #6,SP
2385            016662  005237  002214          1$:     INC     TSTCNT          ; BUMP TEST COUNTER.
2386            016666                                  SETPRI  IPRI            ;PRIORITY THAT OF DEVICE
                016666  013700  002212                  MOV     IPRI,R0
                016672  104441                          TRAP    C$SPRI
2387            016674  005726                  5$:     TST     (SP)+           ;FIX UP THE STACK
2388            016676  013705  002206                  MOV     CSRADDR,R5      ; ADDRESS OF TSV REGISTERS ON UNIBUS
2389            016702  000207                          RTS     PC
2390            016704     045     123     045  TNAM:  .ASCIZ  '%S%T%A Test'
2391                                                    .EVEN
```

```
                                        .SBTTL  TSTEND  - PRINT ERRORS RECEIVED
                                ;
                                ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
                                ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
                                ;
016720                          TSTEND: RFLAGS  R0
016720  104421                         TRAP    C$RFLA
016722  030027  020000                 BIT     R0,#IER
016726  001412                         BEQ     1$                      ; BR IF "IER" NOT SET.
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$PNTF
016750  062706  000006                 ADD     #6,SP
016754  000207                  1$:     RTS     PC

016756  000000                  ERRK:   0                               ; LOCAL ERROR COUNT.
016760     045     101     040  ESUM:   .ASCIZ  /%A %D%A ERRORS/
016777     105     122     122  EMAXDU: .ASCIZ  /ERROR LIMIT REACHED -- DROPPING UNIT/
                                        .EVEN


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

017102  010046                  CKEMAX: MOV     R0,-(SP)                ; SAVE R0
017104  013700  002202                 MOV     UNITN,R0                ; GET UNIT NUMBER
017110  006300                         ASL     R0                      ; ... AND MAKE IT A WORD OFFSET
017112  016000  003176                 MOV     ERTABL(R0),R0           ; GET ERROR TABLE ENTRY
017116  042700  170000                 BIC     #170000,R0              ; EXTRACT ERROR COUNT FIELD
017122  020037  002174                 CMP     R0,GERRMAX              ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
017126  103004                         BHIS    1$                      ; BR IF YES
017130  023737  016756  002172         CMP     ERRK,LERRMAX            ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
017136  103417                         BLO     2$                      ; BR IF NO
017140                          1$:     RFLAGS  R0                      ; GET OPERATOR FLAGS
017140  104421                         TRAP    C$RFLA
017142  032700  000040                 BIT     #IDU,R0                 ; IS DROPPING INHIBITED?
017146  001013                         BNE     2$                      ; BR IF YES.
017150  012737  177777  003112         MOV     #-1,DUFLG               ; NO -- DROP THE UNIT
017156                                  ERRDF   4,EMAXDU
017156  104455                         TRAP    C$ERDF
017160  000004                         .WORD   4
017162  016777                         .WORD   EMAXDU
```

```
                017164  000000                          .WORD   0
     2439       017166                                  DODU    UNITN
                017166  013700  002202                  MOV     UNITN,R0
                017172  104451                          TRAP    C$DODU
     2440       017174                                  DOCLN
                017174  104444                          TRAP    C$DCLN
     2441       017176  012600              2$:         MOV     (SP)+,R0        ; RESTORE R0
     2442       017200  000207                          RTS     PC              ; RETURN TO CALLER
     2443
```

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


                                        .SBTTL  CONFIG  - DETERMINE CONFIGURATION OF SYSTEM
                                ;
                                ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
                                ;
017250                          CONFIG:
017250  004737  015774                  JSR     PC,SOFINIT
017254  000207                          RTS     PC
```

```
                                        .SBTTL  KTON,KTOFF      - ENABLE/DISABLE MEMORY MANAGEMENT
                                ;
                                ; SUBROUTINE - ENABLE MEM MGT.
                                ;
017256  005737  003132          KTON:   TST     KTFLG           ; GOT KT?
017262  001403                          BEQ     1$              ; NO.
017264  012737  000001  177572          MOV     #1,SR0          ; YES. ENABLE KT11.
017272  000207                  1$:     RTS     PC


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

```
                        .SBTTL  SETMAP  - SETUP PAR6 MAPPING

                ;+
                ;
                ;THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
                ;AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
                ;IS RETURNED BIASED TO PAR6.
                ;
                ;INPUTS:
                ;
                ;       R0      HIGH ORDER ADDRESS BITS
                ;       R1      LOW ORDER ADDRESS BITS
                ;
                ;OUTPUTS:
                ;
                ;       R0      OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
                ;       CARRY   SET IF SUCCESS
                ;               CLR IF ERROR
                ;-
017316          SETMAP:
017316                  SAVREG                          ;SAVE R1-R4 UNTIL NEXT RETURN
017322  005737  003132          TST     KTFLG           ;SYSTEM HAVE ABOVE 28K?
017326  001433                  BEQ     10$             ;BR IF NO
017330  010102                  MOV     R1,R2           ;SAVE LOW ORDER BITS
        000006                  .REPT   6
                                ASR     R0              ;CONVERT WORD ADDRESS TO 32W BLOCKS
                                ROR     R1              ;MAKE IT DOUBLE PRECISION
                                .ENDR
017362  042701  000177          BIC     #177,R1         ;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.
                ;
                ; 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
```

```
2596
2597
```

```
                        .SBTTL  CMPMEM  - COMPARE MEMORY TO BACKGROUND PATTERN
                ;+
                ; COMPARE MEMORY WITH A BACKGROUND PATTERN
                ;
                ; INPUTS:
                ;
                ;       R0 = BACKGROUND PATTERN
                ;       FREE    = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
                ;       KTFLG   = SET TO HIGHEST MEMORY LOCATION IF > 28K.
                ;
                ; OUTPUTS:
                ;
                ;       CARRY   - SET IF NO ERROR
                ;       CARRY   - CLR IF ERROR
                ;
                ; IMPLICIT OUTPUTS:
                ;
                ;       ERRHI   - ERROR HIGH ADDRESS
                ;       ERRLO   - ERROR LOW ADDRESS
                ;       EXPD    - EXPECTED DATA
                ;       RECV    - RECEIVED DATA
                ;-
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
```

```
2656 020040  010137  002240            MOV     R1,ERRLO        ;SAVE LOW ORDER IN ERROR
2657 020044  010337  002232            MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
2658 020050  011437  002234            MOV     (R4),RECV       ;SAVE RECV FOR ERROR REPORT
2659 020054  000421                    BR      50$             ;
2660 020056  062701  000002      32$:  ADD     #2,R1           ;UPDATE NON PAR6 ADDRESS
2661 020062  005500                    ADC     R0              ;MAKE IT DOUBLE PRECISION ADD
2662 020064  062704  000002            ADD     #2,R4           ;UPDATE PAR FORMAT ADDRESS
2663 020070  020427  160000            CMP     R4,#160000      ;END OF PAR6 MAPPING AREA?
2664 020074  103755                    BLO     30$             ;BR IF NO
2665 020076  162704  020000            SUB     #20000,R4       ;BACKUP INTO PAR6 MAPPING BEGIN
2666 020102  062737  000200  172354    ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
2667 020110  023737  172354  003132    CMP     @#KIPAR6,KTFLG  ;END OF MEMORY?
2668 020116  101744                    BLOS    30$             ;BR IF NO
2669 020120  004737  017274      50$:  JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
2670 020124  000241                    CLC                     ;SET FAILURE
2671 020126  000403                    BR      60$             ;
2672 020130  004737  017274      55$:  JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
2673 020134  000261                    SEC                     ;SET SUCCESS
2674 020136  000207              60$:  RTS     PC
2675
```

```
                                        .SBTTL  REGSAV  - SAVE R1-R5 ON STACK
                                ;+
                                ;
                                ;ROUTINE TO
                                ;SAVE R1 THROUGH R5 ON THE STACK
                                ;
                                ;CALLING SEQUENCE:
                                ;
                                ;       JSR     R5,REGSAV
                                ;
                                ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
                                ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
                                ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
                                ;REGISTERS.
                                ;
                                ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
                                ;CALLED VIA A JSR PC INSTRUCTION
                                ;
                                ;-

020140                          REGSAV:
020140  010446                          MOV     R4,-(SP)
020142  010346                          MOV     R3,-(SP)
020144  010246                          MOV     R2,-(SP)
020146  010146                          MOV     R1,-(SP)
020150  010546                          MOV     R5,-(SP)
020152  016605  000012                  MOV     10.(SP),R5
020156  004736                          JSR     PC,@(SP)+
020160  012601                          MOV     (SP)+,R1
020162  012602                          MOV     (SP)+,R2
020164  012603                          MOV     (SP)+,R3
020166  012604                          MOV     (SP)+,R4
020170  012605                          MOV     (SP)+,R5
020172  000207                          RTS     PC

```

```
                        .SBTTL  GETPAT  - GET 8 BIT PATTERN FROM OPERATOR
                ;+
                ;
                ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
                ;
                ;INPUTS:
                ;
                ;       NONE.
                ;
                ;OUTPUTS:
                ;
                ;       R0      OCTAL NUMBER FROM THE OPERATOR
                ;
                ;CALLING SEQUENCE:
                ;
                ;       JSR     PC,GETPAT
                ;
                ;-
                GETPAT::
                        SAVREG                          ;SAVE THE GENERAL REGISTERS
                1$:     GMANID  DATASC,PATDAT,0,377,0,377,NO
020200  104443          TRAP    C$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$HILIM
020220                  10000$:
020220                          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
                                TRAP    C$PNTF
                                ADD     #6,SP
                                BR      2$
                        3$:     GMANID  MENASC,MENRES,D,-1,0,-1,NO
                                TRAP    C$GMAN
                                BR      10001$
                                .WORD   MENRES
                                .WORD   T$CODE
                                .WORD   MENASC
                                .WORD   -1
                                .WORD   T$LOLIM
                                .WORD   T$HILIM
                        10001$:
                                BNCOMPLETE      1$              ;RETRY IF ERROR
                                BCC     1$
                                MOV     MENRES,R0               ;GET THE OPERATOR'S REPLY
                                CMP     R0,R1                   ;COMPARE TO MAXIMUM ALLOWED
                                BLOS    5$                      ;BRANCH IF OK
                                PRINTF  #MENERR                 ;DISPLAY ERROR MESSAGE
                                MOV     #MENERR,-(SP)
                                MOV     #1,-(SP)
                                MOV     SP,R0
                                TRAP    C$PNTF
                                ADD     #4,SP
                                BR      1$                      ;RETRY
                        5$:     RTS     PC                      ;RETURN TO CALLER
                        MENERR: .ASCIZ  '%N%A *** Menu Selection Too Large ***'
                        SELASC: .ASCIZ  '%N%T'
                        MENASC: .ASCIZ  'Enter Menu Selection: '
```

```
                                        .EVEN
020476  000000          MENRES:         .WORD   0
```

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

020500          CHKMAN::
020500                  SAVREG                  ;SAVE THE REGISTERS
020504                  MANUAL                  ;SEE IF MANUAL INTERVENTION OK
020504  104450          TRAP    C$MANI
020506                  BCOMPLETE 1$            ;BRANCH IF ALLOWED
020506  103411          BCS     1$
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$PNTF
020524  062706  000004          ADD     #4,SP
020530  000241                  CLC                     ;CLEAR CARRY FOR ERROR
020532  000207          1$:     RTS     PC              ;RETURN

020534  045  116  045  NOMAN:  .ASCIZ  '%N%A *** Manual Intervention not Allowed - Test Aborted ***'
                        .even
```

```
                        .SBTTL  ENVIRN  - SETUP FREE DIAGNOSTIC SPACE
                ;
                ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
                ;
020630          ENVIRN: MEMORY  R0
020630 104431           TRAP    C$MEM
020632 010037 003124    MOV     R0,FREE         ; GET 1ST FREE ADDRESS...
020636 062737 000002 003124     ADD     #2,FREE
020644 011037 003126    MOV     (R0),FRESIZ     ;...AND WORD COUNT.
020650 162737 000004 003126     SUB     #4,FRESIZ
020656 013702 002012    MOV     L$UNIT,R2       ; GET NUMBER OF UNITS
020662 162737 000007 003126 10$:        SUB     #7,FRESIZ       ; TAKE AWAY 7 WORDS PER UNIT
020670 005302           DEC     R2
020672 001373           BNE     10$
020674 013700 003124    MOV     FREE,R0         ;GET FIRST FREE ADDRESS
020700 063700 003126    ADD     FRESIZ,R0       ;POINT TO LAST FREE ADDRESS
020704 162700 000002    SUB     #2,R0           ;BACKUP 1 WORD
020710 010037 003130    MOV     R0,FREEHI       ;STORE LAST FREE ADDRESS
020714 000240           NOP                     ;***********************************************
020716 012701 177520    MOV     #BDVPCR,R1      ;GET BDV11 PCR ADDRESS
020722 010102           MOV     R1,R2           ;COPY TO R2
020724 062702 000002    ADD     #2,R2           ;SET THE RANGE
020730 004737 016376    JSR     PC,XNXM         ;SEE IF WE HAVE ONE
020734 103001           BCC     15$             ;OK TO SET FLAGS
020736 000445           BR      40$             ;RETURN WITH FLAGS CLEAR
020740 013701 177520 15$:       MOV     BDVPCR,R1       ;SAVE PCR CONTENTS
020744 062701 000001    ADD     #1,R1           ;ADD ONE TO IT
020750 012702 177520    MOV     #BDVPCR,R2      ;GET BDV11 PCR ADDRESS
020754 005212           INC     (R2)            ;TRY TO WRITE TO IT
020756 013703 177520    MOV     BDVPCR,R3       ;GET RESULTS
020762 020103           CMP     R1,R3           ;DID IT CHANGE?
020764 001017           BNE     20$             ;NO, MUST BE 11/23B
020766 005237 003144    INC     T23A            ;SET THE FLAG
020772 042737 170000 002120     BIC     #170000,L$HIME  ;SUPERVISOR COULD BE WRONG
021000 000240           NOP                     ;BR 40$ FOR RELEASE
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$PNTF
021016 062706 000004    ADD     #4,SP
021022 000413           BR      40$             ;RETURN
021024 005237 003146 20$:       INC     T23B            ;SET THE FLAG
021030 000240           NOP                     ;BR 40$ FOR RELEASE
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$PNTF
021046 062706 000004    ADD     #4,SP
021052 000207           40$:    RTS     PC              ;RETURN
```

```
                                        .SBTTL  KTINIT  - SETUP KT11 MEMORY MANAGEMENT REGISTERS
                                ;+
                                ;
                                ;ROUTINE TO INIT KT-11
                                ;
                                ;-
021054                          KTINIT:
021054  005037  003132                  CLR     KTFLG           ; INIT >28K MEMORY FLAG
021060  005037  003134                  CLR     KTENABLE        ; INIT TEST >28K FLAG
021064  023727  002120  001577          CMP     L$HIME,#1577    ; GOT ENOUGH MEMORY (>28K)?
021072  101444                          BLOS    9$              ; NO.
021074  013700  000004                  MOV     @#ERRVEC,R0     ; SAVE OLD ERR VEC PTR.
021100  012737  021172  000004          MOV     #2$,@#ERRVEC    ; SET ERR VEC PTR.
021106  005737  177572                  TST     @#SR0           ; GOT KT11?
021112  000240                          NOP                     ;  (TRAP IF NO).
021114  013737  002120  003132          MOV     L$HIME,KTFLG    ; YES. SET KT FLAG.
021122  042737  000177  003132          BIC     #177,KTFLG      ;
021130  010037  000004                  MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
021134  005000                          CLR     R0              ; R0 = AR DATA.
021136  012701  172340                  MOV     #KIPAR0,R1      ; R1 = KI REGS PTR.
021142  012761  077406  177740  1$:     MOV     #77406,-40(R1)  ; SET DESCRIPTOR REG.
021150  010021                          MOV     R0,(R1)+        ; SET KIPAR REG.
021152  062700  000200                  ADD     #200,R0         ; BUMP AR DATA BY "4K".
021156  020027  002000                  CMP     R0,#2000        ; AT "I/O"?
021162  001367                          BNE     1$              ; NO.
021164  012741  177600                  MOV     #177600,-(R1)   ; YES. SET KTPAR7 FOR I/O.
021170  000405                          BR      9$
                                ;
021172  012716  021200          2$:     MOV     #6$,(SP)        ; SET UP RETURN
021176  000002                          RTI                     ; RTI TO NEXT LOCATION
                                ;
021200  010037  000004          6$:     MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
                                ;
021204  000207                  9$:     RTS     PC
```

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

021206                          INVERT::

021206  005737  002226                TST     EXTFEA                  ; IS SWITCH SET?
021212  001020                       BNE     1$                      ; YES,EXIT STAGE RIGHT!(or the next one outa town!)
021214  012737  100206  021260        MOV     #100206,CMDPKT          ; WRT SUB-SYS MEM CMD
021222  012737  021270  021262        MOV     #WSMBK,CMDPKT+2         ; MSG BUF ADDR
021230  012737  000006  021266        MOV     #6,CMDPKT+6             ; BYTE COUNT
021236  012737  100010  021270        MOV     #100010,WSMBK           ; INVERT THE SWITCH
021244  012704  021260                MOV     #CMDPKT,R4              ; SET CMDPKT INTO R4
021250  004737  010662                JSR     PC,WRTCHR               ; DO IT
021254  000207                  1$:   RTS     PC                      ; RETURN

                                ;       COMMAND PACKET.

021260                          .       =       <.+3>&177774    ;MUST BE ON MOD 4 BOUNDRY.

021260  000000                  CMDPKT:: 0                      ;1ST WORD IS TSO5 COMMAND.
021262  000000                           0                      ;2ND WORD IS THE BUFFER LOW ADDRESS.
021264  000000                           0                      ;3RD WORD IS THE BUFFER HIGH ADDRESS.
021266  000000                           0                      ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.

                                ;       WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.

021270  000000                  WSMBK::  0                      ;1ST WORD:: SEL 0
021272  000000                           0                      ;2ND WORD:: SEL 2
021274  000000                           0                      ;3RD WORD:: SEL 4
                                        .EVEN
```

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

021276                                  MEMCK::

021276                                          SAVREG                          ;SAVE THE REGISTERS
021302  005037  003136                          CLR     NXMFLG                  ;CLEAR THE FLAG
021306  005037  003140                          CLR     NXMLO                   ;CLEAR THE TEST ADDRESS LO
021312  005037  003142                          CLR     NXMHI                   ;CLEAR THE TEST ADDRESS HI
021316  005737  003146                          TST     T23B                    ;IS IT A 11/23B?
021322  001407                                  BEQ     1$                      ;NO
021324  023727  002120  007777                  CMP     L$HIME,#7777            ; GREATER THAN 128K
021332  103406                                  BLO     2$                      ; NO
021334  004737  021452                          JSR     PC,NXMTST               ;SETUP THE ADDRESS
021340  000427                                  BR      13$                     ;SET THE FLAG AND EXIT
021342  005737  003144          1$:             TST     T23A                    ;IS IT A 11/23A?
021346  001413                                  BEQ     4$                      ;NO
021350  023727  002120  005777  2$:             CMP     L$HIME,#5777            ;GREATER THAN 96K
021356  101023                                  BHI     14$                     ;YES,23A/23B WITH 128K MEMORY
021360  023727  002120  003777                  CMP     L$HIME,#3777            ;GREATER THAN 64K BUT LESS THAN 92K?
021366  103403                                  BLO     4$                      ;NO, CHECK 24K
021370  004737  021452                          JSR     PC,NXMTST               ;SETUP THE ADDRESS
021374  000411                                  BR      13$                     ;SET THE FLAG AND EXIT
021376  023727  002120  001577  4$:             CMP     L$HIME,#1577            ;GREATER THAN 24K BUT LESS THAN 64K?
021404  103410                                  BLO     14$                     ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
021406  004737  021452                          JSR     PC,NXMTST               ;SETUP THE ADDRESS
021412  062737  000077  003142                  ADD     #77,NXMHI               ;FOOL THE 11/02 & 11/03
021420  005237  003136          13$:            INC     NXMFLG                  ;SET THE FLAG
021424  000411                                  BR      15$                     ;EXIT
021426  000410                  14$:            BR      15$                     ;NOP FOR PRINTOUT
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$PNTF
021444  062706  000004                          ADD     #4,SP
021450  000207                  15$:            RTS     PC                      ;RETURN

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

021452  013701  002120          NXMTST: MOV     L$HIME,R1                       ;GET TOP OF MEMORY
021456  062701  000200                  ADD     #200,R1                         ;MAKE IT I/O BLOCK OR OTHER NXM
021462  042701  000177                  BIC     #177,R1
021466  010102                          MOV     R1,R2                           ;RESAVE RESULTS
```

```
2994            000006                          .REPT   6
2995                                            ASL     R1              ;PUT IN PLACE FOR XFER
2996                                            .ENDR
2997 021504  010137  003140                     MOV     R1,NXMLO        ;SAVE TEST ADDRESS LOW
2998            000012                          .REPT   10.
2999                                            ASR     R2              ;PUT IN PLACE FOR XFER
3000                                            .ENDR
3001 021534  042702  177700                     BIC     #177700,R2      ;DON'T WANT ILA!
3002 021540  010237  003142                     MOV     R2,NXMHI        ;SAVE TEST ADDRESS HIGH
3003 021544  000207                             RTS     PC              ;RETURN
3004
3005
3006
3007
3008 021546                                     ENDMOD
```

```
                                                    I 9
      7                                     .TITLE  TSV4 - MISCELLANEOUS SECTIONS
      8
      9 021546                              BGNMOD  TSV4
        021546                        TSV4::
     10
     16
```

```
                                        .SBTTL  PROTECTION TABLE
021546                                  BGNPROT
021546                          L$PROT::
021546  177777  177777  177777        .WORD   -1, -1, -1, -1          ;NO DEVICE PROTECTION REQUIRED.
021556                                  ENDPROT
```

```
                                        .SBTTL  INITIALIZE SECTION

                                ;++
                                ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ;AT THE BEGINNING OF EACH PASS.
                                ;
                                ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
                                ;IF "CONTINUE", NOTHING IS REQUIRED.
                                ;
                                ;--
                                ;+
                                ;INSERT TEMPORARY JUMP TO ODT
                                ;-
021556                                  BGNINIT
021556                          L$INIT::
021556 005037 002226            40$:    CLR     EXTFEA
021562 005037 003136                    CLR     NXMFLG
021566 012737 006360 002200             MOV     #EPRT1,EPRTSW           ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
021574 005037 003154                    CLR     SIFLAG                  ;CLEAR "SOFT INIT" FLAG
021600 005037 003134                    CLR     KTENABLE                ;CLEAR TEST ABOVE 28K FLAG
021604 005037 002302                    CLR     RAMSIZ                  ;CLEAR RAM SIZE FOR RAMERR ROUTINE
021610                                  READEF  #EF.CONTINUE
021610 012700 000036                    MOV     #EF.CONTINUE,R0
021614 104447                           TRAP    C$REFG
021616                                  BNCOMPLETE 1$
021616 103023                           BCC     1$
021620 023737 002202 002012             CMP     UNITN,L$UNIT            ;UNIT IN RANGE?
021626 103070                           BHIS    4$                      ;BR IF NO.
021630 005737 003112                    TST     DUFLG                   ;DROPPED UNIT?
021634 100472                           BMI     NXTU                    ;BR IF YES
021636 013701 002202                    MOV     UNITN,R1
021642 006301                           ASL     R1
021644 005761 003176                    TST     ERTABL(R1)
021650 001516                           BEQ     SETU
021652 032761 040000 003176             BIT     #BIT14,ERTABL(R1)       ;DROPPED?
021660 001060                           BNE     NXTU
021662                                  EXIT    INIT                    ;DO NOTHING IF "CONTINUE".
021662 104432                           TRAP    C$EXIT
021664 000416                           .WORD   L10030-.
021666                          1$:     READEF  #EF.NEW
021666 012700 000035                    MOV     #EF.NEW,R0
021672 104447                           TRAP    C$REFG
021674                                  BNCOMPLETE NXTU                 ;TAKE NEXT UNIT IF NOT NEW PASS.
021674 103052                           BCC     NXTU
021676                                  READEF  #EF.START
021676 012700 000040                    MOV     #EF.START,R0
021702 104447                           TRAP    C$REFG
021704                                  BCOMPLETE 2$
021704 103404                           BCS     2$
021706                                  READEF  #EF.RESTART
021706 012700 000037                    MOV     #EF.RESTART,R0
021712 104447                           TRAP    C$REFG
021714                                  BNCOMPLETE 31$
021714 103031                           BCC     31$
021716                          2$:                                     ;1ST PASS, BUS-INIT...
021716                                  BRESET                          ;BUS RESET.
021716 104433                           TRAP    C$RESET
```

```
                                                CLR     TSTCNT                  ;NUMBER OF TESTS RUN IN PASS
                                                CLR     FATFLG                  ;CLEAR FATAL ERROR COUNT
                                                CLR     T23A                    ;CLEAR 11/23A FLAG
                                                CLR     T23B                    ;CLEAR 11/23B FLAG
                                        ;       MOV     #340,-(SP)
                                        ;       MOV     #20$,-(SP)              ;RETURN TO DEBUGGER
                                        ;       JMP     0.ODT           ;;@@ENTER THE DEBUGGER
                                                CLR     SKIPT           ;CLEAR THE SUBTEST "SKIPPER"
                                        20$:
                                                MOV     #-1,QVP                 ;...QUICK VERIFY...
                                                JSR     PC,ENVIRN               ;SET ENVIRONMENT.
                                                JSR     PC,KTINIT               ;INITIALIZE KT MEMORY MANAGEMENT
                                                MOV     #ERTABL,R0
                                        30$:    CLR     (R0)+                   ;CLEAR THE ERROR TABLE
                                                CMP     R0,#ERTABE
                                                BLO     30$
                                                BR      4$
                                        31$:    CLR     QVP
                                                JMP     PASRPT                  ;GO REPORT THE STATUS

                                        4$:
                                        NEWPAS: MOV     #-1,UNITN               ;INIT UNIT NUMBER...
                                                CLR     DEVCNT                  ;CLEAR COUNT OF DEVICES RUNNING
                                        NXTU:   BREAK
                                                TRAP    C$BRK
                                                INC     UNITN                   ;...AND SET NEXT UNIT NUMBER.
                                                CMP     UNITN,L$UNIT
                                                BLO     SETU
                                                MOV     #-1,DUFLG
                                                BR      11$
                                                DOCLN
                                                TRAP    C$DCLN                  ;ABORT, NO MORE UNITS.
                                        11$:    NOP
                                        PASRPT:
                                                CMP     L$UNIT,#1               ;HOW MANY UNITS SELECTED?
                                                BLOS    NEWPAS                  ;BR IF ONLY 1
                                                TST     DEVCNT                  ;ARE ANY STILL RUNNING?
                                                BEQ     NEWPAS                  ;BR IF NO
                                                RFLAGS  R0
                                                TRAP    C$RFLA
                                                BIT     #ISR,R0                 ;SHOULD WE PRINT STATISTICS
                                                BNE     NEWPAS                  ;BR IF NO

                                                DORPT
                                                TRAP    C$DRPT
                                                BR      NEWPAS
                                        10$:

                                        SETU:   GPHARD  UNITN,R0                ;GET UNIT N P-TABLE POINTER.
                                                MOV     UNITN,R0
                                                TRAP    C$GPHRD
                                                BNCOMPLETE NXTU                 ;BR IF UNIT NOT AVAILABLE.
                                                BCC     NXTU
                                                CLR     DUFLG                   ;CLEAR "DROPPED" FLAG.
                                                INC     DEVCNT
                                                MOV     (R0)+,R1                ;GET 1ST REGISTER ADDRESS.
                                                MOV     R1,CSRADDR              ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
```

```
                                                ;GET VECTOR ADDRESS.
                                                ;GET INTERRUPT PRIORITY
                                                ;SET INTERRUPT PRIORITY.
                                                ;SET INTERRUPT VECTOR POINTER...
                                                ;...VECTOR...
                                                ;...AND PRIORITY.

                                                ;1ST PASS ??
                                                ;NO, SKIP THE PASS 1 STUFF.

                                                ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
                                                ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.

                                                ;SAY DEVICE RUNNING
                                                ;CLEAR ERROR EXTENSION FLAG.
                                                ;ARE WE TESTING MULTIPLE UNITS?
                                                ;BR IF NO.
                                                ;YES -- GET OPERATOR FLAGS.

                                                ;SHOULD WE PRINT UNIT #?
                                                ;BR IF NOT.
                                                ;PRINT THE UNIT #

                                                ;ADDRESS OF FIRST REGISTER
                                                ;START OF REGISTERS
                                                ;ADDRESS OF TSSR REGISTER
                                                ;TEST BOTH CONTROLLER REGISTERS...
                                                ;...AND BR IF ALL OK.
                                                ;FLAG DEVICE AS NON-EXISTENT
                                                ;DROP THIS UNIT.

                                                ;ENABLE INTERRUPTS.

 157 022304      045      116      045  PUNIT:  .ASCIZ  /%N%N%A***** TESTING UNIT %D2%A *****/
```

```
                                        .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.
                                ;--
022352                                  BGNAU
022352                          L$AU::
022352  010001                          MOV     R0,R1                   ; GET UNIT TO BE ADDED (R0)
022354  006301                          ASL     R1                      ; MAKE IT A WORD INDEX
022356  052761  100000  003176          BIS     #100000,ERTABL(R1)      ; SET THE "ACTIVE" BIT
022364  042761  040000  003176          BIC     #40000,ERTABL(R1)       ; CLEAR THE "DROPPED" BIT
022372                                  PRINTF  #1$,R0
022372  010046                          MOV     R0,-(SP)
022374  012746  022420                  MOV     #1$,-(SP)
022400  012746  000002                  MOV     #2,-(SP)
022404  010600                          MOV     SP,R0
022406  104417                          TRAP    C$PNTF
022410  062706  000006                  ADD     #6,SP
022414                                  EXIT    AU
022414  000167                          .WORD   J$JMP
022416  000026                          .WORD   L10031-2-.
022420     045     116     045  1$:     .ASCIZ  /%N%A  UNIT %D%A ADDED/
                                        .EVEN


022446                                  ENDAU           ; UNUSED.
022446                          L10031:
022446  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.


022450                                  BGNDU
022450                          L$DU::
022450  012737  177777  003112          MOV     #-1,DUFLG
022456  010001                          MOV     R0,R1
022460  006301                          ASL     R1
022462  052761  140000  003176          BIS     #140000,ERTABL(R1)      ; SAY DROPPED
022470  000240  000240  000240          240,240,240                     ; ?????????
022476                                  PRINTF  #1$,R0
022476  010046                          MOV     R0,-(SP)
022500  012746  022524                  MOV     #1$,-(SP)
022504  012746  000002                  MOV     #2,-(SP)
022510  010600                          MOV     SP,R0
022512  104417                          TRAP    C$PNTF
022514  062706  000006                  ADD     #6,SP
022520                                  EXIT    DU
022520  000167                          .WORD   J$JMP
022522  000030                          .WORD   L10032-2-.
```

```
                                1$:     .ASCIZ  /%N%A  UNIT %D%A DROPPED/
                                        .EVEN
                                        ENDDU
                                L10032:
                                        TRAP    C$DU
                                ;++
                                ; AUTO-DROP CODE SECTION.
                                ;--
                                        BGNAUTO
                                L$AUTO::
022556  013705  002206                  MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
022562  012703  000550                  MOV     #360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
022566  004737  016250          10$:    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
022572  103420                          BCS     20$                     ;LEAVE WHEN SSR IS SET
022574                                  DELAY   250.                    ;WAIT FOR .25 SECONDS
022574  012727  000372                  MOV     #250.,(PC)+
022600  000000                          .WORD   0
022602  013727  002116                  MOV     L$DLY,(PC)+
022606  000000                          .WORD   0
022610  005367  177772                  DEC     -6(PC)
022614  001375                          BNE     .-4
022616  005367  177756                  DEC     -22(PC)
022622  001367                          BNE     .-20
022624  005303                          DEC     R3                      ;BUMP COUNTER DOWN
022626  001357                          BNE     10$                     ;KEEP GOING
022630  004737  017202                  JSR     PC,CKDROP               ;TRY AND DROP UNIT
022634                          20$:
                                        ENDAUTO                 ; UNUSED.
                                L10033:
022634  104461                          TRAP    C$AUTO
```

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

                                ;++
                                ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
                                ; EXECUTED AT THE END OF EACH PASS (OR SUB-PASS).
                                ; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
                                ;--
022636                                  BGNCLN
022636                          L$CLEAN::
022636  013705  002206                  MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
022642  005737  003112                  TST     DUFLG           ;"DROPPED" FLAG IS SET ON...
022646  100405                          BMI     1$              ;...AND GROSS CONTROLLER FAULT...
                                                                ;...DON'T TRY TO XCT CLEANUP CODE.

022650  012765  000000  000002          MOV     #0,TSSR(R5)     ;DO SOFT INIT
022656  004737  016250                  JSR     PC,WAITF
022662                          1$:
022662                          2$:     ENDCLN
022662                          L10034:
022662  104412                          TRAP    C$CLEAN
                                ;++
                                ; THE REPORT CODING SECTION CONTAINS THE
                                ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;--
022664                                  BGNRPT
022664                          L$RPT::
022664                                  PRINTS  #DEVSUM
022664  012746  023126                  MOV     #DEVSUM,-(SP)
022670  012746  000001                  MOV     #1,-(SP)
022674  010600                          MOV     SP,R0
022676  104416                          TRAP    C$PNTS
022700  062706  000004                  ADD     #4,SP
022704  010246                          MOV     R2,-(SP)
022706  010346                          MOV     R3,-(SP)
022710  010446                          MOV     R4,-(SP)
022712  012704  003176                  MOV     #ERTABL,R4      ; GET START OF ERROR TABLE.
022716  005003                          CLR     R3              ; CLEAR UNIT NUMBER
022720  011402                  1$:     MOV     (R4),R2         ; GET ERROR TABLE ENTRY & TEST IT.
022722  001467                          BEQ     4$              ; ZERO IF UNIT NOT RUN
022724  100066                          BPL     4$
022726  032702  040000                  BIT     #BIT14,R2       ; WAS UNIT DROPPED?
022732  001015                          BNE     2$              ; BR IF YES
022734  042702  170000                  BIC     #^C7777,R2      ; GET ERROR COUNT FIELD
022740                                  PRINTS  #DEVONL,R3,R2   ; PRINT
022740  010246                          MOV     R2,-(SP)
022742  010346                          MOV     R3,-(SP)
022744  012746  023163                  MOV     #DEVONL,-(SP)
022750  012746  000003                  MOV     #3,-(SP)
022754  010600                          MOV     SP,R0
022756  104416                          TRAP    C$PNTS
022760  062706  000010                  ADD     #10,SP
022764  000446                          BR      4$
022766  020227  160000          2$:     CMP     R2,#160000      ; WAS UNIT NON-EXISTENT?
022772  001012                          BNE     3$              ; BR IF NO
022774                                  PRINTS  #DEVNXR,R3
022774  010346                          MOV     R3,-(SP)
022776  012746  023233                  MOV     #DEVNXR,-(SP)
```

```
                023002  012746  000002                  MOV     #2,-(SP)
                023006  010600                          MOV     SP,R0
                023010  104416                          TRAP    C$PNTS
                023012  062706  000006                  ADD     #6,SP
254             023016  000431                          BR      4$
255             023020  020227  160001          3$:     CMP     R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
256             023024  001012                          BNE     30$             ; BR IF NO.
257             023026                                  PRINTS  #DEVNRD,R3
                023026  010346                          MOV     R3,-(SP)
                023030  012746  023315                  MOV     #DEVNRD,-(SP)
                023034  012746  000002                  MOV     #2,-(SP)
                023040  010600                          MOV     SP,R0
                023042  104416                          TRAP    C$PNTS
                023044  062706  000006                  ADD     #6,SP
258             023050  000414                          BR      4$
259             023052  042702  170000          30$:    BIC     #^C7777,R2
260             023056                                  PRINTS  #DEVDRO,R3,R2
                023056  010246                          MOV     R2,-(SP)
                023060  010346                          MOV     R3,-(SP)
                023062  012746  023376                  MOV     #DEVDRO,-(SP)
                023066  012746  000003                  MOV     #3,-(SP)
                023072  010600                          MOV     SP,R0
                023074  104416                          TRAP    C$PNTS
                023076  062706  000010                  ADD     #10,SP
261             023102  062704  000002          4$:     ADD     #2,R4
262             023106  005203                          INC     R3
263             023110  020427  003376                  CMP     R4,#ERTABE
264             023114  103701                          BLO     1$
265             023116  012604                          MOV     (SP)+,R4
266             023120  012603                          MOV     (SP)+,R3
267             023122  012602                          MOV     (SP)+,R2
268             023124                                  ENDRPT          ; UNUSED.
                023124                          L10035:
                023124  104425                          TRAP    C$RPT
269
270
271             023126      045     116     045 DEVSUM: .ASCIZ  /%N%ADEVICE STATUS SUMMARY:%N/
272             023163      045     101     040 DEVONL: .ASCIZ  /%A  UNIT %D3%A   ONLINE,  ERRORS = %D%N/
273             023233      045     101     040 DEVNXR: .ASCIZ  /%A  UNIT %D3%A   DROPPED, NON-EXISTENT REGISTER%N/
274             023315      045     101     040 DEVNRD: .ASCIZ  /%A  UNIT %D3%A   DROPPED, NOT READY AT STARTUP%N/
275             023376      045     101     040 DEVDRO: .ASCIZ  /%A  UNIT %D3%A   DROPPED, ERRORS = %D%N/
276                                                 .EVEN
277
278             023446                              ENDMOD
279
280
```

```
                                .TITLE  TSV5 - HARDWARE TESTS
                                BGNMOD  TSV5
023446                  TSV5::
023446
```

```
                        .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
                ;--


023446                  BGNTST
023446                                                                  T1::
023446  012700  024102          MOV     #TST13ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
023452  004737  016510          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
023456  012737  000012  002216  MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
023464                  T13LOOP:
023464  004737  024356          JSR     PC,T13REST              ;SET PACKET TO START-UP VALUES

023470  012703  002764          MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
023474  012704  024040          MOV     #T13PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
023500  012764  000010  000006  MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
023506                  5$:
023506  004737  015774          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
023512  103405                  BCS     10$                     ;BR IF SOFT INIT OKAY
023514  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
023516                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
023516  104455                                                                  TRAP    C$ERDF
023520  000144                                                                  .WORD   100
023522  003652                                                                  .WORD   SFIERR
023524  012034                                                                  .WORD   SFIMSG

                ;Do WRITE CHARACTERISTICS command.
023526  005037  002222  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
023532  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
023536  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
023542                          FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
023556  103407                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
023560  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
023562                          NEXT.ERRNO
023562                  12$:    ERRDF   ERRNO,T13SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
023562  104455                                                                  TRAP    C$ERDF
023564  000145                                                                  .WORD   101
023566  024267                                                                  .WORD   T13SSR
023570  012046                                                                  .WORD   PKTSSR
023572  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
023576                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

```
023576  104406                                                          TRAP    C$CLP1
 79 023600  016501  000002          MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 80 023604  012702  000200          MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
 81 023610  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 82 023614  001402                  BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 83 023616  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 84
 85                             ;If the NBA bit in the TSSR register is NOT=0 then Print Error.
 86 023622                      25$:
 87 023622                              FORCERROR       27$                     ;a@D
 88 023636  020201                      CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 89 023640  001404                      BEQ     30$                     ;OKAY IF MATCH
 90 023642                              NEXT.ERRNO
 91 023642                      27$:    ERRHRD  ERRNO,T13NBA,PKTSSR     ;NBA NOT ZERO
    023642  104456                                                          TRAP    C$ERHRD
    023644  000146                                                          .WORD   102
    023646  024214                                                          .WORD   T13NBA
    023650  012046                                                          .WORD   PKTSSR
 92 023652                      30$:    CKLOOP                          ;LOOP ON ERROR ?
    023652  104406                                                          TRAP    C$CLP1
 93
 94                             ;Write to TSSR register to soft initialize the controller
 95 023654                      40$:
 96 023654  004737  015774              JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
 97 023660                              FORCERROR       42$                     ;a@D
 98 023674  103405                      BCS     50$                     ;BR IF SOFT INIT OKAY
 99 023676  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
100 023700                              NEXT.ERRNO
101 023700                      42$:    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
    023700  104455                                                          TRAP    C$ERDF
    023702  000147                                                          .WORD   103
    023704  003652                                                          .WORD   SFIERR
    023706  012034                                                          .WORD   SFIMSG
102
103                             ;If controller RAM 310-377 NOT=0 then Print Error
104 023710  012704  000310      50$:    MOV     #310,R4                 ;START WITH LOC 310
105 023714  005002                      CLR     R2                      ;MEMORY EXPECTED SHOULD BE 000000
106 023716  105065  000000              CLRB    TSDB(R5)                ;SET MAINTENANCE MODE
107 023722  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR READY
108 023726  010465  000000      60$:    MOV     R4,TSDB(R5)             ;SELECT RAM ADDRESS
109 023732  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR READY
110 023736  116501  000000              MOVB    TSBA(R5),R1             ;READ LOC CONTENTS
111 023742                              FORCERROR       62$,NOTSSR              ;a@D
112 023752  120102                      CMPB    R1,R2                   ;CHECK MEMORY FOR 000000
113 023754  001406                      BEQ     70$                     ;BRANCH IF DATA OKAY
114 023756                              NEXT.ERRNO
115 023756                      62$:    ERRDF   ERRNO,T13MEM,RAMEXP     ;MEMORY NOT ZERO AFTER INIT.
    023756  104455                                                          TRAP    C$ERDF
    023760  000150                                                          .WORD   104
    023762  024155                                                          .WORD   T13MEM
    023764  015530                                                          .WORD   RAMEXP
116 023766  005237  002222              INC     FATFLG                  ;SET THE FATAL ERROR FLAG
117 023772                      70$:    CKLOOP
    023772  104406                                                          TRAP    C$CLP1
118 023774                              ESCAPE  TST                     ;EXIT ON FATAL ERROR
    023774  104410                                                          TRAP    C$ESCAPE
    023776  000426                                                          .WORD   L10036-.
```

```
                                        82$:    INC     R4                      ;LOOK AT NEXT RAM LOC.
024000  005204
                                                CMP     R4,#400                 ;AT TOP OF RAM ADDRESS SPACE
024002  020427  000400
                                                BNE     60$                     ;BRANCH TILL ALL MEMORY TESTED
024006  001347
                                                TST     FATFLG                  ;ANY FATAL ERRORS ?
024010  005737  002222
                                                BEQ     160$                    ;BRANCH IF NOT
024014  001402
                                                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
024016  004737  017202
                                        160$:   JSR     PC,TSTLOOP              ;DONE ALL ITERATIONS?
024022  004737  016456
                                                BCC     165$                    ;BR IF YES
024026  103002
                                                JMP     T13LOOP                 ;LOOP UNTIL ITERATION COUNT DONE
024030  000137  023464
                                        165$:
024034
                                                EXIT    TST
024034
                                                                                TRAP    C$EXIT
024034  104432
                                                                                .WORD   L10036-.
024036  000366
```

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

024040                                  T13PACKET:                      ;COMMAND PACKET FOR TEST
024040  100004                                .WORD   100004            ;WRITE CHARACTERISTICS COMMAND, WITH ACK
024042  024050                                .WORD   T13DATA           ;ADDRESS OF CHARACTERISTICS BLOCK
024044  000000                                .WORD   0
024046  000010                                .WORD   8.                ;STARTING VALUE OF BLOCK SIZE

024050                                  T13DATA:                                ;CHARACTERISTICS DATA BLOCK
024050  024062                                .WORD   T13BFR            ;ADDRESS OF MESSAGE BUFFER
024052  000000                                .WORD   0
024054  000016                                .WORD   14.               ;LENGTH OF MESSAGE BUFFER
024056  000000  000000                       .WORD   0,0

024062                                  T13BFR: .BLKW   8.              ;MESSAGE BUFFER
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-

024102      111     156     151     TST13ID:        .ASCIZ  'Initialization After WRITE CHARACTERISTICS'
024155      111     156     143     T13MEM: .ASCIZ  'Incorrect RAM Data After Init'
                                            .EVEN
024214      127     122     111     T13NBA: .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
024267      103     157     156     T13SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
```

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

                                        .EVEN

                                        T13REST:
                                                SAVREG                          ;SAVE THE REGISTERS
024362  012701  024040                          MOV     #T13PACKET,R1           ;START OF THE PACKET
024366  012721  100004                          MOV     #100004,(R1)+           ;WRITE CHARACTERISTICS WITH ACK
024372  012721  024050                          MOV     #T13DATA,(R1)+          ;ADDRESS OF CHAR DATA BLOCK
024376  005021                                  CLR     (R1)+                   ;EXTENDED ADDRESS
024400  012721  000010                          MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
024404  012721  024062                          MOV     #T13BFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
024410  005021                                  CLR     (R1)+
024412  012721  000016                          MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
024416  005021                                  CLR     (R1)+
024420  005011                                  CLR     (R1)
024422  000207                                  RTS     PC                      ;RETURN
024424                                          ENDTST
024424                                                                  L10036:
024424  104401                                                                  TRAP    C$ETST
```

```
                        .SBTTL  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.
                    ;
                    ;-
024426                      BGNTST
024426                                                          T2::
024426 012700 026241        MOV     #TST14ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
024432 004737 016510        JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
024436 012737 000024 002216 MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
024444                  T14LOOP:
024444                      BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
024444                                                                  T2.1:
024444 104402                                                           TRAP    C$BSUB
024446 004737 026306        JSR     PC,T14REST              ;SET PACKET TO INITIAL VALUES
024452 004737 026344        JSR     PC,T14RST               ;SET PACKET TO INITIAL VALUES
024456                      SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
024456 012700 000000                                                    MOV     #PRI00,R0
024462 104441                                                           TRAP    C$SPRI
024464                  5$:
024464                      BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
024464 104404                                                           TRAP    C$BSEG
024466 004737 015774        JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
024472 103405               BCS     10$                     ;BR IF SOFT INIT = OK
024474 010001               MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
024476                      ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
024476 104455                                                           TRAP    C$ERDF
024500 000311                                                           .WORD   201
024502 003652                                                           .WORD   SFIERR
024504 012034                                                           .WORD   SFIMSG
024506                  10$:
024506 012704 025600        MOV     #T14PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
024512 004737 010662        JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
024516 103405               BCS     11$                     ;BR, IF COMMAND ISSUED OK
024520 010001               MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
024522                      ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
024522 104456                                                           TRAP    C$ERHRD
024524 000312                                                           .WORD   202
024526 005056                                                           .WORD   WRTMSG
024530 012034                                                           .WORD   SFIMSG
024532                  11$:
024532 005037 002222        CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
024536 005037 002224        CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
024542 012704 025160        MOV     #T14PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
024546 010465 000000        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
024552 004737 016336        JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
024556 103407               BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
024560 010001               MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
024562                      ERRDF   ERRNO,T14SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
024562 104455                                                           TRAP    C$ERDF
024564 000313                                                           .WORD   203
```

```
                024566  025773                                                  .WORD   T14SSR
                024570  012046                                                  .WORD   PKTSSR
244             024572  005237  002222          INC     FATFLG          ;SET FATAL ERROR FLAG
245             024576                  15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
                024576  104406                                                  TRAP    C$CLP1
246             024600                          ESCAPE  SEG             ;BY-PASS SUBTEST IF FATAL ERROR
                024600  104410                                                  TRAP    C$ESCAPE
                024602  000074                                                  .WORD   10000$-.
247             024604  005737  002224          TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
248             024610  001004                  BNE     22$             ;BRANCH IF YES
252             024612                          ERRHRD  ERRNO,T14NINT,PKTSSR
                024612  104456                                                  TRAP    C$ERHRD
                024614  000314                                                  .WORD   204
                024616  026063                                                  .WORD   T14NINT
                024620  012046                                                  .WORD   PKTSSR
253             024622  016501  000002  22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
254             024626  012702  000200          MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
255             024632  032701  000100          BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
256             024636  001402                  BEQ     25$             ;BRANCH IF NOT OFF-LINE
257             024640  052702  000100          BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
258             024644  020201          25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
259             024646  001404                  BEQ     30$             ;OKAY IF MATCH
263             024650                          ERRHRD  ERRNO,T14NBA,PKTSSR     ;NBA NOT ZERO
                024650  104456                                                  TRAP    C$ERHRD
                024652  000315                                                  .WORD   205
                024654  025622                                                  .WORD   T14NBA
                024656  012046                                                  .WORD   PKTSSR
264             024660                  30$:
265             024660  004737  011114  35$:    JSR     PC,CKRAM        ;CHECK RAM TO MEMORY
266             024664  103405                  BCS     59$             ;RAM OK GO ON
270             024666                          ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
                024666  104456                                                  TRAP    C$ERHRD
                024670  000316                                                  .WORD   206
                024672  004745                                                  .WORD   PKTRAM
                024674  015510                                                  .WORD   RAMERR
271             024676                          ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                024676                                                  10000$:
                024676  104405                                                  TRAP    C$ESEG
272
273
274             024700                  59$:    ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                024700                                                  L10040:
                024700  104403                                                  TRAP    C$ESUB
275
276             024702  005737  002222          TST     FATFLG          ;ANY FATAL ERRORS ?
277             024706  001402                  BEQ     60$             ;BRANCH IF NOT
278             024710  004737  017202          JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
279             024714                  60$:
```

```
                                        ;+
                                        ;
                                        ;TEST 2, SUBTEST 2
                                        ;
                                        ;CHECK THAT NON-ZERO MODE BITS BEING SET CAUSES
                                        ;WRITE SUBSYSTEM MEMORY COMMAND TO BE REJECTED
                                        ;
                                        ;-

                                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
                                                                                                        T2.2:
024714  104402                                                                                                  TRAP    C$BSUB

                                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
024716  012700  000000                                                                                          MOV     #PRI00,R0
024722  104441                                                                                                  TRAP    C$SPRI
024724  012704  025600        5$:     MOV     #T14PK2,R4              ;GET THE ADDRESS OF COMMAND PACKET
024730  004737  026306                JSR     PC,T14REST              ;RESTORE PACKET TO STARTING VALUES
024734  004737  026344                JSR     PC,T14RST               ;RESTORE PACKET TO STARTING VALUES
024740                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
024740  104404                                                                                                  TRAP    C$BSEG

024742  004737  015774                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
024746  103405                        BCS     10$                     ;BR IF SOFT INIT = OK
024750  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
024752                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
024752  104455                                                                                                  TRAP    C$ERDF
024754  000317                                                                                                  .WORD   207
024756  003652                                                                                                  .WORD   SFIERR
024760  012034                                                                                                  .WORD   SFIMSG
024762                          10$:
024762  004737  010662                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
024766  103405                        BCS     11$                     ;BR, IF COMMAND ISSUED OK
024770  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
024772                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE SUBSYSTEM MEMORY FAILED
024772  104456                                                                                                  TRAP    C$ERHRD
024774  000320                                                                                                  .WORD   208
024776  005056                                                                                                  .WORD   WRTMSG
025000  012034                                                                                                  .WORD   SFIMSG
025002                          11$:
025002  005037  002224                CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
025006  012704  025160                MOV     #T14PACKET,R4           ;SET-UP WITH WRT SUBSYS MEM PACKET
025012  052714  007000                BIS     #007000,(R4)            ;NON-ZERO COMMAND MODE BITS
025016  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
025022  004737  016250                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
025026  103405                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
025030  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
025032                                  ERRDF   ERRNO,T14SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
025032  104455                                                                                                  TRAP    C$ERDF
025034  000321                                                                                                  .WORD   209
025036  025773                                                                                                  .WORD   T14SSR
025040  012046                                                                                                  .WORD   PKTSSR
025042                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
025042  104406                                                                                                  TRAP    C$CLP1
025044                                  ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
025044  104410                                                                                                  TRAP    C$ESCAPE
025046  000074                                                                                                  .WORD   10000$-.
```

```
 327 025050  005737  002224          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 328 025054  001004                  BNE     22$                     ;BRANCH IF YES
 332 025056                          ERRHRD  ERRNO,T14NINT,PKTSSR
     025056  104456                                                      TRAP    C$ERHRD
     025060  000322                                                      .WORD   210
     025062  026063                                                      .WORD   T14NINT
     025064  012046                                                      .WORD   PKTSSR
 333 025066  016501  000002  22$:   MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 334 025072  012702  100206          MOV     #SC!SSR!TSREJ,R2        ;EXPECTED CONTENTS OF TSSR
 335 025076  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 336 025102  001402                  BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 337 025104  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 338 025110  020201  25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 339 025112  001404                  BEQ     30$                     ;OKAY IF MATCH
 343 025114                          ERRHRD  ERRNO,T142REJ,PKTSSR    ;COMMAND NOT REJECTED
     025114  104456                                                      TRAP    C$ERHRD
     025116  000323                                                      .WORD   211
     025120  025676                                                      .WORD   T142REJ
     025122  012046                                                      .WORD   PKTSSR
 344 025124          30$:
 345 025124  004737  011114  35$:   JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
 346 025130  103405                  BCS     59$                     ;RAM OK GO ON
 350 025132                          ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
     025132  104456                                                      TRAP    C$ERHRD
     025134  000324                                                      .WORD   212
     025136  004745                                                      .WORD   PKTRAM
     025140  015510                                                      .WORD   RAMERR
 351 025142                          ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     025142                                                          10000$:
     025142  104405                                                      TRAP    C$ESEG
 352
 353 025144          59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     025144                                                          L1004i:
     025144  104403                                                      TRAP    C$ESUB
```

```
355 025146                              EXIT    TST             ;ALL DONE THIS TEST
    025146  104432                                                      TRAP    C$EXIT
    025150  001246                                                      .WORD   L10037-.
356
357                                 ;+
358                                 ;LOCAL STORAGE FOR THIS TEST
359                                 ;-
360
362         025160                          .=<.+10>&177770                 .
364 025160                          T14PACKET:                      ;COMMAND PACKET FOR TEST
365 025160  100206                          .WORD   100206          ;WRITE SUBSYSTEM MEM. COMMAND, WITH IE, ACK
366 025162  025170                          .WORD   T14DATA         ;ADDRESS OF CHARACTERISTICS BLOCK
367 025164  000000                          .WORD   0
368 025166  000006                          .WORD   6.              ;STARTING VALUE OF BLOCK SIZE
369 025170                          T14DATA:                        ;CHARACTERISTICS DATA BLOCK
370 025170     000                  T14BS0: .BYTE   0               ;BSEL0 BYTE
371 025171     000                  T14BS1: .BYTE   0               ;BSEL1 BYTE
372 025172  000000                  T14BS2: .WORD   0               ;BSEL1 WORD
373 025174  000000                          .WORD   0               ;DATA
374 025176                          T14BFR: .BLKW   128.            ;MESSAGE BUFFER
375
376
378         025600                          .=<.+10>&177770
380 025600                          T14PK2:                         ;COMMAND PACKET FOR TEST
381 025600  100204                          .WORD   100204          ;WRITE CHARA. MEM. CMND., WITH IE, ACK
382 025602  025610                          .WORD   T14DTA          ;ADDRESS OF SELECT DATA BLOCK
383 025604  000000                          .WORD   0
384 025606  000010                          .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
385
386
387 025610                          T14DTA:                         ;SELECT DATA BLOCK
388 025610  025176                          .WORD   T14BFR          ;ADDRESS OF MESSAGE BUFFER
389 025612  000000                          .WORD   0
390 025614  000400                          .WORD   256.            ;LENGTH OF MESSAGE BUFFER
391 025616  000000  000000                  .WORD   0,0
392
393
394                                 ;+
395                                 ;LOCAL TEXT MESSAGES FOR TEST
396                                 ;-
397
398 025622    127    122    111  T14NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
399 025676    127    122    111  T142REJ:        .ASCIZ  'WRITE SUBSYSTEM MEMORY Not Rejected With Non-Zero Mode Field'
400 025773    103    157    156  T14SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
401 026063    105    170    160  T14NINT:        .ASCIZ  'Expected Interrupt Not Received On WRITE SUBSYSTEM MEMORY'
402 026155    111    156    143  T14TSBA:        .ASCIZ  'Incorrect TSBA Address After WRITE SUBSYSTEM MEMORY'
403 026241    102    141    163  TST14ID:        .ASCIZ  'Basic WRITE SUBSYSTEM MEMORY Command'
404                                         .EVEN
405
```

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

026306                  T14REST:
026306                          SAVREG                          ;SAVE THE REGISTERS
026312  012701  025160          MOV     #T14PACKET,R1           ;START OF THE PACKET
026316  012721  100206          MOV     #100206,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK, IE
026322  012721  025170          MOV     #T14DATA,(R1)+          ;ADDRESS OF DATA BLOCK
026326  005021                  CLR     (R1)+                   ;EXTENDED ADDRESS
026330  012721  000006          MOV     #6.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
026334  005021                  CLR     (R1)+                   ;CLEAR BSEL0 AND BSEL1
026336  005021                  CLR     (R1)+                   ;CLEAR SEL2
026340  005011                  CLR     (R1)                    ;CLEAR DATA AREA
026342  000207                  RTS     PC                      ;RETURN

026344                  T14RST:
026344                          SAVREG                          ;SAVE THE REGISTERS
026350  012701  025600          MOV     #T14PK2,R1              ;START OF THE PACKET
026354  012721  100204          MOV     #100204,(R1)+           ;WRITE CHARA. WITH ACK, IE
026360  012721  025610          MOV     #T14DTA,(R1)+           ;ADDRESS OF CHARAISTICS DATA BLOCK
026364  005021                  CLR     (R1)+                   ;EXTENDED ADDRESS
026366  012721  000010          MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
026372  012721  025176          MOV     #T14BFR,(R1)+           ;MESSAGE BUFFER ADDRESS
026376  005021                  CLR     (R1)+
026400  012721  000400          MOV     #256.,(R1)+             ;LENGTH OF MESSAGE BUFFER
026404  005021                  CLR     (R1)+
026406  005011                  CLR     (R1)
026410  005037  025176          CLR     T14BFR                  ;CLEAR 1ST LOC IN MESSAGE BUFFER
026414  000207                  RTS     PC                      ;RETURN
026416                  ENDTST
026416
026416  104401                                  L10037:
                                                        TRAP    C$ETST
```

```
444
```

```
                        .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
                ;
                ;--

026420                  BGNTST
026420                                                          T3::
026420  012700  030450          MOV     #TST12ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
026424  004737  016510          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
026430  012737  000012  002216  MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
026436  005237  003150          INC     T3BFLG                  ;SET TEST FLAG
026442  004737  021276          JSR     PC,MEMCK                ;CHECK MEMORY

026446  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
                ;
                ;       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
                ;--
                ;
026446          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
                                                        T3.1:
026446  104402                                          TRAP    C$BSUB
                ;
                ;
                ;Write to TSSR to soft initialize
026450  004737  015774          JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
026454  103405                  BCS     15$             ;BR IF SOFT INIT = OK
026456                          NEXT.ERRNO
026456  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
026460                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
026460  104455                                                  TRAP    C$ERDF
026462  000455                                                  .WORD   301
026464  003652                                                  .WORD   SFIERR
026466  012034                                                  .WORD   SFIMSG
                ;
                ;Do a WRITE CHARACTERISTICS to setup a message buffer
026470          15$:
026470  012704  030240          MOV     #T12PACKET,R4   ;GET THE ADDRESS OF COMMAND PACKET
026474  004737  031620          JSR     PC,T12SWRT      ;RESTORE PACKET TO STARTING VALUES
026500  005037  003134          CLR     KTENABLE        ;TURN OFF KT-11
026504  010465  000000          MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
026510  004737  016336          JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
026514                          FORCERROR       17$
026530  103405                  BCS     20$             ;BR IF SSR SET IN CHKTSSR
026532  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
026534                          NEXT.ERRNO
026534          17$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
026534  104455                                                  TRAP    C$ERDF
```

```
026536  000456                                                          .WORD   302
026540  030552                                                          .WORD   T12WRTSSR
026542  012046                                                          .WORD   PKTSSR
536
537                             ;Verify a Get Status can be fetched from each address
538                             ;Get a valid modulo-4 test address
539                             ;Do a GET STATUS command from the test address
540 026544  005037  002222      20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
541 026550  005037  030310              CLR     T12KT                   ;TEST ABOVE 28K SWITCH
542 026554  012702  030314              MOV     #T12BLK,R2              ;POINT TO TEST PATTERN TABLE
543 026560                      T121LOOP:
544 026560  005037  003134              CLR     KTENABLE                ;TURN OFF ABOVE 28K TEST FLAG
545 026564  012201                      MOV     (R2)+,R1                ;GET TEST PATTERN ADDRESS
546 026566  005000                      CLR     R0                      ;ASSUME NO TEST ABOVE 28K
547 026570  005737  030310              TST     T12KT                   ;TEST ABOVE 28K THIS TIME?
548 026574  001407                      BEQ     25$                     ;BR IF NO
549 026576  016200  177776              MOV     -2(R2),R0               ;GET TEST PATTERN AGAIN
550 026602  042700  177774              BIC     #^C<A1716>,R0           ;SAVE 18 BIT ADDRESS ONLY
551 026606  012737  000001  003134      MOV     #1,KTENABLE             ;TURN ON ABOVE 28K TEST FLAG
552 026614  004737  031316      25$:    JSR     PC,T12CONVERT           ;CONVERT TEST PATTERN TO TEST ADDRESS
553 026620  103034                      BCC     65$                     ;BR IF INVALID PACKET ADDRESS
554 026622  013704  030304              MOV     T12LOADD,R4             ;COPY CURRENT PACKET LOW ADDRESS
555 026626  013703  030302              MOV     T12HIADD,R3             ;COPY CURRENT PACKET HIGH ADDRESS
556 026632  004737  031666              JSR     PC,T12SETGET            ;SETUP CURRENT PACKET TO GET STATUS
557 026636  042703  177774              BIC     #^C<A1716>,R3           ;SAVE ADDRESS BITS 17+16
558 026642  050304                      BIS     R3,R4                   ;SETUP 18 BIT PACKET ADDRESS
559 026644  004737  017274              JSR     PC,KTOFF                ;TURN OFF KT-11
560 026650  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
561 026654  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
562 026660                              FORCERROR       32$
563 026674  103405                      BCS     40$                     ;BR IF SSR SET IN CHKTSSR
564 026676  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
565 026700                              NEXT.ERRNO
566 026700                  32$:        ERRDF   ERRNO,T12GETSSR,PKTGETS ;DEVICE FATAL SSR FAILED TO SET
    026700  104455                                                          TRAP    C$ERDF
    026702  000457                                                          .WORD   303
    026704  030476                                                          .WORD   T12GETSSR
    026706  012064                                                          .WORD   PKTGETS
567 026710                  40$:        CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
    026710  104406                                                          TRAP    C$CLP1
568 026712                  65$:
569 026712                              FORCEXIT        80$
570 026722  020227  030446              CMP     R2,#T12TBE              ;DONE ALL TSTBLK TEST PATTERNS?
571 026726  103002                      BHIS    70$                     ;BR IF YES
572 026730  000137  026560              JMP     T121LOOP                ;DO ANOTHER MODULO- 4 ADDRESS
573 026734  005737  030310      70$:    TST     T12KT                   ;DONE ABOVE 28K TESTING TOO?
574 026740  003012                      BGT     80$                     ;BR IF YES
575 026742  005737  003132              TST     KTFLG                   ;ANY MEMORY ABOVE 28K ON SYSTEM?
576 026746  001407                      BEQ     80$                     ;BR IF NO
577 026750  012737  000001  030310      MOV     #1,T12KT                ;SET SWITCH
578 026756  012702  030314              MOV     #T12BLK,R2              ;RESET TEST PATTERN TABLE
579 026762  000137  026560              JMP     T121LOOP                ;DO ABOVE 28K TESTING
580 026766  004737  017274      80$:    JSR     PC,KTOFF                ;TURN OFF KT11
581 026772                              ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
    026772                                                                  L10043:
    026772  104403                                                          TRAP    C$ESUB
582 026774  005737  002222              TST     FATFLG                  ;ANY FATAL ERRORS ?
```

```
583 027000  001402                    BEQ     100$                ;BRANCH IF NOT
584 027002  004737  017202            JSR     PC,CKDROP           ;TRY TO DROP THE UNIT
585 027006                      100$:
586
```

```
                        .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
                ;--

027006                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
027006                                                                  T3.2:
027006  104402                                                                  TRAP    C$BSUB

                ;Write to TSSR to soft initialize
027010  004737  015774          JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
027014  103405                  BCS     15$             ;BR IF SOFT INIT = OK
027016                          NEXT.ERRNO
027016  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
027020                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
027020  104455                                                                  TRAP    C$ERDF
027022  000460                                                                  .WORD   304
027024  003652                                                                  .WORD   SFIERR
027026  012034                                                                  .WORD   SFIMSG

                ;Do a WRITE CHARACTERISTICS to setup a message buffer to compare
027030          15$:
027030  012704  030240          MOV     #T12PACKET,R4   ;GET THE ADDRESS OF COMMAND PACKET
027034  004737  031620          JSR     PC,T12SWRT      ;SET PACKET TO WRITE CHARACTERISTICS
027040  004737  017274          JSR     PC,KTOFF        ;TURN OFF KT-11
027044  010465  000000          MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
027050  004737  016336          JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
027054                          FORCERROR       17$
027070  103405                  BCS     20$             ;BR IF SSR SET IN CHKTSSR
027072  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
027074                          NEXT.ERRNO
```

```
                                17$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
027074  104455                                                                    TRAP    C$ERDF
027076  000461                                                                    .WORD   305
027100  030552                                                                    .WORD   T12WRTSSR
027102  012046                                                                    .WORD   PKTSSR

                                ;Get a valid modulo-4 test address
                                ;Set the packet message buffer to the test address
                                ;Do a WRITE CHARACTERISTICS
027104  005037  002222          20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
027110  012703  030314                  MOV     #T12BLK,R3              ;POINT TO TEST PATTERN TABLE
027114                          T122LOOP:
027114  012301                          MOV     (R3)+,R1                ;GET TEST PATTERN ADDRESS
027116  010100                          MOV     R1,R0                   ;GET ADDRESS ALL "18 BITS"
027120  042700  177774                  BIC     #177774,R0              ;LEAVE ONLY A17 AND A16
027124  042701  000003                  BIC     #3,R1                   ;GET RID OF A17 AND A16
027130  004737  031316                  JSR     PC,T12CONVERT           ;CONVERT TEST PATTERN TO TEST ADDRESS
027134  103402                          BCS     25$                     ;BR IF VALID MESSAGE BUFFER ADDRESS
027136  000137  027234                  JMP     150$                    ;GET ANOTHER TEST PATTERN TO TRY
027142  012704  030240          25$:    MOV     #T12PACKET,R4           ;SET THE COMMAND PACKET ADDRESS
027146  004737  031620                  JSR     PC,T12SWRT              ;SETUP T12PACKET TO WRITE CHAR.
027152  013737  030304  030250          MOV     T12LOADD,T12DATA        ;SETUP LOW ORDER MESSAGE BUFFER ADD.
027160  013737  030302  030252          MOV     T12HIADD,T12DATA+2      ;SETUP HIGH ORDER MESSAGE BUFFER ADD.
027166  004737  017274                  JSR     PC,KTOFF                ;TURN OFF KT-11
027172  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
027176  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
027202                                  FORCERROR       32$
027216  103405                          BCS     50$                     ;BR IF SSR SET IN CHKTSSR
027220  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
027222                                  NEXT.ERRNO
027222                          32$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
027222  104455                                                                  TRAP    C$ERDF
027224  000462                                                                  .WORD   306
027226  030552                                                                  .WORD   T12WRTSSR
027230  012046                                                                  .WORD   PKTSSR
027232                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
027232  104406                                                                  TRAP    C$CLP1
027234                          150$:
027234                                  FORCEXIT        160$
027244  020327  030446                  CMP     R3,#T12TBE              ;DONE ALL TST12BLK TEST PATTERNS?
027250  103002                          BHIS    160$                    ;BR IF YES
027252  000137  027114                  JMP     T122LOOP                ;DO ANOTHER MODULO- 4 ADDRESS
027256  004737  017274          160$:   JSR     PC,KTOFF                ;TURN OFF KT11
027262                                  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
027262                                                                          L10044:
027262  104403                                                                  TRAP    C$ESUB
027264  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
027270  001402                          BEQ     180$                    ;BRANCH IF NOT
027272  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
027276                          180$:
```

```
                        .SBTTL  TEST 3: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS
                ;++
                ; TEST 3: SUBTEST 3:
                ;
                ; SUBTEST DESCRIPTION:
                ;
                ;       This subtest verifies the controller can fetch a
                ;       Write Characteristics data block from all available
                ;       memory locations.
                ;       Write Characteristics commands are executed with
                ;       characteristic data blocks at various memory addresses.
                ;       The various memory addresses are determined by floating
                ;       a 1 then a 0 through the address bits.
                ;
                ; TEST STEPS:
                ;
                ;    BEGIN
                ;       Write to TSSR to soft initialize
                ;
                ;       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
                ;--

027276                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
027276                                                          T3.3:
027276  104402                                                      TRAP    C$BSUB

                ;Write to TSSR to soft initialize
027300  004737  015774          JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
027304  103405                  BCS     20$             ;BR IF SOFT INIT = OK
027306                          NEXT.ERRNO
027306  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
027310                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
027310  104455                                                      TRAP    C$ERDF
027312  000463                                                      .WORD   307
027314  003652                                                      .WORD   SFIERR
027316  012034                                                      .WORD   SFIMSG

                ;Get a valid test address
027320  005037  002222  20$:    CLR     FATFLG          ;CLEAR FATAL ERROR FLAG
027324  005037  030310          CLR     T12KT           ;TEST ABOVE 28K SWITCH
027330  012703  030314          MOV     #T12BLK,R3      ;POINT TO TEST PATTERN TABLE
027334                  T123LOOP:
027334  005037  003134          CLR     KTENABLE        ;TURN OFF ABOVE 28K TEST FLAG
027340  012301                  MOV     (R3)+,R1        ;GET TEST PATTERN ADDRESS
027342  010100                  MOV     R1,R0           ;GET ADDRESS ALL "18 BITS"
027344  042700  177774          BIC     #177774,R0      ;LEAVE ONLY A17 AND A16
027350  042701  000003          BIC     #3,R1           ;GET RID OF A17 AND A16
027354  005737  030310          TST     T12KT           ;TEST ABOVE 28K THIS TIME?
```

```
733 027360  001407                        BEQ     25$                     ;BR IF NO
734 027362  016300  177776                MOV     -2(R3),R0               ;GET TEST PATTERN AGAIN
735 027366  042700  177774                BIC     #^C<A1716>,R0           ;SAVE 18 BIT ADDRESS ONLY
736 027372  012737  000001  003134        MOV     #1,KTENABLE             ;TURN ON ABOVE 28K TEST FLAG
737 027400  004737  031316          25$:  JSR     PC,T12CONVERT           ;CONVERT TEST PATTERN TO TEST ADDRESS
738 027404  103402                        BCS     30$                     ;BR IF VALID TEST ADDRESS
739 027406  000137  027510                JMP     60$                     ;GET NEXT TEST PATTERN
740                                 ;Set the test packet characteristics data pointer to the test address
741 027412  012704  030240          30$:  MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
742 027416  004737  031620                JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
743 027422  013764  030304  000002        MOV     T12LOADD,PKLOW(R4)      ;STORE CHAR. DATA PTR LOW ADDRESS
744 027430  013764  030302  000004        MOV     T12HIADD,PKHI(R4)       ;STORE CHAR. DATA PTR HIGH ADDRESS
745 027436  004737  031730                JSR     PC,T12CHAR              ;STORE EXPECTED DATA IN DATA BLOCK
746                                 ;Do a WRITE CHARACTERISTIC command
747 027442  004737  017274                JSR     PC,KTOFF                ;TURN OFF KT-11
748 027446  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
749 027452  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
750 027456                                FORCERROR       32$
751 027472  103405                        BCS     40$                     ;BR IF SSR SET IN CHKTSSR
752 027474  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
753 027476                                NEXT.ERRNO
754 027476                          32$:  ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
    027476  104455                                                                  TRAP    C$ERDF
    027500  000464                                                                 .WORD   308
    027502  030552                                                                 .WORD   T12WRTSSR
    027504  012046                                                                 .WORD   PKTSSR
755 027506                          40$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
    027506  104406                                                                 TRAP    C$CLP1
756 027510                          60$:
757 027510  020327  030446                CMP     R3,#T12TBE              ;DONE ALL TSTBLK TEST PATTERNS?
758 027514  103002                        BHIS    65$                     ;BR IF YES
759 027516  000137  027334                JMP     T123LOOP                ;DO ANOTHER MODULO- 4 ADDRESS
760 027522  005737  030310          65$:  TST     T12KT                   ;DONE ABOVE 28K TESTING TOO?
761 027526  003012                        BGT     70$                     ;BR IF YES
762 027530  005737  003132                TST     KTFLG                   ;ANY MEMORY ABOVE 28K ON SYSTEM?
763 027534  001407                        BEQ     70$                     ;BR IF NO
764 027536  012737  000001  030310        MOV     #1,T12KT                ;SET SWITCH
765 027544  012703  030314                MOV     #T12BLK,R3              ;RESET TEST PATTERN TABLE
766 027550  000137  027334                JMP     T123LOOP                ;DO ABOVE 28K TESTING
767 027554  004737  017274          70$:  JSR     PC,KTOFF                ;TURN OFF KT11
768 027560                                ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\
    027560                                                                      L10045:
    027560  104403                                                                 TRAP    C$ESUB
769 027562  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
770 027566  001402                        BEQ     75$                     ;BRANCH IF NOT
771 027570  004737  017202                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
772 027574                          75$:
773
```

```
                        .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
                ;--
027574                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
027574                                                                  T3.4:
027574  104402                                                          TRAP    C$BSUB

027576  005737  003136          TST     NXMFLG                  ;GOT ENOUGH MEMORY?
027602  001002                  BNE     10$                     ;IF SET STAY
027604  000137  030164          JMP     NOEXTF                  ;LEAVE IF NOT SET

                        ;Write to TSSR to soft initialize

027610  004737  015774  10$:    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
027614  103405                  BCS     11$                     ;BR IF SOFT INIT = OK
027616                          NEXT.ERRNO
027616  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
027620                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
027620  104455                                                          TRAP    C$ERDF
027622  000465                                                          .WORD   309
```

```
027624  003652                                                          .WORD   SFIERR
027626  012034                                                          .WORD   SFIMSG
828
829                             ;Do a WRITE CHARACTERISTIC command so to invert switch
830
831 027630                      11$:    CKLOOP                          ;LOOP IF SELECTED
    027630  104406                                                      TRAP    C$CLP1
832 027632  012704  030240              MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
833 027636  004737  031620              JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
834 027642  005037  003134              CLR     KTENABLE                ;TURN OFF KT-11
835 027646  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
836 027652  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
837 027656                              FORCERROR       15$
838 027672  103405                      BCS     17$                     ;BR IF SSR SET IN CHKTSSR
839 027674  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
840 027676                              NEXT.ERRNO
841 027676                      15$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
    027676  104455                                                      TRAP    C$ERDF
    027700  000466                                                      .WORD   310
    027702  030552                                                      .WORD   T12WRTSSR
    027704  012046                                                      .WORD   PKTSSR
842 027706                      17$:    CKLOOP                          ;LOOP IF SELECTED
    027706  104406                                                      TRAP    C$CLP1
843 027710  004737  021206              JSR     PC,INVERT               ;INVERT THE SWITCH
844
845                             ;Get an invalid test address
846
847 027714  005037  002222      20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
848 027720                      25$:
849 027720  013737  003142  030302      MOV     NXMHI,T12HIADD          ;SAVE TEST ADDRESS HIGH
850 027726  013737  003140  030304      MOV     NXMLO,T12LOADD          ;SAVE TEST ADDRESS LOW
851 027734                      T124LOOP:
852
853                             ;Set the test packet characteristics data pointer to the
854                             ;  test address.
855
856 027734  012704  030240      30$:    MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
857 027740  004737  031620              JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
858 027744  013764  030304  000002      MOV     T12LOADD,PKLOW(R4)      ;STORE CHAR. DATA PTR LOW ADDRESS
859 027752  013764  030302  000004      MOV     T12HIADD,PKHI(R4)       ;STORE CHAR. DATA PTR HIGH ADDRESS
860
861                             ;Do a WRITE CHARACTERISTIC command
862 027760  004737  017274              JSR     PC,KTOFF                ;TURN OFF KT-11
863 027764  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
864 027770  004737  016250              JSR     PC,WAITF                ;WAIT FOR SSR TO SET
865 027774                              FORCERROR       32$
866 030010  103407                      BCS     40$                     ;BR IF SSR SET IN CHKTSSR
867 030012  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
868 030014                              NEXT.ERRNO
869 030014                      32$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
    030014  104455                                                      TRAP    C$ERDF
    030016  000467                                                      .WORD   311
    030020  030552                                                      .WORD   T12WRTSSR
    030022  012046                                                      .WORD   PKTSSR
870 030024  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
871 030030                      40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
    030030  104406                                                      TRAP    C$CLP1
```

```
                                        FORCERROR       45$,NOTSSR
                                        ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
030042  104410                                                                  TRAP    C$ESCAPE
030044  000124                                                                  .WORD   L10046-.
                                ;If TSSR register NXM bit not set then print error message
                                45$:
                                        MOV     TSSR(R5),R1             ;GET TSSR CONTENTS
                                        FORCERROR       52$
                                        BIT     #NXM,R1                 ;NXM SET?
                                        BNE     60$                     ;BR IF YES
                                        NEXT.ERRNO
                                52$:    MOV     T12LOADD,ERRLO          ;MEMORY TEST ADDRESS LOW
                                        MOV     T12HIADD,ERRHI          ;MEMORY TEST ADDRESS HIGH
                                        ERRHRD  ERRNO,T12NXM,ADDSSR     ;REPORT ADDRESS AND TSSR ERROR
030110  104456                                                                 TRAP    C$ERHRD
030112  000470                                                                 .WORD   312
030114  031207                                                                 .WORD   T12NXM
030116  012126                                                                 .WORD   ADDSSR
                                60$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
030120  104406                                                                 TRAP    C$CLP1
                                        FORCEXIT        90$
                                        TST     T23A                    ;IS IT A 11/23A?
                                        BNE     90$                     ;YES WERE DONE
                                        MOV     T12HIADD,R0             ;GET CURRENT HIGH ADDRESS
                                65$:    INC     R0                      ;GET NEXT ADDRESS
                                        CMP     R0,#77                  ;DONE A21-A16?
                                        BHI     90$                     ;BR IF YES
                                75$:    MOV     R0,T12HIADD             ;SETUP NEW HIGH ORDER ADDRESS
                                        JMP     T124LOOP                ;DO ANOTHER NON-EXISTENT ADDRESS
                                90$:
                                NOEXTF:
                                        JSR     PC,KTOFF                ;TURN OFF KT11
                                        ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\
                                                                                L10046:
030170  104403                                                                   TRAP    C$ESUB
                                        TST     FATFLG                  ;ANY FATAL ERRORS ?
                                        BEQ     100$                    ;BRANCH IF NOT
                                        JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
                                100$:   JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS?
                                        BCC     105$                    ;BR IF NO
                                        JMP     T12LOOP                 ;LOOP UNTIL ITERATION COUNT DONE
                                105$:
                                        JSR     PC,KTOFF                ;TURN OFF MEMORY MANAGEMENT
                                        CLR     T3BFLG                  ;CLEAR TEST FLAG
                                        EXIT    TST                     ;ALL DONE THIS TEST
030226  104432                                                                 TRAP    C$EXIT
030230  001542                                                                 .WORD   L10042-.


                                ;+
                                ;LOCAL STORAGE FOR THIS TEST
                                ;-

                                        .=<.+10>&177770
                                T12PACKET:
                                        .WORD   100004                  ;COMMAND PACKET FOR TEST
                                                                        ;WRITE CHARACTERISTICS COMMAND, WITH ACK
```

```
                                                .WORD   T12DATA         ;ADDRESS OF CHARACTERISTICS BLOCK
                                                .WORD   0
                                                .WORD   8.              ;STARTING VALUE OF BLOCK SIZE

                                        T12DATA:                        ;CHARACTERISTICS DATA BLOCK
                                                .WORD   T12BFR          ;LOW ADDRESS OF MESSAGE BUFFER
                                                .WORD   0               ;HIGH ORDER OF MESSAGE BUFFER
                                                .WORD   14.             ;LENGTH OF MESSAGE BUFFER
                                                .WORD   0,0

                                        T12BFR: .BLKW   8.              ;MESSAGE BUFFER

                                        T12HIADD:       .WORD   0       ;HIGH ADDRESS
                                        T12LOADD:       .WORD   0       ;LOW ADDRESS
                                        T12PAR6:        .WORD   0       ;ADDRESS IN PAR FORMAT
                                        T12KT:          .WORD   0       ;TEST ABOVE 28K SWITCH
                                        T124TST:        .WORD   0       ;ADDRESS TEST BIT
                                        ;+
                                        ;
                                        ;TABLE OF ADDRESSES
                                        ;
                                        ;-
                                        T12BLK: .WORD   000001
                                                .WORD   000002
                                                .WORD   000003
                                                .WORD   000005
                                                .WORD   000006
                                                .WORD   000007
                                                .WORD   000011
                                                .WORD   000012
                                                .WORD   000013
                                                .WORD   000021
                                                .WORD   000022
                                                .WORD   000023
                                                .WORD   000041
                                                .WORD   000042
                                                .WORD   000043
                                                .WORD   000101
                                                .WORD   000102
                                                .WORD   000103
                                                .WORD   000201
                                                .WORD   000202
                                                .WORD   000203
                                                .WORD   000401
                                                .WORD   000402
                                                .WORD   000403
                                                .WORD   001001
                                                .WORD   001002
                                                .WORD   001003
                                                .WORD   002001
                                                .WORD   002002
                                                .WORD   002003
                                                .WORD   004001
                                                .WORD   004002
                                                .WORD   004003
                                                .WORD   010001
                                                .WORD   010002
```

```
                                        .WORD   010003
                                        .WORD   020001
                                        .WORD   020002
                                        .WORD   020003
                                        .WORD   040001
                                        .WORD   040002
                                        .WORD   040003
                                        .WORD   100001
                                        .WORD   100002
                                        .WORD   100003
                                T12TBE: .WORD           177777
                                ;+
                                ;LOCAL TEXT MESSAGES FOR TEST
                                ;-

                                TST12ID:        .ASCIZ  'DMA Memory Addressing'
                                T12GETSSR:      .ASCIZ  'Contents of TSSR Incorrect After GET STATUS'
                                T12WRTSSR:      .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
                                T12MSGBUF:      .ASCIZ  'Message Buffer Contents Incorrect After WRITE CHARACTERISTICS'
                                T12BKGND:       .ASCIZ  'Background Pattern Disturbed By WRITE CHARACTERISTICS'
                                T12NINT:        .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
                                T12DPR:         .ASCIZ  'Write Characteristic data in ram does not match expected'
                                T12NXM:         .ASCIZ  'TSSR NXM bit failed to set when non-existent memory address specifi
                                        .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
                                ;-
031316                          T12CONVERT:
031316                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
031322  005037  030304                   CLR     T12LOADD                ;CLEAR LOW ADDRESS
031326  005037  030302                   CLR     T12HIADD                ;CLEAR HIGH ADDRESS
031332  005037  030306                   CLR     T12PAR6                 ;CLEAR PAR6 BIASED ADDRESS
031336  042701  170000                   BIC     #^C<7777>,R1            ;FORCE TO LOWER 12 BITS OF ADDRESS
031342  010005                          MOV     R0,R5                   ;SAVE HIGH ORDER ADDRESS BITS
031344  004737  017274                   JSR     PC,KTOFF                ;SHUTOFF MEMORY MANAGEMENT
031350  013702  003124                   MOV     FREE,R2                 ;GET FIRST FREE ADDRESS
031354  062702  000020                   ADD     #16.,R2                 ;IN CASE TEST PATTERN=0
031360  060102                          ADD     R1,R2                   ;ADD IN TEST PATTERN
031362  042702  000003                   BIC     #3,R2                   ;MAKE IT MODULO-4
031366  013703  003130           25$:    MOV     FREEHI,R3               ;GET LAST FREE ADDRESS
031372  162703  000020                   SUB     #16.,R3                 ;SAVE AT LEAST 8 WORDS (IN CASE MESSAGE BUFFER)
031376  010237  030304                   MOV     R2,T12LOADD             ;SAVE POSSIBLE LOW ADDRESS
031402  010237  030306                   MOV     R2,T12PAR6              ;SAVE IT IN PAR6 BIASED TOO
031406  020203                          CMP     R2,R3                   ;IS THIS ADDRESS ABOVE FREE SPACE?
031410  101007                          BHI     35$                     ;BR IF YES
031412  020237  003124                   CMP     R2,FREE                 ;IS IT IN FREE SPACE?
031416  103007                          BHIS    50$                     ;BR IF YES- ITS GOOD
031420  005737  003134                   TST     KTENABLE                ;TESTING ABOVE 28K?
031424  001004                          BNE     50$                     ;BR IF YES
031426  000424                          BR      90$                     ;BR IF NOT IN FREE SPACE
031430  162702  000020           35$:    SUB     #16.,R2                 ;FORCE FIT THE TEST PATTERN
031434  000754                          BR      25$                     ;TRY THIS TEST PATTERN ADDRESS
031436                          50$:
031436  005737  003134                   TST     KTENABLE                ;TESTING ABOVE 28K?
031442  001420                          BEQ     100$                    ;BR IF NO
031444  005737  003132                   TST     KTFLG                   ;ANY MEMORY ABOVE 28K?
031450  001413                          BEQ     90$                     ;BR IF NO
031452  004737  017256                   JSR     PC,KTON                 ;TURN ON MEMORY MANAGEMENT
031456  010500                          MOV     R5,R0                   ;GET HIGH ORDER ADDRESS
031460  010037  030302                   MOV     R0,T12HIADD             ;SAVE POSSIBLE HIGH ADDRESS
031464  010201                          MOV     R2,R1                   ;GET COMPUTED LOW ORDER ADDRESS
031466  004737  017316                   JSR     PC,SETMAP               ;RETURN PAR6 BIASED ADDRESS IN R0
031472  010037  030306                   MOV     R0,T12PAR6              ;COPY PAR6 BIASED ADDRESS
031476  103403                          BCS     105$                    ;BR IF VALID ADDRESS
```

```
                                90$:    CLC                     ;CLR C BIT FOR FAILURE
                                        BR      105$            ;
                                100$:   SEC                     ;SET SUCCESS
                                105$:   RTS     PC              ;RETURN


                                ;+
                                ;
                                ;ROUTINE TO READ THE FIRST 2 BYTES FROM RAM
                                ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
                                ;
                                ;INPUT:
                                ;
                                ;       R4      ADDRESS OF THE COMMAND PACKET
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;
                                ;OUTPUT:
                                ;
                                ;       CARRY   SET - RAM MATCHES PACKET
                                ;               CLR - RAM DOES NOT MATCH PACKET
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       THE TABLE RAMDATA IS FILLED WITH THE
                                ;       DATA HELD IN RAM.
                                ;       RAMSIZ  SET TO 2 FOR PRAMPKT ROUTINE
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
                                ;
                                ;-
                                T12CKRAM::
                                        SAVREG                  ;SAVE THE GENERAL REGISTERS
                                        MOV     #RAMDATA,R1     ;ADDRESS TO SAVE THE RAM DATA
                                        MOV     #RMPKTBEG,R2    ;BYTE ADDRESS OF FIRST RAM DATA
                                        CLR     R3              ;CLEAR THE ERROR FLAG
                                        JSR     PC,CHKTSSR      ;WAIT FOR SSR
                                        MOVB    #0,TSDB(R5)     ;SET MAINTENANCE MODE
                                10$:    JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
                                        MOV     R2,TSDB(R5)     ;SELECT NEXT RAM ADDRESS
                                        JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
                                        MOVB    TSBA(R5),(R1)   ;READ THE RAM DATA
                                        CMPB    (R1)+,(R4)+     ;COMPARE TO EXPECTED
                                        BEQ     20$             ;BRANCH IF OK
                                        INC     R3              ;SET ERROR FLAG
                                20$:    INC     R2              ;ADDRESS OF NEXT RAM LOCATION
                                        CMP     R2,#RMPKTBEG+2  ;DONE 2 BYTES?
                                        BLT     10$             ;BR IF NO
                                        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     #2,RAMSIZ       ;SETUP RAMSIZ
                                        RTS     PC              ;RETURN
```

```
                                        ;+
                                        ;
                                        ;ROUTINE TO SETUP PACKET TO WRITE CHARACTERISTICS
                                        ;-

031620                                  T12SWRT:
031620                                          SAVREG                          ;SAVE THE REGISTERS
031624  012701  030240                          MOV     #T12PACKET,R1           ;START OF THE PACKET
031630  012721  100004                          MOV     #100004,(R1)+           ;WRITE CHARACTERISTICS WITH ACK
031634  012721  030250                          MOV     #T12DATA,(R1)+          ;ADDRESS OF CHAR DATA BLOCK
031640  005021                                  CLR     (R1)+                   ;EXTENDED ADDRESS
031642  012721  000010                          MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
031646  012721  030262                          MOV     #T12BFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
031652  005021                                  CLR     (R1)+
031654  012721  000016                          MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
031660  005021                                  CLR     (R1)+
031662  005011                                  CLR     (R1)
031664  000207                                  RTS     PC                      ;RETURN
                                        ;+
                                        ;
                                        ;ROUTINE TO SETUP A GET STATUS COMMAND PACKET AT CURRENT PACKET ADDRESS
                                        ;
                                        ;       R3      HIGH ORDER PACKET ADDRESS
                                        ;       R4      LOW ORDER PACKET ADDRESS
                                        ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG CLEAR
                                        ;
                                        ;-

031666                                  T12SETGET:
031666                                          SAVREG                          ;SAVE THE REGISTERS
031672  010401                                  MOV     R4,R1                   ;GET LOW ORDER ADDRESS
031674  005737  003134                          TST     KTENABLE                ;TESTING ABOVE 28K?
031700  001404                                  BEQ     10$                     ;BR IF NO
031702  010300                                  MOV     R3,R0                   ;GET HIGH ORDER ADDRESS
031704  004737  017316                          JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
031710  010001                                  MOV     R0,R1                   ;GET ADDRESS
031712  012700  000017          10$:            MOV     #P.GETSTATUS,R0         ;GET STATUS COMMAND CODE NO IE
031716  052700  100000                          BIS     #P.ACK,R0               ;SET ACK
031722  010021                                  MOV     R0,(R1)+                ;STORE GET STATUS IN PACKET
031724  005021                                  CLR     (R1)+                   ;CLEAR UNUSED WORD
031726  000207                                  RTS     PC                      ;RETURN


                                        ;+
                                        ;
                                        ;ROUTINE TO SETUP A CHARACTERISTIC DATA BLOCK AT A TEST ADDRESS
                                        ;
                                        ;-

031730                                  T12CHAR:
031730                                          SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
031734  012700  030250                          MOV     #T12DATA,R0             ;GET T12PACKET DATA POINTER
031740  013701  030304                          MOV     T12LOADD,R1             ;ASSUME NOT ABOVE 28K
031744  005737  003134                          TST     KTENABLE                ;TESTING ABOVE 28K?
031750  001402                                  BEQ     10$                     ;BR IF NO
031752  013701  030306                          MOV     T12PAR6,R1              ;SET TEST ADDRESS ABOVE 28K
031756  012021                  10$:            MOV     (R0)+,(R1)+             ;STORE DATA WORD 1
031760  012021                                  MOV     (R0)+,(R1)+             ;STORE DATA WORD 2
```

```
1174 031762  012021                    MOV     (R0)+,(R1)+     ;STORE DATA WORD 3
1175 031764  012021                    MOV     (R0)+,(R1)+     ;STORE DATA WORD 4
1176 031766  012021                    MOV     (R0)+,(R1)+     ;STORE DATA WORD 5
1177 031770  000207                    RTS     PC              ;RETURN
1178
1179 031772                            ENDTST
     031772                                                  L10042:
     031772  104401                                                TRAP    C$ETST
```

```
1181
```

```
                                        .SBTTL  TEST  4: RAM EXERCISER TEST
                                ;+
                                ;
                                ;THIS TEST USES THE READ AND WRITE RAM (BOTH SINGLE AND 256
                                ;LOCATIONS) SELECT CODES OF THE WRITE SUBSYSTEM MEMORY COMMAND
                                ;TO EXERCISE THE CONTROLLER'S RAM MEMORY AND DMA LOGIC
                                ;
                                ;-
031774                                  BGNTST
031774                                                                  T4::

1198 031774  005737  002214                TST     TSTCNT                  ;CHECK FOR RUN MODE
1199 032000  001402                        BEQ     10$                     ;BR, IF NOT ONLY PROGRAM RUN
1200 032002  005237  003400                INC     SKIPT                   ;SET SKIP SW
1201 032006  012700  034433        10$:    MOV     #TST15ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
1202 032012  004737  016510                JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
1203 032016  012737  000005  002216        MOV     #5,LOOPCNT              ;PERFORM 5 ITERATIONS
1204 032024                             T15LOOP:
                                ;
                                ;TEST 4, SUBTEST 1
                                ;
                                ;       THIS SUBTEST WRITES THE ADDRESS (8 BITS) INTO THE
                                ;       RAM MEMORY SINGLE WORD (8 BITS) MODE
                                ;
                                ;-
032024                                  BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
                                                                                T4.1:
032024  104402                                                                  TRAP    C$BSUB
                                        SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
032026  012700  000000                                                          MOV     #PRI00,R0
032032  104441                                                                  TRAP    C$SPRI
1215 032034  005737  003400                TST     SKIPT                   ;SHOULD WE SKIP THIS SUBTEST
1216 032040  001402                        BEQ     10$                     ;BR, IF NOW SKIP REQUIRED
1217 032042  000137  032324                JMP     50$                     ;SKIP SUBTEST
1218 032046  004737  034452        10$:    JSR     PC,T15REST              ;SET COMMAND PACKET
1219 032052  004737  034524                JSR     PC,T15RT2               ;SET UP OTHER COMMAND PACKET
1220 032056  004737  015774                JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
1221 032062  103405                        BCS     20$                     ;BR IF INIT WAS OK
1225 032064  010001                        MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
1226 032066                                ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
032066  104455                                                                  TRAP    C$ERDF
032070  000621                                                                  .WORD   401
032072  003652                                                                  .WORD   SFIERR
032074  012034                                                                  .WORD   SFIMSG
1227 032076                             20$:
1228 032076  012704  033350                MOV     #T15PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
1229 032102  004737  010662                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
1230 032106  103405                        BCS     23$                     ;BR, IF COMMAND ISSUED OK
1234 032110  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1235 032112                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
032112  104456                                                                  TRAP    C$ERHRD
032114  000622                                                                  .WORD   402
032116  005056                                                                  .WORD   WRTMSG
032120  012034                                                                  .WORD   SFIMSG
1236 032122  012703  000400        23$:    MOV     #256.,R3                ;STARTING ADDRESS FOR RAM WRITE
```

```
1237 032126  112737  000001  034061          MOVB    #1,T15BS1               ;SIZE OF TRANSFER
1238 032134  112737  000002  034060          MOVB    #2,T15BS0               ;WRITE RAM "COMMAND"
1239 032142                          25$:
1240 032142  010337  034062                   MOV     R3,T15S2                ;ADDRESS FOR RAM
1241 032146  012704  034050                   MOV     #T15PK2,R4              ;WRITE SUBSYS MEM PACKET
1242 032152  110337  034064                   MOVB    R3,T15S3                ;DATA FOR WRITE (ADDRESS)
1243 032156  010465  000000                   MOV     R4,TSDB(R5)             ;ISSUE COMMAND
1244 032162  004737  016336                   JSR     PC,CHKTSSR              ;WAIT FOR SSR
1245 032166  103407                           BCS     30$                     ;BR, IF NO ERROR
1246 032170  010001                           MOV     R0,R1                   ;ERROR, SAVE TSSR
1250 032172                                   ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT AFTER WRITE SUB MEM
     032172  104456                                                           TRAP    C$ERHRD
     032174  000623                                                           .WORD   403
     032176  034066                                                           .WORD   T15SSR
     032200  012046                                                           .WORD   PKTSSR
1251 032202                                   ESCAPE  SUB                     ;DON'T CONTINUE IF ERROR ON WRITE
     032202  104410                                                           TRAP    C$ESCAPE
     032204  000122                                                           .WORD   L10050-.
1252 032206                          30$:     CKLOOP                          ;SCOPE LOOP
     032206  104406                                                           TRAP    C$CLP1
1253
1254
1255 032210  005203                           INC     R3                      ;NEXT ADDRESS
1256 032212  020327  010000                   CMP     R3,#10000               ;END OF RAM MEMORY CHECK
1257 032216  001351                           BNE     25$                     ;LOOP TILL ALL RAM WRITTEN
1258 032220  005002                           CLR     R2                      ;CLEAR OUT R2 HIGH BITS
1259 032222  005303                           DEC     R3                      ;SET BACK TO 7777
1260 032224  110337  034064          40$:     MOVB    R3,T15S3                ;GET DATA PATTERN BACK IN SHAPE
1261 032230  010337  034062                   MOV     R3,T15S2                ;ADDRESS FOR RAM READ
1262 032234  112737  000001  034060          MOVB    #1,T15BS0               ;READ RAM COMMAND
1263 032242  010465  000000                   MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS TO CONTR.
1264 032246  004737  016336                   JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1265 032252  103405                           BCS     43$                     ;BR, IF NO PROBLEM
1266 032254  010001                           MOV     R0,R1                   ;SAVE TSSR
1270 032256                                   ERRDF   ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032256  104455                                                           TRAP    C$ERDF
     032260  000624                                                           .WORD   404
     032262  034066                                                           .WORD   T15SSR
     032264  012046                                                           .WORD   PKTSSR
1271 032266                          43$:     CKLOOP                          ;SCOPE LOOP
     032266  104406                                                           TRAP    C$CLP1
1272 032270  013701  033412                   MOV     T15BFR+20,R1            ;GET RAM READ DATA
1273 032274  010302                           MOV     R3,R2                   ;SET UP FOR COMPARE
1274 032276  120102                           CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
1275 032300  001404                           BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
1279 032302                                   ERRHRD  ERRNO,T15AM4,EXPBREC    ;WRITTEN DATA NOT = TO READ
     032302  104456                                                           TRAP    C$ERHRD
     032304  000625                                                           .WORD   405
     032306  034345                                                           .WORD   T15AM4
     032310  015502                                                           .WORD   EXPBREC
1280 032312                          45$:     CKLOOP                          ;SCOPE LOOP
     032312  104406                                                           TRAP    C$CLP1
1281 032314  005303                           DEC     R3                      ;DROP DATA COUNTER (PATTERN)
1282 032316  020327  000377                   CMP     R3,#255.                ;AT BOTTOM YET
1283 032322  001340                           BNE     40$                     ;BR, IF MORE TO CHECK
1284 032324                          50$:     CKLOOP                          ;SCOPE LOOP
     032324  104406                                                           TRAP    C$CLP1
```

```
1285 032326                            ENDSUB                    ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     032326                                                                  L10050:
     032326  104403                                                                TRAP    C$ESUB
1286
```

```
 1288
 1289 032330                            BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
      032330                                                                          T4.2:
      032330  104402                                                                    TRAP    C$BSUB
 1290
 1291                                   ;+
 1292                                   ;
 1293                                   ;TEST 4, SUBTEST 2
 1294                                   ;
 1295                                   ;
 1296                                   ;       THIS SUBTEST WRITES RAM WITH ALL ZEROS
 1297                                   ;       THEN WALKS AN ALL ONES WORD DOWN THROUGH MEMORY
 1298                                   ;
 1299 032332  004737  034452            JSR     PC,T15REST              ;RESTORE PACKET FOR WRITE CHARA
 1300 032336  004737  034524            JSR     PC,T15RT2               ;RESTORE PACKET FOR WRT SUB SYS MEM
 1301 032342  004737  015774            JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
 1302 032346  103405                    BCS     20$                     ;BR IF INIT WAS OK
 1306 032350  010001                    MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
 1307 032352                            ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
      032352  104455                                                                  TRAP    C$ERDF
      032354  000626                                                                  .WORD   406
      032356  003652                                                                  .WORD   SFIERR
      032360  012034                                                                  .WORD   SFIMSG
 1308 032362                            20$:
 1309 032362  012704  033350            MOV     #T15PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
 1310 032366  004737  010662            JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 1311 032372  103405                    BCS     25$                     ;BR, IF COMMAND ISSUED OK
 1315 032374  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 1316 032376                            ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      032376  104456                                                                  TRAP    C$ERHRD
      032400  000627                                                                  .WORD   407
      032402  005056                                                                  .WORD   WRTMSG
      032404  012034                                                                  .WORD   SFIMSG
 1317 032406                            25$:
 1318 032406  112737  000001  034061    MOVB    #1,T15BS1               ;SET SIZE OF TRANSFER 1 BYTE
 1319 032414  012704  034050            MOV     #T15PK2,R4              ;SET NEW PACKET ADDRESS
 1320 032420  012703  000400            MOV     #256.,R3                ;STARTING ADDRESS IN RAM
 1321 032424  112737  000002  034060    MOVB    #2,T15BS0               ;WRITE RAM COMMAND
 1322 032432  105037  034064            CLRB    T15S3                   ;SET DATA TO 000
 1323 032436  010337  034062    30$:    MOV     R3,T15S2                ;ADDRESS TO PACKET DATA AREA
 1324 032442  010465  000000            MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
 1325 032446  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR
 1326 032452  103405                    BCS     33$                     ;BR, IF NO PROBLEM
 1327 032454  010001                    MOV     R0,R1                   ;SAVE TSSR
 1331 032456                            ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
      032456  104456                                                                  TRAP    C$ERHRD
      032460  000630                                                                  .WORD   408
      032462  034066                                                                  .WORD   T15SSR
      032464  012046                                                                  .WORD   PKTSSR
 1332 032466                    33$:    CKLOOP                          ;SCOPE LOOP
      032466  104406                                                                  TRAP    C$CLP1
 1333
 1334
 1335 032470  005203                    INC     R3                      ;NEXT ADDRESS
 1336 032472  020327  010000            CMP     R3,#10000               ;END OF RAM MEMORY CHECK
 1337 032476  001357                    BNE     30$                     ;BR, MORE RAM TO GO
 1338 032500  005303            35$:    DEC     R3                      ;SET BACK TO 7777
```

```
1339 032502  005002                    40$:  CLR     R2                      ;SET TO ALL ZEROS
1340 032504  112737  000001  034060          MOVB    #1,T15BS0               ;READ RAM COMMAND
1341 032512  010337  034062                  MOV     R3,T15S2                ;ADDRESS TO BE READ TO PACKET DATA
1342 032516  010465  000000                  MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
1343 032522  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1344 032526  103405                          BCS     41$                     ;BR, IF ALL IS WELL
1345 032530  010001                          MOV     R0,R1                   ;SAVE TSSR
1349 032532                                  ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032532  104456                                                          TRAP    C$ERHRD
     032534  000631                                                          .WORD   409
     032536  034066                                                          .WORD   T15SSR
     032540  012046                                                          .WORD   PKTSSR
1350 032542                            41$:  CKLOOP                          ;SCOPE LOOP
     032542  104406                                                          TRAP    C$CLP1
1351 032544  013701  033412                  MOV     T15BFR+20,R1            ;PICK UP READ DATA
1352 032550  120102                          CMPB    R1,R2                   ;BOTH SHOULD BE 00000000 BINARY
1353 032552  001404                          BEQ     42$                     ;BR, IF DATA IS GOOD
1357 032554                                  ERRHRD  ERRNO,T15AM3,EXPBREC    ;CHARACTERISTICS DATA NOT CORRECT
     032554  104456                                                          TRAP    C$ERHRD
     032556  000632                                                          .WORD   410
     032560  034243                                                          .WORD   T15AM3
     032562  015502                                                          .WORD   EXPBREC
1358 032564                            42$:  CKLOOP                          ;SCOPE LOOPER
     032564  104406                                                          TRAP    C$CLP1
1359 032566  012702  000377                  MOV     #000377,R2              ;SET ALL ONES WORD
1360 032572  112737  000002  034060          MOVB    #2,T15BS0               ;WRITE RAM COMMAND
1361 032600  112737  000377  034064          MOVB    #000377,T15S3           ;ALL ONES PATTERN
1362 032606  010465  000000                  MOV     R4,TSDB(R5)             ;PASS PACKET ADDRESS TO CONTR.
1363 032612  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
1364 032616  103405                          BCS     43$                     ;BR, IF OK (NO ERROR)
1365 032620  010001                          MOV     R0,R1                   ;SAVE TSSR
1369 032622                                  ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032622  104456                                                          TRAP    C$ERHRD
     032624  000633                                                          .WORD   411
     032626  034066                                                          .WORD   T15SSR
     032630  012046                                                          .WORD   PKTSSR
1370 032632                            43$:  CKLOOP                          ;SCOPE LOOP
     032632  104406                                                          TRAP    C$CLP1
1371 032634  112737  000001  034060          MOVB    #1,T15BS0               ;SET UP FOR RAM READ
1372 032642  010465  000000                  MOV     R4,TSDB(R5)             ;ISSUE RAM READ
1373 032646  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1374 032652  103405                          BCS     44$                     ;BR, IF OK (NO ERROR)
1375 032654  010001                          MOV     R0,R1                   ;SAVE TSSR
1379 032656                                  ERRDF   ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032656  104455                                                          TRAP    C$ERDF
     032660  000634                                                          .WORD   412
     032662  034066                                                          .WORD   T15SSR
     032664  012046                                                          .WORD   PKTSSR
1380 032666  013701  033412            44$:  MOV     T15BFR+20,R1            ;PICK UP REC'D DATA
1381 032672  120102                          CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
1382 032674  001404                          BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
1386 032676                                  ERRHRD  ERRNO,T15AM2,EXPBREC    ;WRITTEN DATA NOT = TO READ
     032676  104456                                                          TRAP    C$ERHRD
     032700  000635                                                          .WORD   413
     032702  034142                                                          .WORD   T15AM2
     032704  015502                                                          .WORD   EXPBREC
1387 032706                            45$:  CKLOOP                          ;SCOPE LOOP
```

```
        032706  104406                                                  TRAP    C$CLP1
1388    032710  005303                  DEC     R3                      ;DROP RAM ADDRESS POINTER
1389    032712  020327  000377          CMP     R3,#255.                ;AT START YET
1390    032716  001271                  BNE     40$                     ;BR, IF MORE RAM TO CHECK
1391
1392    032720                          ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
        032720                                                          L10051:
        032720  104403                                                  TRAP    C$ESUB
1393
```

```
1395 032722                             BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     032722                                                                                  T4.3:
     032722  104402                                                                             TRAP    C$BSUB
1396                                    ;+
1397                                    ;
1398                                    ;TEST 4, SUBTEST 3
1399                                    ;
1400                                    ;
1401                                    ;       THIS SUBTEST WRITES RAM WITH ALL ONES
1402                                    ;       THEN WALKS AN ALL ZEROS WORD DOWN THROUGH MEMORY
1403                                    ;
1404 032724  005737  003400             10$:    TST     SKIPT                   ;CHECK RUN MODE
1405 032730  001402                             BEQ     10$                     ;BR, IF NO SKIP
1406 032732  000137  033326                     JMP     50$                     ;SKIP SUBTEST
1407 032736  004737  034452                     JSR     PC,T15REST              ;RESTORE PACKET FOR WRITE CHARA
1408 032742  004737  034524                     JSR     PC,T15RT2               ;RESTORE PACKET FOR WRT SUB SYS MEM
1409 032746  004737  015774                     JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
1410 032752  103405                           BCS     20$                     ;BR IF INIT WAS OK
1414 032754  010001                           MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
1415 032756                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
     032756  104455                                                                          TRAP    C$ERDF
     032760  000636                                                                          .WORD   414
     032762  003652                                                                          .WORD   SFIERR
     032764  012034                                                                          .WORD   SFIMSG
1416 032766                             20$:
1417 032766  012704  033350                     MOV     #T15PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
1418 032772  004737  010662                     JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
1419 032776  103405                           BCS     25$                     ;BR, IF COMMAND ISSUED OK
1423 033000  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1424 033002                                    ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     033002  104456                                                                          TRAP    C$ERHRD
     033004  000637                                                                          .WORD   415
     033006  005056                                                                          .WORD   WRTMSG
     033010  012034                                                                          .WORD   SFIMSG
1425 033012                             25$:
1426 033012  112737  000001  034061             MOVB    #1,T15BS1               ;SET SIZE TO 1 BYTE
1427 033020  012704  034050                     MOV     #T15PK2,R4              ;SET NEW PACKET ADDRESS
1428 033024  012703  000400                     MOV     #256.,R3                ;STARTING ADDRESS IN RAM
1429 033030  112737  000002  034060             MOVB    #2,T15BS0               ;WRITE RAM COMMAND
1430 033036  112737  000377  034064             MOVB    #377,T15S3              ;SET DATA TO 377
1431 033044  010337  034062             30$:    MOV     R3,T15S2                ;ADDRESS TO PACKET DATA AREA
1432 033050  010465  000000                     MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
1433 033054  004737  016336                     JSR     PC,CHKTSSR              ;WAIT FOR SSR
1434 033060  103405                           BCS     33$                     ;BR, IF NO PROBLEM
1435 033062  010001                           MOV     R0,R1                   ;SAVE TSSR
1439 033064                                    ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     033064  104456                                                                          TRAP    C$ERHRD
     033066  000640                                                                          .WORD   416
     033070  034066                                                                          .WORD   T15SSR
     033072  012046                                                                          .WORD   PKTSSR
1440 033074                             33$:    CKLOOP                          ;SCOPE LOOP
     033074  104406                                                                          TRAP    C$CLP1
1441
1442
1443 033076  005203                             INC     R3                      ;NEXT ADDRESS
1444 033100  020327  010000                     CMP     R3,#10000               ;END OF RAM MEMORY CHECK
1445 033104  001357                             BNE     30$                     ;BR, MORE RAM TO GO
```

```
1446 033106  005303                    35$:   DEC     R3                      ;SET BACK TO 7777
1447 033110  112702  000377            40$:   MOVB    #377,R2                 ;SET TO ALL ONES
1448 033114  112737  000001  034060           MOVB    #1,T15BS0               ;READ RAM COMMAND
1449 033122  010337  034062                    MOV     R3,T15S2                ;ADDRESS TO BE READ TO PACKET DATA
1450 033126  010465  000000                    MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
1451 033132  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1452 033136  103405                            BCS     41$                     ;BR, IF ALL IS WELL
1453 033140  010001                            MOV     R0,R1                   ;SAVE TSSR
1457 033142                                    ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     033142  104456                                                            TRAP    C$ERHRD
     033144  000641                                                            .WORD   417
     033146  034066                                                            .WORD   T15SSR
     033150  012046                                                            .WORD   PKTSSR
1458 033152                            41$:    CKLOOP                          ;SCOPE LOOP
     033152  104406                                                            TRAP    C$CLP1
1459 033154  013701  033412                    MOV     T15BFR+20,R1            ;PICK UP READ DATA
1460 033160  120102                            CMPB    R1,R2                   ;BOTH SHOULD BE 11111111 BINARY
1461 033162  001404                            BEQ     42$                     ;BR, IF DATA IS GOOD
1465 033164                                    ERRHRD  ERRNO,T15AM3,EXPBREC    ;CHARACTERISTICS DATA NOT CORRECT
     033164  104456                                                            TRAP    C$ERHRD
     033166  000642                                                            .WORD   418
     033170  034243                                                            .WORD   T15AM3
     033172  015502                                                            .WORD   EXPBREC
1466 033174  012702  000377            42$:   MOV     #000377,R2              ;SET ALL ONES WORD
1467 033200  012737  000002  034060           MOV     #2,T15BS0               ;WRITE RAM COMMAND
1468 033206  112737  000377  034064           MOVB    #000377,T15S3           ;ALL ONES PATTERN
1469 033214  010465  000000                    MOV     R4,TSDB(R5)             ;PASS PACKET ADDRESS TO CONTR.
1470 033220  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR
1471 033224  103405                            BCS     43$                     ;BR, IF OK (NO ERROR)
1472 033226  010001                            MOV     R0,R1                   ;SAVE TSSR
1476 033230                                    ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     033230  104456                                                            TRAP    C$ERHRD
     033232  000643                                                            .WORD   419
     033234  034066                                                            .WORD   T15SSR
     033236  012046                                                            .WORD   PKTSSR
1477 033240                            43$:    CKLOOP                          ;SCOPE LOOP
     033240  104406                                                            TRAP    C$CLP1
1478 033242  112737  000001  034060           MOVB    #1,T15BS0               ;SET UP FOR RAM READ
1479 033250  010465  000000                    MOV     R4,TSDB(R5)             ;ISSUE RAM READ
1480 033254  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1481 033260  103405                            BCS     44$                     ;BR, IF OK (NO ERROR)
1482 033262  010001                            MOV     R0,R1                   ;SAVE TSSR
1486 033264                                    ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     033264  104456                                                            TRAP    C$ERHRD
     033266  000644                                                            .WORD   420
     033270  034066                                                            .WORD   T15SSR
     033272  012046                                                            .WORD   PKTSSR
1487 033274  013701  033412            44$:   MOV     T15BFR+20,R1            ;PICK UP REC'D DATA
1488 033300  120102                            CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
1489 033302  001404                            BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
1493 033304                                    ERRHRD  ERRNO,T15AM2,EXPBREC    ;WRITTEN DATA NOT = TO READ
     033304  104456                                                            TRAP    C$ERHRD
     033306  000645                                                            .WORD   421
     033310  034142                                                            .WORD   T15AM2
     033312  015502                                                            .WORD   EXPBREC
1494 033314                            45$:    CKLOOP                          ;SCOPE LOOP
     033314  104406                                                            TRAP    C$CLP1
```

```
1495 033316  005303                    DEC     R3              ;DROP RAM ADDRESS POINTER
1496 033320  020327  000377             CMP     R3,#255.        ;AT START YET
1497 033324  001271                    BNE     40$             ;BR, IF MORE RAM TO CHECK
1498
1499 033326                        50$:
1500 033326                            ENDSUB                  ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     033326                                                                  L10052:
     033326  104403                                                              TRAP    C$ESUB
```

```
1502
1503 033330  004737  016456          JSR     PC,TSTLOOP              ;DO WE NEED TO ITERATE TEST ?
1504 033334  103002                  BCC     63$                     ;BRANCH IF NOT
1505 033336  000137  032024          JMP     T15LOOP                 ;EXECUTE AGAIN
1506 033342                  63$:    EXIT    TST                     ;ALL DONE THIS TEST
     033342  104432                                                          TRAP    C$EXIT
     033344  001216                                                          .WORD   L10047-.
1507                                 ;+
1508                                 ;LOCAL STORAGE FOR THIS TEST
1509                                 ;-
1510
1512         033350                          .=<.+10>&177770
1514 033350                          T15PACKET:                      ;COMMAND PACKET FOR TEST
1515 033350  100204                          .WORD   100204          ;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
1516 033352  033360                          .WORD   T15DATA         ;ADDRESS OF CHARACTERISTICS BLOCK
1517 033354  000000                          .WORD   0
1518 033356  000010                          .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
1519 033360                          T15DATA:                        ;CHARACTERISTICS DATA BLOCK
1520 033360  033372                          .WORD   T15BFR          ;ADDRESS OF MESSAGE BUFFER
1521 033362  000000                          .WORD   0
1522 033364  000400                          .WORD   256.            ;LENGTH OF MESSAGE BUFFER
1523 033366  000000  000000                  .WORD   0,0
1524 033372                          T15BFR: .BLKW   150.            ;MESSAGE BUFFER
1525                                 ;
1526                                 ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
1527                                 ;
1529         034050                          .=<.+10>&177770
1531 034050                          T15PK2:
1532 034050  100206                          .WORD   100206          ;WRITE SUB SYS MEM COMMAND, IE AND ACK
1533 034052  034060                          .WORD   T15BF2          ;ADDRESS OF SELECT BLOCK DATA
1534 034054  000000                          .WORD   0
1535 034056  000006                          .WORD   6.              ;SIZE OF DATA PACKET
1536
1537                                         .EVEN
1538 034060                          T15BF2:
1539 034060     000                  T15BS0: .BYTE   0               ;BSEL0 AREA
1540 034061     000                  T15BS1: .BYTE   0               ;BSEL1 AREA
1541 034062  000000                  T15S2:  .WORD   0               ;SEL 2 AREA
1542 034064  000000                  T15S3:  .WORD   0               ;DATA AREA
1543
1544
1545
```

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

034066     127     122     111  T15SSR: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
034142     127     122     111  T15AM2: .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Ones Word Read Back'
034243     127     122     111  T15AM3: .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Zeros Word Read Back'
034345     127     122     111  T15AM4: .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On Address Test'
034433     122     101     115  TST15ID:        .ASCIZ  'RAM Exerciser'
                                        .EVEN
                                        ;+
                                        ;
                                        ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
                                        ;WRITE SUBSYSTEM MEMORY COMMAND
                                        ;
                                        ;-

034452                                  T15REST:
034452                                          SAVREG                          ;SAVE THE REGISTERS
034456  012701  033350                          MOV     #T15PACKET,R1           ;START OF THE PACKET
034462  012721  100204                          MOV     #100204,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK, IE
034466  012721  033360                          MOV     #T15DATA,(R1)+          ;ADDRESS OF CHARAISTICS DATA BLOCK
034472  005021                                  CLR     (R1)+                   ;EXTENDED ADDRESS
034474  012721  000010                          MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
034500  012721  033372                          MOV     #T15BFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
034504  005021                                  CLR     (R1)+
034506  012721  000400                          MOV     #256.,(R1)+             ;LENGTH OF MESSAGE BUFFER
034512  005021                                  CLR     (R1)+
034514  005011                                  CLR     (R1)
034516  005037  033372                          CLR     T15BFR                  ;CLEAR 1ST LOC IN MESSAGE BUFFER
034522  000207                                  RTS     PC                      ;RETURN


034524                                  T15RT2:
034524                                          SAVREG                          ;SAVE THE REGISTERS
034530  012701  034050                          MOV     #T15PK2,R1              ;START OF THE PACKET
034534  012721  100206                          MOV     #100206,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK, IE
034540  012721  034060                          MOV     #T15BF2,(R1)+           ;ADDRESS OF DATA BLOCK
034544  005021                                  CLR     (R1)+                   ;EXTENDED ADDRESS
034546  012721  000006                          MOV     #6.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
034552  005021                                  CLR     (R1)+
034554  005021                                  CLR     (R1)+
034556  005011                                  CLR     (R1)
034560  000207                                  RTS     PC                      ;RETURN
034562                                          ENDTST
034562
034562  104401                                                  L10047:
                                                                        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
                                ;--


034564                                  BGNTST
034564                                                                          T5::
034564  012700  036642                  MOV     #TST16ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
034570  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
034574  012737  000012  002216          MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
034602                          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
                        ;--
034602                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
034602                                                                  T5.1:
034602  104402                                                                  TRAP    C$BSUB

034604                  5$:
                        ;       Write to TSSR register to soft initialize the controller
034604  004737  015774          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
034610  103405                  BCS     10$                     ;BR IF SOFT INIT OKAY
```

```
1679 034612  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1680 034614                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     034614  104455                                                                TRAP    C$ERDF
     034616  000764                                                                .WORD   500
     034620  003652                                                                .WORD   SFIERR
     034622  012034                                                                .WORD   SFIMSG
1681                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
1682 034624  004737  040010          10$:    JSR     PC,T16REST              ;RESTORE PACKET DEFAULTS
1683 034630  005037  002222                  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1684 034634  012704  040170                  MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
1685 034640  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
1686 034644                                 FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
1687 034660  103407                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1688 034662  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1689 034664                                NEXT.ERRNO
1690 034664                                12$:    ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     034664  104455                                                                TRAP    C$ERDF
     034666  000765                                                                .WORD   501
     034670  036712                                                                .WORD   T16SSR
     034672  012046                                                                .WORD   PKTSSR
1691 034674  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
1692 034700                                15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
     034700  104406                                                                TRAP    C$CLP1
1693
1694                                ;       If Extended Features Hardware Switch Clear then:
1695                                ;         (* Verify Extended Features switch can be Inverted to SET *)
1696                                ;         REPEAT FOR TEST PATTERNS IN TSTBLK TABLE
1697 034702  012701  040212                  MOV     #T16BFR,R1              ;MESSAGE BUFFER ADDRESS
1698 034706  032761  000200  000012          BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH CLEAR?
1699 034714  001402                        BEQ     20$                     ;BR IF YES
1700 034716  000137  035266                JMP     200$                    ;
1701 034722  012703  002764          20$:    MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
1702                                ;       Do Write Subsystem Write Miscellaneous to SET Extended Features.
1703
1704 034726  004737  040150                  JSR     PC,T16SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
1705 034732  012704  040240                  MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
1706 034736  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
1707 034742  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1708 034746                                 FORCERROR       32$                     ;@@DFORCE ERROR IF FORCER=1
1709 034762  103407                        BCS     40$                     ;BR IF CARRY SET (GOOD RETURN)
1710 034764  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1711 034766                                NEXT.ERRNO
1712 034766                                32$:    ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     034766  104455                                                                TRAP    C$ERDF
     034770  000766                                                                .WORD   502
     034772  036747                                                                .WORD   T162SSR
     034774  012046                                                                .WORD   PKTSSR
1713 034776  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
1714 035002                                40$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
     035002  104406                                                                TRAP    C$CLP1
1715
1716                                ;         DO a WRITE CHARACTERISTICS with an extended characteristic word
1717 035004  012737  125252  002312          MOV     #125252,DATA            ;SETUP TEST DATA FOR EXTENDED WORD
1718 035012  012704  040170                  MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
1719 035016  012764  000020  000006          MOV     #16.,PKBCNT(R4)         ;STORE MESSAGE PACKET SIZE
1720 035024  013737  002312  040210          MOV     DATA,T16DATA+10         ;STORE TEST DATA IN EXTENDED WORD
1721 035032  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
```

```
1722 035036                          FORCERROR     42$                ;@@DFORCE ERROR IF FORCER=1
1723 035052  103407                  BCS     50$                      ;BR IF CARRY SET (GOOD RETURN)
1724 035054  010001                  MOV     R0,R1                    ;SAVE CONTENTS OF TSSR
1725 035056                          NEXT.ERRNO
1726 035056                  42$:    ERRDF   ERRNO,T16SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     035056  104455                                                       TRAP    C$ERDF
     035060  000767                                                       .WORD   503
     035062  036712                                                       .WORD   T16SSR
     035064  012046                                                       .WORD   PKTSSR
1727 035066  005237  002222          INC     FATFLG                   ;SET FATAL ERROR FLAG
1728 035072                  50$:    CKLOOP                           ;LOOP ON ERROR, IF FLAG SET
     035072  104406                                                       TRAP    C$CLP1
1729                         ;         If the TSBA Address Register NOT= Expected  Then Print Error
1730 035074  016501  000000          MOV     TSBA(R5),R1              ;GET TSBA REGISTER CONTENTS
1731 035100  012702  040212          MOV     #T16BFR,R2               ;START OF THE DATA BUFFER
1732 035104  062702  000020  62$:    ADD     #16.,R2                  ;EXPECTED CONTENTS OF TSBA
1733 035110                          FORCERROR     72$,NOTSSR               ;@@DFORCE ERROR IF FORCER=1
1734 035120  020102                  CMP     R1,R2                    ;COMPARE EXPECTED TO RECEIVED
1735 035122  001404                  BEQ     80$                      ;ERROR IF NOT EQUAL
1736 035124                          NEXT.ERRNO
1737 035124                  72$:    ERRHRD  ERRNO,T16TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     035124  104456                                                       TRAP    C$ERHRD
     035126  000770                                                       .WORD   504
     035130  037060                                                       .WORD   T16TSBA
     035132  015474                                                       .WORD   EXPREC
1738 035134                  80$:    CKLOOP                           ;LOOP ON ERROR, IF FLAG SET
     035134  104406                                                       TRAP    C$CLP1
1739                         ;         Compare the controller ram to the extended characteristic word
1740                         ;         If Data word in controller ram NOT= to word sent  Then Print Error
1741 035136  012704  040200          MOV     #T16DATA,R4              ;GET CHARACTERISTIC DATA ADDRESS
1742 035142  004737  011224          JSR     PC,CKRAM2                ;DOES RAM DATA EQUAL DATA SENT?
1743 035146                          FORCERROR     92$                      ;@@DFORCE ERROR IF FORCER=1
1744 035162  103404                  BCS     100$                     ;BR IF YES
1745 035164                          NEXT.ERRNO
1746 035164                  92$:    ERRHRD  ERRNO,PKTRAM,RAMERR      ;REPORT THE RAM ERROR(S)
     035164  104456                                                       TRAP    C$ERHRD
     035166  000771                                                       .WORD   505
     035170  004745                                                       .WORD   PKTRAM
     035172  015510                                                       .WORD   RAMERR
1747 035174                  100$:   CKLOOP                           ;LOOP ON ERROR, IF FLAG SET
     035174  104406                                                       TRAP    C$CLP1
1748                         ;         If Message Buffer Data Length NOT= 12.  Then Print Error
1749 035176  012702  040212          MOV     #T16BFR,R2               ;GET MESSAGE BUFFER ADDRESS
1750 035202  016201  000002          MOV     2(R2),R1                 ;GET RECV DATA FIELD LENGTH
1751 035206  012702  000014          MOV     #12.,R2                  ;GET EXPD DATA FIELD LENGTH
1752 035212                          FORCERROR     112$,NOTSSR              ;@@DFORCE ERROR IF FORCER=1
1753 035222  020102                  CMP     R1,R2                    ;COMPARE EXPECTED TO RECEIVED
1754 035224  001404                  BEQ     120$                     ;ERROR IF NOT EQUAL
1755 035226                          NEXT.ERRNO
1756 035226                  112$:   ERRHRD  ERRNO,T16LEN,EXPREC      ;PRINT THE ERROR & EXPD/RECV
     035226  104456                                                       TRAP    C$ERHRD
     035230  000772                                                       .WORD   506
     035232  037162                                                       .WORD   T16LEN
     035234  015474                                                       .WORD   EXPREC
1757 035236                  120$:   CKLOOP                           ;LOOP ON ERROR, IF FLAG SET
     035236  104406                                                       TRAP    C$CLP1
1758
```

```
1759 035240  004737  015774          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
1760 035244  103405                  BCS     125$                    ;BR IF SOFT INIT OKAY
1761 035246  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1762 035250                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     035250  104455                                                          TRAP    C$ERDF
     035252  000772                                                          .WORD   506
     035254  003652                                                          .WORD   SFIERR
     035256  012034                                                          .WORD   SFIMSG
1763 035260                  125$:   CKLOOP                          ;LOOP IF SELECTED
     035260  104406                                                          TRAP    C$CLP1
1764 035262  000137  035446          JMP     300$                    ;
1765
1766                         ;       (* Verify Extended Features switch can be Inverted to CLEAR *)
1767 035266                  200$:
1768                         ;       Do Write Subsystem Write Miscellaneous to CLEAR Extended Features.
1769 035266  004737  040150          JSR     PC,T16SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
1770 035272  012704  040240          MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
1771 035276  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
1772 035302  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1773 035306                          FORCERROR       232$                    ;@@DFORCE ERROR IF FORCER=1
1774 035322  103407                  BCS     240$                    ;BR IF CARRY SET (GOOD RETURN)
1775 035324  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1776 035326                          NEXT.ERRNO
1777 035326                  232$:   ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     035326  104455                                                          TRAP    C$ERDF
     035330  000773                                                          .WORD   507
     035332  036747                                                          .WORD   T162SSR
     035334  012046                                                          .WORD   PKTSSR
1778 035336  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
1779 035342                  240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035342  104406                                                          TRAP    C$CLP1
1780
1781                         ;         DO a WRITE CHARACTERISTICS without an extended characteristic word
1782 035344  012704  040170          MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
1783 035350  012764  000016  000006  MOV     #14.,PKBCNT(R4)         ;STORE MESSAGE PACKET SIZE
1784 035356  004737  010662          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
1785 035362                          FORCERROR       242$                    ;@@DFORCE ERROR IF FORCER=1
1786 035376  103407                  BCS     250$                    ;BR IF CARRY SET (GOOD RETURN)
1787 035400  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1788 035402                          NEXT.ERRNO
1789 035402                  242$:   ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     035402  104455                                                          TRAP    C$ERDF
     035404  000774                                                          .WORD   508
     035406  036712                                                          .WORD   T16SSR
     035410  012046                                                          .WORD   PKTSSR
1790 035412  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
1791 035416                  250$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035416  104406                                                          TRAP    C$CLP1
1792                         ;         If Message Buffer Data Length NOT= 10.  Then Print Error
1793 035420  013701  040214          MOV     T16BFR+2,R1             ;GET RECV DATA FIELD LENGTH
1794 035424  012702  000012          MOV     #10.,R2                 ;GET EXPD DATA FIELD LENGTH
1795 035430  020102                  CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1796 035432  001404                  BEQ     270$                    ;ERROR IF NOT EQUAL
1797 035434                          NEXT.ERRNO
1798 035434                  262$:   ERRHRD  ERRNO,T16LEN,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     035434  104456                                                          TRAP    C$ERHRD
     035436  000775                                                          .WORD   509
```

```
        035440  037162                                                          .WORD   T16LEN
        035442  015474                                                          .WORD   EXPREC
  1799  035444                        270$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        035444  104406                                                          TRAP    C$CLP1
  1800
  1801
  1802                                ;       (* Verify Function Reject when Message Buffer 21-19 are non-zero *)
  1803                                ;       Write to TSSR register to soft initialize the controller
  1804  035446                        300$:
  1805                                ;       REPEAT FOR MESSAGE BUFFER ADDRESS bits <21:19> FROM 0 TO 7
  1806  035446  012737  000001  002312 320$:  MOV     #1,DATA                 ;START AT BITS<21:19>=001
  1807                                ;       DO a WRITE CHARACTERISTICS with a message address bit<21:19> non-zero
  1808  035454                        325$:
  1809  035454  012704  040170               MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
  1810  035460  012764  000016  000006       MOV     #14.,PKBCNT(R4)         ;STORE MESSAGE PACKET SIZE
  1811  035466  013700  002312               MOV     DATA,R0                 ;GET TEST DATA
  1812          000003                       .REPT   3
  1813                                       ASL     R0                      ;SHIFT INTO BITS 21:19
  1814                                       .ENDR
  1815  035500  010037  040202               MOV     R0,T16DATA+2            ;STORE BUFFER ADDRESS BITS 21:19
  1816  035504  010465  000000               MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
  1817  035510  004737  016250               JSR     PC,WAITF                ;WAIT FOR SSR
  1818  035514                               FORCERROR       342$            ;@@DFORCE ERROR IF FORCER=1
  1819  035530  103407                       BCS     350$                    ;BR IF CARRY SET (GOOD RETURN)
  1820  035532  010001                       MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  1821  035534                               NEXT.ERRNO
  1822  035534                        342$:  ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        035534  104455                                                          TRAP    C$ERDF
        035536  000776                                                          .WORD   510
        035540  036712                                                          .WORD   T16SSR
        035542  012046                                                          .WORD   PKTSSR
  1823  035544  005237  002222               INC     FATFLG                  ;SET FATAL ERROR FLAG
  1824  035550                        350$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        035550  104406                                                          TRAP    C$CLP1
  1825
  1826                                ;       If TSSR termination code NOT= Function Reject  Then Print Error
  1827  035552  016501  000002               MOV     TSSR(R5),R1             ;GET RECV TSSR
  1828  035556  010102                       MOV     R1,R2                   ;COPY RECV TSSR
  1829  035560  042702  000016               BIC     #TERCLS,R2              ;CLEAR TC<2:0> EXPD
  1830  035564  052702  000006               BIS     #TSREJ,R2               ;SET EXPD TC<2:0>= FUNCTION REJECT
  1831  035570                               FORCERROR       352$,NOTSSR             ;@@DFORCE ERROR IF FORCER=1
  1832  035600  020102                       CMP     R1,R2                   ;EXPD EQUAL RECV?
  1833  035602  001404                       BEQ     360$                    ;BR IF YES
  1834  035604                               NEXT.ERRNO
  1835  035604                        352$:  ERRHRD  ERRNO,T16REJ,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        035604  104456                                                          TRAP    C$ERHRD
        035606  000777                                                          .WORD   511
        035610  037274                                                          .WORD   T16REJ
        035612  012046                                                          .WORD   PKTSSR
  1836  035614                        360$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        035614  104406                                                          TRAP    C$CLP1
  1837  035616                               FORCEXIT        370$
  1838  035626  005237  002312               INC     DATA                    ;GET NEXT TST PATTERN
  1839  035632  023727  002312  000007       CMP     DATA,#7                 ;DONE ALL DATA?
  1840  035640  101002                       BHI     370$                    ;BR IF YES
  1841  035642  000137  035454               JMP     325$                    ;DO ANOTHER TEST PATTERN
  1842                                ;       END-REPEAT
```

```
1843 035646                          370$:
1844 035646                                  ENDSUB                  ;////////////// END SUBTEST //////////
     035646                                                                  L10054:
     035646  104403                                                             TRAP    C$ESUB
1845
1846 035650  005737  002222                     TST     FATFLG          ;ANY FATAL ERRORS ?
1847 035654  001402                             BEQ     460$            ;BRANCH IF NOT
1848 035656  004737  017202                     JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
1849 035662                          460$:
1850
1851
1852
1853
1854
```

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

                        ;++
                        ; TEST 5: SUBTEST 2:
                        ;
                        ; SUBTEST DESCRIPTION:
                        ;
                        ;       This subtest verifies that timers A,B can be reset
                        ;       and that Timer A is twice the frequency of Timer B.
                        ;       Timer A has a period of 25 microseconds and Timer B
                        ;       has a period of 50 microseconds. The timers are
                        ;       checked at 1, 28, 53, and 78 micoseconds.
                        ;
                        ; TEST STEPS:
                        ;
                        ;
                        ;       Write to TSSR register to soft initialize the controller
                        ;       Do WRITE CHARACTERISTICS to setup a Message Buffer
                        ;       (* Verify Timers A,B after RESET TIMER with 0 microsecond delay *)
                        ;       Do a Write Control RESET TIMER with 1 microsecond delay
                        ;       Do a Write Subsystem READ STATUS
                        ;       If Timer A NOT= 0  Then Print Error
                        ;       If Timer B NOT= 0  Then Print Error
                        ;       (* Verify Timers A,B after RESET TIMER with 28 microsecond delay *)
                        ;       Do a Write Control RESET TIMER with 28 microsecond delay
                        ;       If Timer A NOT= 1  Then Print Error
                        ;       If Timer B NOT= 1  Then Print Error
                        ;       Do a Write Control RESET TIMER with 53 microsecond delay
                        ;       If Timer A NOT= 0  Then Print Error
                        ;       If Timer B NOT= 1  Then Print Error
                        ;       Do a Write Control RESET TIMER with 78 microsecond delay
                        ;       If Timer A NOT= 1  Then Print Error
                        ;       If Timer B NOT= 0  Then Print Error
                        ;--
035662                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
035662                                                                  T5.2:
035662  104402                                                          TRAP    C$BSUB
                        ;
                        ;       Write to TSSR register to soft initialize the controller
                        ;
035664                  5$:
035664  004737  015774          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
035670  103405                  BCS     10$                     ;BR IF SOFT INIT OKAY
035672  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
035674                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
035674  104455                                                          TRAP    C$ERDF
035676  000777                                                          .WORD   511
035700  003652                                                          .WORD   SFIERR
035702  012034                                                          .WORD   SFIMSG
                        ;
                        ;       Do WRITE CHARACTERISTICS to setup a Message Buffer
                        ;
035704  004737  040010  10$:     JSR     PC,T16REST              ;RESTORE PACKET DEFAULTS
035710  005037  002222          CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
035714  012704  040170          MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
035720  012764  000010  000006  MOV     #8.,PKBCNT(R4)          ;MESSAGE PACKET SIZE NO EXTEND
035726  004737  010662          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
035732                          FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
035746  103407                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
035750  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
035752                          NEXT.ERRNO
```

```
1907 035752                          12$:   ERRDF   ERRNO,T16SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     035752  104455                                                              TRAP    C$ERDF
     035754  001000                                                              .WORD   512
     035756  036712                                                              .WORD   T16SSR
     035760  012046                                                              .WORD   PKTSSR
1908 035762  005237  002222                 INC     FATFLG                     ;SET FATAL ERROR FLAG
1909 035766                          15$:   CKLOOP                             ;LOOP ON ERROR, IF FLAG SET
     035766  104406                                                              TRAP    C$CLP1
1910
1911                                 ;      (* Verify Timers A,B after RESET TIMER with 1 microsecond delay *)
1912                                 ;      Do a Write Control RESET TIMER with 1 microsecond delay
1913 035770  012700  000001                 MOV     #MS.RSD,R0                 ;RESET TIMER COMMAND
1914 035774  013701  036632                 MOV     T16D01,R1                  ;1 MICROSECOND DELAY
1915 036000  004737  040122                 JSR     PC,T16WMISC                ;SETUP T16PK2 COMMAND PACKET
1916 036004  012704  040240                 MOV     #T16PK2,R4                 ;GET WRITE SUBSYSTEM COMMAND PACKET
1917 036010  010465  000000                 MOV     R4,TSDB(R5)                ;SET THE PACKET ADDRESS TO EXECUTE
1918 036014  004737  016336                 JSR     PC,CHKTSSR                 ;WAIT FOR SSR TO SET
1919 036020                                 FORCERROR       32$                        ;@@DFORCE ERROR IF FORCER=1
1920 036034  103407                         BCS     40$                        ;BR IF CARRY SET (GOOD RETURN)
1921 036036  010001                         MOV     R0,R1                      ;SAVE CONTENTS OF TSSR
1922 036040                                 NEXT.ERRNO
1923 036040                          32$:   ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     036040  104455                                                              TRAP    C$ERDF
     036042  001001                                                              .WORD   513
     036044  036747                                                              .WORD   T162SSR
     036046  012046                                                              .WORD   PKTSSR
1924 036050  005237  002222                 INC     FATFLG                     ;SET FATAL ERROR FLAG
1925 036054                          40$:   CKLOOP                             ;LOOP ON ERROR, IF FLAG SET
     036054  104406                                                              TRAP    C$CLP1
1926                                 ;      If Timer A NOT= 0  Then Print Error
1927                                 ;      If Timer B NOT= 0  Then Print Error
1928 036056  005002                         CLR     R2                         ;INIT EXPD
1929 036060  042702  000010                 BIC     #S2.ATIM,R2                ;TIMER A EXPD=0
1930 036064  042702  000004                 BIC     #S2.BTIM,R2                ;TIMER B EXPD=0
1931 036070  012700  040232                 MOV     #T16BFSTA,R0               ;GET RECV READ STATUS
1932 036074  016001  000002                 MOV     2(R0),R1                   ;GET RECV BYTE 2
1933 036100  042701  177763                 BIC     #^C<S2.ATIM!S2.BTIM>,R1     ;SAVE TIMER A:B RECV ONLY
1934 036104                                 FORCERROR       72$,NOTSSR         ;@@D
1935 036114  020201                         CMP     R2,R1                      ;EXPD EQUAL RECV?
1936 036116  001404                         BEQ     80$                        ;BR IF YES
1937 036120                                 NEXT.ERRNO
1938 036120                          72$:   ERRHRD  ERRNO,T16T01,TIMEXP     ;REPORT ERROR
     036120  104456                                                              TRAP    C$ERHRD
     036122  001002                                                              .WORD   514
     036124  037411                                                              .WORD   T16T01
     036126  015552                                                              .WORD   TIMEXP
1939 036130                          80$:   CKLOOP                             ;LOOP ON ERROR, IF FLAG SET
     036130  104406                                                              TRAP    C$CLP1
1940
1941                                 ;      Do a Write Control RESET TIMER with 28 microsecond delay
1942 036132  012700  000001                 MOV     #MS.RSD,R0                 ;RESET TIMER COMMAND
1943 036136  013701  036634                 MOV     T16D28,R1                  ;28 MICROSECOND DELAY
1944 036142  004737  040122                 JSR     PC,T16WMISC                ;SETUP T16PK2 COMMAND PACKET
1945 036146  012704  040240                 MOV     #T16PK2,R4                 ;GET WRITE SUBSYSTEM COMMAND PACKET
1946 036152  010465  000000                 MOV     R4,TSDB(R5)                ;SET THE PACKET ADDRESS TO EXECUTE
1947 036156  004737  016336                 JSR     PC,CHKTSSR                 ;WAIT FOR SSR TO SET
1948 036162                                 FORCERROR       112$                       ;@@DFORCE ERROR IF FORCER=1
```

```
1949 036176  103407                    BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
1950 036200  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1951 036202                            NEXT.ERRNO
1952 036202                    112$:   ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     036202  104455                                                            TRAP    C$ERDF
     036204  001003                                                            .WORD   515
     036206  036747                                                            .WORD   T162SSR
     036210  012046                                                            .WORD   PKTSSR
1953 036212  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
1954 036216                    120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036216  104406                                                            TRAP    C$CLP1
1955                           ;       If Timer A NOT= 1  Then Print Error
1956                           ;       If Timer B NOT= 1  Then Print Error
1957 036220  005002                    CLR     R2                      ;INIT EXPD
1958 036222  052702  000010            BIS     #S2.ATIM,R2             ;TIMER A EXPD=1
1959 036226  052702  000004            BIS     #S2.BTIM,R2             ;TIMER B EXPD=1
1960 036232  012700  040232            MOV     #T16BFSTA,R0            ;GET RECV READ STATUS
1961 036236  016001  000002            MOV     2(R0),R1                ;GET RECV BYTE 2
1962 036242  042701  177763            BIC     #^C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
1963 036246                            FORCERROR       172$,NOTSSR     ;@@D
1964 036256  020201                    CMP     R2,R1                   ;EXPD EQUAL RECV?
1965 036260  001404                    BEQ     180$                    ;BR IF YES
1966 036262                            NEXT.ERRNO
1967 036262                    172$:   ERRHRD  ERRNO,T16T28,TIMEXP     ;REPORT ERROR
     036262  104456                                                            TRAP    C$ERHRD
     036264  001004                                                            .WORD   516
     036266  037510                                                            .WORD   T16T28
     036270  015552                                                            .WORD   TIMEXP
1968 036272                    180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036272  104406                                                            TRAP    C$CLP1
1969
1970                           ;       Do a Write Control RESET TIMER with 53 microsecond delay
1971 036274  012700  000001            MOV     #MS.RSD,R0              ;RESET TIMER COMMAND
1972 036300  013701  036636            MOV     T16D53,R1               ;53 MICROSECOND DELAY
1973 036304  004737  040122            JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
1974 036310  012704  040240            MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
1975 036314  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
1976 036320  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1977 036324                            FORCERROR       212$            ;@@DFORCE ERROR IF FORCER=1
1978 036340  103407                    BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
1979 036342  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1980 036344                            NEXT.ERRNO
1981 036344                    212$:   ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     036344  104455                                                            TRAP    C$ERDF
     036346  001005                                                            .WORD   517
     036350  036747                                                            .WORD   T162SSR
     036352  012046                                                            .WORD   PKTSSR
1982 036354  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
1983 036360                    220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036360  104406                                                            TRAP    C$CLP1
1984                           ;       If Timer A NOT= 0  Then Print Error
1985                           ;       If Timer B NOT= 1  Then Print Error
1986 036362  005002                    CLR     R2                      ;INIT EXPD
1987 036364  042702  000010            BIC     #S2.ATIM,R2             ;TIMER A EXPD=0
1988 036370  052702  000004            BIS     #S2.BTIM,R2             ;TIMER B EXPD=1
1989 036374  012700  040232            MOV     #T16BFSTA,R0            ;GET RECV READ STATUS
1990 036400  016001  000002            MOV     2(R0),R1                ;GET RECV BYTE 2
```

```
1991 036404  042701  177763            BIC     #^C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
1992 036410                            FORCERROR       272$,NOTSSR     ;aaD
1993 036420  020201                    CMP     R2,R1                   ;EXPD EQUAL RECV?
1994 036422  001404                    BEQ     280$                    ;BR IF YES
1995 036424                            NEXT.ERRNO
1996 036424                    272$:   ERRHRD  ERRNO,T16T53,TIMEXP     ;REPORT ERROR
     036424  104456                                                            TRAP    C$ERHRD
     036426  001006                                                            .WORD   518
     036430  037610                                                            .WORD   T16T53
     036432  015552                                                            .WORD   TIMEXP
1997 036434                    280$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036434  104406                                                            TRAP    C$CLP1
1998                           ;       Do a Write Control RESET TIMER with 78 microsecond delay
1999 036436  012700  000001            MOV     #MS.RSD,R0              ;RESET TIMER COMMAND
2000 036442  013701  036640            MOV     T16D78,R1               ;78 MICROSECOND DELAY
2001 036446  004737  040122            JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
2002 036452  012704  040240            MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2003 036456  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2004 036462  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2005 036466                            FORCERROR       312$                    ;aaDFORCE ERROR IF FORCER=1
2006 036502  103407                    BCS     320$                    ;BR IF CARRY SET (GOOD RETURN)
2007 036504  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2008 036506                            NEXT.ERRNO
2009 036506                    312$:   ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     036506  104455                                                            TRAP    C$ERDF
     036510  001007                                                            .WORD   519
     036512  036747                                                            .WORD   T162SSR
     036514  012046                                                            .WORD   PKTSSR
2010 036516  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
2011 036522                    320$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036522  104406                                                            TRAP    C$CLP1
2012                           ;       If Timer A NOT= 1  Then Print Error
2013                           ;       If Timer B NOT= 0  Then Print Error
2014 036524  005002                    CLR     R2                      ;INIT EXPD
2015 036526  052702  000010            BIS     #S2.ATIM,R2             ;TIMER A EXPD=1
2016 036532  042702  000004            BIC     #S2.BTIM,R2             ;TIMER B EXPD=0
2017 036536  012700  040232            MOV     #T16BFSTA,R0            ;GET RECV READ STATUS
2018 036542  016001  000002            MOV     2(R0),R1                ;GET RECV BYTE 2
2019 036546  042701  177763            BIC     #^C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
2020 036552                            FORCERROR       372$,NOTSSR     ;aaD
2021 036562  020201                    CMP     R2,R1                   ;EXPD EQUAL RECv?
2022 036564  001404                    BEQ     380$                    ;BR IF YES
2023 036566                            NEXT.ERRNO
2024 036566                    372$:   ERRHRD  ERRNO,T16T78,TIMEXP     ;REPORT ERROR
     036566  104456                                                            TRAP    C$ERHRD
     036570  001010                                                            .WORD   520
     036572  037710                                                            .WORD   T16T78
     036574  015552                                                            .WORD   TIMEXP
2025 036576                    380$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036576  104406                                                            TRAP    C$CLP1
2026
2027 036600                            ENDSUB                          ;////////////// END SUBTEST //////////
     036600                                                                  L10055:
     036600  104403                                                            TRAP    C$ESUB
2028
2029 036602  005737  002222            TST     FATFLG                  ;ANY FATAL ERRORS ?
2030 036606  001402                    BEQ     460$                    ;BRANCH IF NOT
```

```
2031 036610  004737  017202            JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
2032 036614  004737  016456  460$:     JSR     PC,TSTLOOP      ;SHOULD WE DO ITERATIONS?
2033 036620  103002                    BCC     465$            ;BR IF NO
2034 036622  000137  034602            JMP     T16LOOP         ;LOOP UNTIL ITERATIONS DONE
2035 036626                    465$:
2036
2037
2038 036626                    EXIT    TST             ;////////// EXIT TEST //////////
     036626  104432                                                    TRAP    C$EXIT
     036630  001524                                                    .WORD   L10053-.
2039
2040
```

```
                        ;+
                        ;LOCAL STORAGE FOR THIS TEST
                        ;-
036632  000001          T16D01:         .WORD   1               ;1 MICROSECOND DELAY (ACTUALLY .8 MIC)
036634  000040          T16D28:         .WORD   40              ;28 MICROSECOND DELAY (.8 MICROS PER)
036636  000076          T16D53:         .WORD   76              ;53 MICROSECOND
036640  000142          T16D78:         .WORD   142             ;78 MICROSECOND
                        ;+
                        ;LOCAL TEXT MESSAGES FOR TEST
                        ;-

036642      105     170     164 TST16ID:        .ASCIZ  'Extended Features Switch and Timers A,B'
036712      127     122     111 T16SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
036747      127     122     111 T162SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
037013      127     122     111 T163SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
037060      102     165     163 T16TSBA:.ASCIZ  'Bus Address Register (TSBA) Incorrect after Write Characteristics'
037162      104     141     164 T16LEN:.ASCIZ   'Data Field Length in Message Buffer Incorrect after Write Characteristics'
037274      124     123     123 T16REJ:.ASCIZ   'TSSR Function Reject Not Returned When Non-Existent Buffer Address Specifie
037411      124     151     155 T16T01:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 1 microsecond Delay'
037510      124     151     155 T16T28:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 28 microsecond Delay'
037610      124     151     155 T16T53:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 53 microsecond Delay'
037710      124     151     155 T16T78:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 78 microsecond Delay'
                                    .EVEN

                        ;+
                        ; SET DEFAULT PACKET
                        ;-
040010                  T16REST:
040010  012700  040170          MOV     #T16PACKET,R0           ;PACKET ADDRESS
040014  012720  100004          MOV     #100004,(R0)+           ;WRITE CHARACTERISTICS WITH ACK
040020  012720  040200          MOV     #T16DATA,(R0)+          ;ADDRESS OF CHAR DATA BLOCK
040024  005020                  CLR     (R0)+                   ;EXTENDED ADDRESS
040026  012720  000012          MOV     #10.,(R0)+              ;SIZE OF MESSAGE PACKET
040032  012720  040212          MOV     #T16BFR,(R0)+           ;MESSAGE BUFFER ADDRESS
040036  005020                  CLR     (R0)+                   ;CLEAR EXTENDED BUFFER ADDRESS
040040  012720  000024          MOV     #20.,(R0)+              ;LENGTH OF MESSAGE BUFFER
040044  005020                  CLR     (R0)+                   ;CLEAR ESS,ENB,EAI,ERI
040046  005010                  CLR     (R0)                    ;CLEAR EXTENDED FEATURES WORD
040050  005037  040212          CLR     T16BFR                  ;CLEAR 1ST LOCATION IN MESSAGE BUFFER
040054  000207                  RTS     PC                      ;

                        ;+
                        ; CLEAR MESSAGE BUFFER
                        ;-
040056                  T16CLRBUF:                              ;
040056                          SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
040062  012701  040212          MOV     #T16BFR,R1              ;GET MESSAGE BUFFER ADDRESS
040066  012702  000026          MOV     #T16BEND-T16BFR,R2      ;SIZE OF MESSAGE BUFFER IN BYTES
040072  105021          10$:    CLRB    (R1)+                   ;CLEAR A BYTE
040074  005302                  DEC     R2                      ;DONE?
040076  003375                  BGT     10$                     ;BR IF NO
040100  000207                  RTS     PC                      ;RETURN

                        ;+
                        ; SETUP T16PK2 PACKET FOR READ STATUS
                        ;-
040102                  T16SRD:
```

```
2099 040102  004737  040056                JSR     PC,T16CLRBUF            ;CLEAR MESSAGE BUFFER
2100 040106  012700  040250                MOV     #T16DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
2101 040112  112720  000005                MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
2102 040116  105010                        CLRB    (R0)                    ;CLEAR BSEL1
2103 040120  000207                        RTS     PC                      ;RETURN
2104
2105
2106                                ;+
2107                                ; SETUP T16PK2 PACKET FOR WRITE MISC.
2108                                ;
2109                                ; INPUT:
2110                                ;       R0      CONTAINS WRITE MISC FUNCTION CODE (BSEL1)
2111                                ;       R1      CONTAINS DELAY (TIMES 800 NS) FOR BSEL2
2112                                ;-
2113 040122                        T16WMISC:
2114 040122                                SAVREG                          ;SAVE R1-R5 UNITL NEXT RETURN
2115 040126  004737  040056                JSR     PC,T16CLRBUF            ;CLEAR MESSAGE BUFFER
2116 040132  012702  040250                MOV     #T16DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
2117 040136  112722  000010                MOVB    #PW.WMISC,(R2)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
2118 040142  110022                        MOVB    R0,(R2)+                ;STORE WRITE MISC CODE IN BSEL1
2119 040144  110112                        MOVB    R1,(R2)                 ;STORE DELAY (RESET TIMER) IN BSEL2
2120 040146  000207                        RTS     PC                      ;RETURN
2121                                ;+
2122                                ; SETUP T16PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
2123                                ;-
2124 040150                        T16SEXT:
2125 040150  012700  040250                MOV     #T16DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
2126 040154  112720  000010                MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
2127 040160  112710  000200                MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
2128 040164  000207                        RTS     PC                      ;RETURN
2129
2130
2131
2132
2134         040170                        .=<.+10>&177770
2136                                ;
2137                                ;WRITE CHARACTERISTICS COMMAND PACKET
2138                                ;
2139 040170                        T16PACKET:                              ;COMMAND PACKET FOR TEST
2140 040170  100004                        .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
2141 040172  040200                        .WORD   T16DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
2142 040174  000000                        .WORD   0
2143 040176  000012                        .WORD   10.                     ;MESSAGE PACKET SIZE
2144
2145 040200                        T16DATA:                                ;CHARACTERISTICS DATA BLOCK
2146 040200  040212                        .WORD   T16BFR                  ;ADDRESS OF MESSAGE BUFFER
2147 040202  000000                        .WORD   0
2148 040204  000024                        .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
2149 040206  000000                        .WORD   0                       ;ESS,ENB,EAI,ERI
2150 040210  000000                        .WORD   0                       ;EXTENDED FEATURES WORD
2151
2152
2153                                ;MESSAGE BUFFER
2154
2155 040212                        T16BFR:                                 ;BEGIN MESSAGE BUFFER
2156 040212  000000                        .WORD   0                       ;MESSAGE TYPE
2157 040214  000000                        .WORD   0                       ;DATA FIELD LENGTH
```

```
 2158 040216  000000                    .WORD   0                       ;RBPCR
 2159 040220  000000                    .WORD   0                       ;XST0
 2160 040222  000000                    .WORD   0                       ;XST1
 2161 040224  000000                    .WORD   0                       ;XST2
 2162 040226  000000                    .WORD   0                       ;XST3
 2163 040230  000000                    .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
 2164 040232                    T16BFSTA: .BLKB 6.                        ;READ STATUS AND WRITE FIFO BUFFER
 2165 040240                    T16BEND:                                  ;END OF MESSAGE BUFFER
 2166                           ;
 2167                           ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
 2168                           ;
 2172 040240                    T16PK2:
 2173 040240  100006                    .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
 2174 040242  040250                    .WORD   T16DT2                  ;LOW ADDRESS OF DATA BLOCK
 2175 040244  000000                    .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
 2176 040246  000012                    .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
 2177
 2178 040250                    T16DT2:                                   ;DATA BLOCK
 2179 040250     000                    .BYTE   0                       ;BSEL0
 2180 040251     000                    .BYTE   0                       ;BSEL1
 2181 040252  000000                    .WORD   0                       ;SEL2
 2182 040254                            .BLKB   64.                     ;WRITE FIFO DATA OUTPUT BUFFER
 2183
 2184
 2185 040354                    ENDTST
      040354                                                    L10053:
      040354  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
                        ;--


040356                          BGNTST
040356                                                          T6::
040356  012700  046606          MOV     #TST17ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
040362  004737  016510          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
040366  012737  000012  002216  MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
040374  004737  017274          JSR     PC,KTOFF                ;SHUT OFF MEMORY MANAGEMENT
040400  005037  003134          CLR     KTENABLE                ;REALLY SHUT DOWN KT-11
040404                  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
                        ;--
040404                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
                                                                                T6.1:
040404  104402                                                                 TRAP    C$BSUB

                        ;       Write to TSSR register to soft initialize the controller
                        5$:
040406  004737  015774          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
040412  103405                  BCS     10$                     ;BR IF SOFT INIT OKAY
040414  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040416                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
040416  104455                                                                  TRAP    C$ERDF
040420  001130                                                                  .WORD   600
040422  003652                                                                  .WORD   SFIERR
040424  012034                                                                  .WORD   SFIMSG
                        ;       Do a WRITE CHARACTERISTICS to setup a message buffer
040426  005037  002222  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
040432  012704  050200          MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
040436  004737  010662          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
040442                          FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
040456  103407                  BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
040460  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040462                          NEXT.ERRNO
040462                  42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
040462  104455                                                                  TRAP    C$ERDF
040464  001131                                                                  .WORD   601
040466  046625                                                                  .WORD   T17SSR
040470  012046                                                                  .WORD   PKTSSR
040472  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
040476                  50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
040476  104406                                                                  TRAP    C$CLP1

                        ;       Do a Write Subsystem READ STATUS
```

```
2271 040500  004737  047764                JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2272 040504  012704  050350                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2273 040510  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2274 040514  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2275 040520                                FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
2276 040534  103407                        BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
2277 040536  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2278 040540                                NEXT.ERRNO
2279 040540                        62$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     040540  104455                                                                      TRAP    C$ERDF
     040542  001132                                                                      .WORD   602
     040544  046726                                                                      .WORD   T173SSR
     040546  012046                                                                      .WORD   PKTSSR
2280 040550  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
2281 040554                        70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     040554  104406                                                                      TRAP    C$CLP1
2282                               ;       Set WORDS 0-7  of expd message buffer = to recv since not testing
2283 040556  004737  050146                JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2284 040562  012701  046402                MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2285 040566  012702  050242                MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2286 040572  012221                        MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2287 040574  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2288 040576  052711  000020                BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= TRUE
2289 040602  042711  000040                BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= FALSE
2290 040606  042711  000200                BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = FALSE
2291                               ;       If Input Ready NOT=1 then Print Error
2292                               ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2293 040612  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2294 040614  012701  050222                MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2295 040620  012702  046362                MOV     #T17EXP,R2              ;EXPD ADDRESS
2296 040624  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2297 040630  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2298 040634                                FORCERROR       82$,NOTSSR              ;@@D
2299 040644  103404                        BCS     90$                     ;BR IF YES
2300 040646                                NEXT.ERRNO
2301 040646                        82$:    ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
     040646  104456                                                                      TRAP    C$ERHRD
     040650  001133                                                                      .WORD   603
     040652  047145                                                                      .WORD   T171CMP
     040654  012350                                                                      .WORD   MSGSTAT
2302 040656                        90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     040656  104406                                                                      TRAP    C$CLP1
2303
2304 040660                                ENDSUB                          ;////////////// END SUBTEST //////////
     040660                                                                          L10057:
     040660  104403                                                                      TRAP    C$ESUB
2305
2306 040662  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
2307 040666  001402                        BEQ     160$                    ;BRANCH IF NOT
2308 040670  004737  017202                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2309 040674                        160$:
2310
```

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

                        ;++
                        ; TEST 6: SUBTEST 2:
                        ;
                        ; SUBTEST DESCRIPTION:
                        ;
                        ;       This subtest verifies the ability of the FIFO to correctly
                        ;       pass a single data byte from input to output. For each
                        ;       of 256 data values (0-377 octal) the following is done:
                        ;               1. Initial FIFO status is checked
                        ;               2. The Write FIFO function, specifying a count of
                        ;                  one byte to be written is executed.
                        ;               3. Read Status is executed and FIFO status is checked.
                        ;               4. Read FIFO is executed and the data and final status
                        ;                  is checked.
                        ;
                        ; TEST STEPS:
                        ;
                        ; BEGIN
                        ;       Write to TSSR to soft initialize
                        ;       Do a WRITE CHARACTERISTICS to setup a message buffer
                        ;       Do a Write Subsystem READ STATUS
                        ;       If Input Ready  NOT=1   Then Print Error
                        ;       If Output Ready NOT=0   Then Print Error
                        ;       If Data In Miss NOT=0   Then Print Error
                        ;
                        ;  REPEAT FOR DATA FROM 0 TO 377 OCTAL
                        ;    BEGIN
                        ;       Do a Write Subsystem WRITE NPR to set tape direction out
                        ;       Do a Write Subsystem WRITE FIFO with byte count equal to 1
                        ;       Do a Write Subsystem READ STATUS
                        ;       If Input Ready  NOT=1   Then Print Error
                        ;       If Output Ready NOT=1   Then Print Error
                        ;       If Data In Miss NOT=0   Then Print Error
                        ;       Do Write Subsystem READ FIFO with byte count equal to 1
                        ;       If Data read from FIFO NOT= to Data sent Then Print Error
                        ;       Do a Write Subsystem READ STATUS
                        ;       If Input Ready  NOT=1   Then Print Error
                        ;       If Output Ready NOT=0   Then Print Error
                        ;       If Data In Miss NOT=0   Then Print Error
                        ;    END
                        ;  END
                        ;--
040674                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
040674                                                                  T6.2:
040674  104402                                                                  TRAP    C$BSUB

                        ;       Write to TSSR register to soft initialize the controller
                        5$:
040676  004737  015774          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
040702  103405                  BCS     10$                     ;BR IF SOFT INIT OKAY
040704  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040706                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
040706  104455                                                                  TRAP    C$ERDF
040710  001133                                                                  .WORD   603
040712  003652                                                                  .WORD   SFIERR
```

```
040714  012034                                                          .WORD   SFIMSG
2364                            ;       Do a WRITE CHARACTERISTICS to setup a message buffer
2365 040716  005037  002222      10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
2366 040722  012704  050200              MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
2367 040726  004737  010662              JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
2368 040732                              FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
2369 040746  103407                      BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
2370 040750  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2371 040752                              NEXT.ERRNO
2372 040752                      42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
040752  104455                                                                  TRAP    C$ERDF
040754  001134                                                                  .WORD   604
040756  046625                                                                  .WORD   T17SSR
040760  012046                                                                  .WORD   PKTSSR
2373 040762  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
2374 040766                      50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
040766  104406                                                                  TRAP    C$CLP1
2375                            ;       Do a Write Subsystem READ STATUS
2376 040770  004737  047764              JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2377 040774  012704  050350              MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2378 041000  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2379 041004  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2380 041010                              FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
2381 041024  103407                      BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
2382 041026  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2383 041030                              NEXT.ERRNO
2384 041030                      62$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
041030  104455                                                                  TRAP    C$ERDF
041032  001135                                                                  .WORD   605
041034  046726                                                                  .WORD   T173SSR
041036  012046                                                                  .WORD   PKTSSR
2385 041040  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
2386 041044                      70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
041044  104406                                                                  TRAP    C$CLP1
2387                            ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2388 041046  004737  050146              JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2389 041052  012701  046402              MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2390 041056  012702  050242              MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2391 041062  012221                      MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2392 041064  011211                      MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2393 041066  052711  000020              BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= TRUE
2394 041072  042711  000040              BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= FALSE
2395 041076  042711  000200              BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = FALSE
2396                            ;       If Input Ready NOT=1 then Print Error
2397                            ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2398 041102  005000                      CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2399 041104  012701  050222              MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2400 041110  012702  046362              MOV     #T17EXP,R2              ;EXPD ADDRESS
2401 041114  012703  000024              MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2402 041120  004737  011500              JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2403 041124                              FORCERROR       82$,NOTSSR              ;@@D
2404 041134  103404                      BCS     90$                     ;BR IF YES
2405 041136                              NEXT.ERRNO
2406 041136                      82$:    ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
041136  104456                                                                  TRAP    C$ERHRD
041140  001136                                                                  .WORD   606
041142  047145                                                                  .WORD   T171CMP
```

```
                                                                                .WORD   MSGSTAT
2407 041146                             90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041146  104406                                                             TRAP    C$CLP1
2408
2409                                    ;  Repeat for DATA from 0 to 377
2410 041150  012737  000000  002312             MOV     #0,DATA                 ;GET FIRST DATA
2411 041156                             100$:                                   ;REPEAT LABEL
2412                                    ;       Do a Write Subsystem WRITE NPR to set tape direction out
2413 041156  012700  000100                     MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
2414 041162  004737  050026                     JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
2415 041166  012704  050350                     MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2416 041172  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2417 041176  004737  016336                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2418 041202                                     FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
2419 041216  103407                             BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
2420 041220  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2421 041222                                     NEXT.ERRNO
2422 041222                             102$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041222  104455                                                             TRAP    C$ERDF
     041224  001137                                                             .WORD   607
     041226  046773                                                             .WORD   T174SSR
     041230  012046                                                             .WORD   PKTSSR
2423 041232  005237  002222                     INC     FATFLG                  ;SET FATAL ERROR FLAG
2424 041236                             105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041236  104406                                                             TRAP    C$CLP1
2425                                    ;       Do a Write Subsystem WRITE FIFO with byte count equal to 1
2426 041240  012700  000001                     MOV     #1,R0                   ;WRITE 1 BYTE
2427 041244  012701  002312                     MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
2428 041250  004737  050052                     JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
2429 041254  012704  050350                     MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2430 041260  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2431 041264  004737  016336                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2432 041270                                     FORCERROR       107$                    ;@@DFORCE ERROR IF FORCER=1
2433 041304  103407                             BCS     110$                    ;BR IF CARRY SET (GOOD RETURN)
2434 041306  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2435 041310                                     NEXT.ERRNO
2436 041310                             107$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041310  104455                                                             TRAP    C$ERDF
     041312  001140                                                             .WORD   608
     041314  047036                                                             .WORD   T175SSR
     041316  012046                                                             .WORD   PKTSSR
2437 041320  005237  002222                     INC     FATFLG                  ;SET FATAL ERROR FLAG
2438 041324                             110$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041324  104406                                                             TRAP    C$CLP1
2439
2440                                    ;       Do a Write Subsystem READ STATUS
2441 041326  004737  047764                     JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2442 041332  012704  050350                     MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2443 041336  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2444 041342  004737  016336                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2445 041346                                     FORCERROR       112$                    ;@@DFORCE ERROR IF FORCER=1
2446 041362  103407                             BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
2447 041364  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2448 041366                                     NEXT.ERRNO
2449 041366                             112$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041366  104455                                                             TRAP    C$ERDF
     041370  001141                                                             .WORD   609
```

```
                041372  046726                                                          .WORD   T173SSR
                041374  012046                                                          .WORD   PKTSSR
2450            041376  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
2451            041402          120$:   CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
                041402  104406                                                          TRAP    C$CLP1
2452                    ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2453            041404  004737  050146          JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2454            041410  012701  046402          MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2455            041414  012702  050242          MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2456            041420  012221                  MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2457            041422  011211                  MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2458            041424  052711  000020          BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
2459            041430  052711  000040          BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
2460            041434  042711  000200          BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
2461                    ;       If Input Ready NOT=1 then Print Error
2462                    ;       If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
2463            041440  005000                  CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2464            041442  012701  050222          MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2465            041446  012702  046362          MOV     #T17EXP,R2              ;EXPD ADDRESS
2466            041452  012703  000024          MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2467            041456  004737  011500          JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2468            041462                          FORCERROR       132$,NOTSSR     ;a@D
2469            041472  103404                  BCS     140$                    ;BR IF YES
2470            041474                          NEXT.ERRNO
2471            041474          132$:   ERRHRD  ERRNO,T173CMP,MSGSTAT           ;REPORT ERROR
                041474  104456                                                          TRAP    C$ERHRD
                041476  001142                                                          .WORD   610
                041500  047323                                                          .WORD   T173CMP
                041502  012350                                                          .WORD   MSGSTAT
2472            041504          140$:   CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
                041504  104406                                                          TRAP    C$CLP1
2473
2474                    ;       Do Write Subsystem READ FIFO with byte count equal to 1
2475            041506  012700  000001          MOV     #1,R0                   ;SET READ BYTE COUNT
2476            041512  004737  050106          JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
2477            041516  012704  050350          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2478            041522  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2479            041526  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2480            041532                          FORCERROR       142$                    ;a@DFORCE ERROR IF FORCER=1
2481            041546  103407                  BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
2482            041550  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2483            041552                          NEXT.ERRNO
2484            041552          142$:   ERRDF   ERRNO,T176SSR,PKTSSR            ;DEVICE FATAL SSR FAILED TO SET
                041552  104455                                                          TRAP    C$ERDF
                041554  001143                                                          .WORD   611
                041556  047102                                                          .WORD   T176SSR
                041560  012046                                                          .WORD   PKTSSR
2485            041562  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
2486            041566          150$:   CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
                041566  104406                                                          TRAP    C$CLP1
2487                    ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2488            041570  004737  050146          JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2489            041574  012701  046402          MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2490            041600  012702  050242          MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2491            041604  013721  002312          MOV     DATA,(R1)+              ;SET EXPD WORD #8 = COUNT DATA
2492            041610  011211                  MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV (NOT TESTING)
2493                    ;       If Data read from FIFO NOT= to Data sent Then Print Error
```

```
2494
2495 041612  005000                ;       The data is in WORD #8 of the message buffer
                                    CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2496 041614  012701  050222        MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2497 041620  012702  046362        MOV     #T17EXP,R2              ;EXPD ADDRESS
2498 041624  012703  000022        MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
2499 041630  004737  011500        JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2500 041634                        FORCERROR       152$,NOTSSR     ;@@D
2501 041644  103404                BCS     160$                    ;BR IF YES
2502 041646                        NEXT.ERRNO
2503 041646                152$:   ERRHRD  ERRNO,T172CMP,MSGSUB    ;REPORT ERROR
     041646  104456                                                        TRAP    C$ERHRD
     041650  001144                                                        .WORD   612
     041652  047227                                                        .WORD   T172CMP
     041654  013742                                                        .WORD   MSGSUB
2504 041656                        160$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
     041656  104406                                                        TRAP    C$CLP1
2505
2506                               ;       Do a Write Subsystem READ STATUS
2507 041660  004737  047764        JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2508 041664  012704  050350        MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2509 041670  010465  000000        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2510 041674  004737  016336        JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2511 041700                        FORCERROR       162$                    ;@@DFORCE ERROR IF FORCER=1
2512 041714  103407                BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
2513 041716  010001                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2514 041720                        NEXT.ERRNO
2515 041720                162$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041720  104455                                                        TRAP    C$ERDF
     041722  001145                                                        .WORD   613
     041724  046726                                                        .WORD   T173SSR
     041726  012046                                                        .WORD   PKTSSR
2516 041730  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
2517 041734                        170$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
     041734  104406                                                        TRAP    C$CLP1
2518
2519 041736  004737  050146        ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
                                    JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2520 041742  012701  046402        MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2521 041746  012702  050242        MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2522 041752  012221                MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2523 041754  011211                MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2524 041756  052711  000020        BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
2525 041762  042711  000040        BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
2526 041766  042711  000200        BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
2527                               ;       If Input Ready NOT=1 then Print Error
2528                               ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2529 041772  005000                CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2530 041774  012701  050222        MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2531 042000  012702  046362        MOV     #T17EXP,R2              ;EXPD ADDRESS
2532 042004  012703  000024        MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2533 042010  004737  011500        JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2534 042014                        FORCERROR       172$,NOTSSR     ;@@D
2535 042024  103404                BCS     180$                    ;BR IF YES
2536 042026                        NEXT.ERRNO
2537 042026                172$:   ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
     042026  104456                                                        TRAP    C$ERHRD
     042030  001146                                                        .WORD   614
     042032  047407                                                        .WORD   T174CMP
```

```
                042034  012350                                                          .WORD   MSGSTAT
2538            042036                  180$:   CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
                042036  104406                                                          TRAP    C$CLP1
2539            042040                          FORCEXIT        205$                    ;aad
2540            042050  005237  002312          INC     DATA                            ;GET NEXT TEST DATA
2541            042054  023727  002312  000377  CMP     DATA,#377                       ;DONE 0 TO 377?
2542            042062  101002                  BHI     205$                            ;BR IF YES
2543            042064  000137  041156          JMP     100$                            ;DO ANOTHER TEST PATTERN
2544            042070                  205$:
2545
2546            042070                          ENDSUB                                  ;////////////// END SUBTEST //////////
                042070                                                                          L10060:
                042070  104403                                                                  TRAP    C$ESUB
2547
2548            042072  005737  002222          TST     FATFLG                          ;ANY FATAL ERRORS ?
2549            042076  001402                  BEQ     260$                            ;BRANCH IF NOT
2550            042100  004737  017202          JSR     PC,CKDROP                       ;TRY TO DROP THE UNIT
2551            042104                  260$:
2552
```

```
                                .SBTTL  TEST  6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

                        ;++
                        ; TEST 6: SUBTEST 3:
                        ;
                        ; SUBTEST DESCRIPTION:
                        ;
                        ;       This subtest verifies the ability of the FIFO to correctly
                        ;       pass a multiple data bytes from input to output.
                        ;       The following sequence is done with various data patterns
                        ;       and byte counts from 2 to 64.
                        ;               1. Initial FIFO status is checked
                        ;               2. The Write FIFO function.
                        ;               3. Read Status is executed and FIFO status is checked.
                        ;               4. Read FIFO is executed and the data and final status
                        ;                  is checked.
                        ;
                        ; TEST STEPS:
                        ;
                        ; BEGIN
                        ;       Write to TSSR to soft initialize
                        ;       Do a WRITE CHARACTERISTICS to setup a message buffer
                        ;       Do a Write Subsystem READ STATUS
                        ;       If Input Ready  NOT=1   Then Print Error
                        ;       If Output Ready NOT=0   Then Print Error
                        ;       If Data In Miss NOT=0   Then Print Error
                        ;       If Last Word    NOT=0   Then Print Error
                        ;  REPEAT FOR DATA 0 TO 377, 377 TO 0, FLOATING 1'S,0'S AND ALL 1'S/0'S
                        ;  REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
                        ;    BEGIN
                        ;       Do a Write Subsystem WRITE NPR to set tape direction out
                        ;       Do a Write Subsystem WRITE FIFO
                        ;       Do a Write Subsystem READ STATUS
                        ;       If Input Ready  NOT=1   Then Print Error
                        ;       If Output Ready NOT=1   Then Print Error
                        ;       If Data In Miss NOT=0   Then Print Error
                        ;       If Last Word    NOT=0   Then Print Error
                        ;       Do Write Subsystem READ FIFO
                        ;       If Data read from FIFO NOT= to Data sent Then Print Error
                        ;       Do a Write Subsystem READ STATUS
                        ;       If Input Ready  NOT=1   Then Print Error
                        ;       If Output Ready NOT=0   Then Print Error
                        ;       If Data In Miss NOT=0   Then Print Error
                        ;       If Last Word    NOT=0   Then Print Error
                        ;    END
                        ;  END
                        ;--
042104                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
042104                                                                  T6.3:
042104  104402                                                                  TRAP    C$BSUB
                        ;
                        ;       Write to TSSR register to soft initialize the controller
                        5$:
042106  004737  015774          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
042112  103405                  BCS     10$                     ;BR IF SOFT INIT OKAY
042114  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
042116                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
```

```
042116  104455                                                          TRAP    C$ERDF
042120  001146                                                          .WORD   614
042122  003652                                                          .WORD   SFIERR
042124  012034                                                          .WORD   SFIMSG
2609                            ;       Do a WRITE CHARACTERISTICS to setup a message buffer
2610 042126  005037  002222     10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
2611 042132  012704  050200             MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
2612 042136  004737  010662             JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
2613 042142                             FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
2614 042156  103407                     BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
2615 042160  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2616 042162                             NEXT.ERRNO
2617 042162                     42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
042162  104455                                                          TRAP    C$ERDF
042164  001147                                                          .WORD   615
042166  046625                                                          .WORD   T17SSR
042170  012046                                                          .WORD   PKTSSR
2618 042172  005237  002222             INC     FATFLG                  ;SET FATAL ERROR FLAG
2619 042176                     50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
042176  104406                                                          TRAP    C$CLP1
2620                            ;       Do a Write Subsystem READ STATUS
2621 042200  004737  047764             JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2622 042204  012704  050350             MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2623 042210  010465  000000             MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2624 042214  004737  016336             JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2625 042220                             FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
2626 042234  103407                     BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
2627 042236  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2628 042240                             NEXT.ERRNO
2629 042240                     62$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
042240  104455                                                          TRAP    C$ERDF
042242  001150                                                          .WORD   616
042244  046726                                                          .WORD   T173SSR
042246  012046                                                          .WORD   PKTSSR
2630 042250  005237  002222             INC     FATFLG                  ;SET FATAL ERROR FLAG
2631 042254                     70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
042254  104406                                                          TRAP    C$CLP1
2632                            ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2633 042256  004737  050146             JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2634 042262  012701  046402             MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2635 042266  012702  050242             MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2636 042272  012221                     MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2637 042274  011211                     MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2638 042276  052711  000020             BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY=  1
2639 042302  042711  000040             BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY=  0
2640 042306  042711  000200             BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
2641 042312  042711  000100             BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
2642                            ;       If Input Ready NOT=1 then Print Error
2643                            ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2644                            ;       If Last Word    NOT=0   Then Print Error
2645 042316  005000                     CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2646 042320  012701  050222             MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2647 042324  012702  046362             MOV     #T17EXP,R2              ;EXPD ADDRESS
2648 042330  012703  000024             MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2649 042334  004737  011500             JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2650 042340                             FORCERROR       82$,NOTSSR              ;@@D
2651 042350  103404                     BCS     90$                     ;BR IF YES
```

```
                                        NEXT.ERRNO
                                82$:    ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
042352  104456                                                                  TRAP    C$ERHRD
042354  001151                                                                  .WORD   617
042356  047145                                                                  .WORD   T171CMP
042360  012350                                                                  .WORD   MSGSTAT
                                90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
042362  104406                                                                  TRAP    C$CLP1

                                ;  REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
                                ;  TSTFLAG =1 FOR INCREMENT TEST PATTERN
                                ;          =2 FOR DECREMENT TEST PATTERN
                                ;          =3 FOR TSTBLK TABLE PATTERN
042364  012737  000001  002314          MOV     #1,TSTFLAG              ;TEST PATTERN FLAG
                                95$:
042372  012737  000002  002310          MOV     #2,COUNT                ;GET FIRST BYTE COUNT
                                100$:
                                ;       Do a Write Subsystem WRITE NPR to set tape direction out
042400  012700  000100                  MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
042404  004737  050026                  JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
042410  012704  050350                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
042414  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
042420  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
042424                                  FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
042440  103407                          BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
042442  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
042444                                  NEXT.ERRNO
042444                          102$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
042444  104455                                                                  TRAP    C$ERDF
042446  001152                                                                  .WORD   618
042450  046773                                                                  .WORD   T174SSR
042452  012046                                                                  .WORD   PKTSSR
042454  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
                                105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
042460  104406                                                                  TRAP    C$CLP1
                                ;       Do a Write Subsystem WRITE FIFO
042462  004737  050126                  JSR     PC,T17CLEXP             ;CLEAR EXPD BUFFER
042466  012701  046504                  MOV     #T17WFDATA,R1           ;EXPD WRITE FIFO DATA BUFFER
042472  013702  002310                  MOV     COUNT,R2                ;TEST PATTERN SIZE
042476  022737  000001  002314          CMP     #1,TSTFLAG              ;INCREMENT PATTERN THIS TIME THRU?
042504  001005                          BNE     115$                    ;BR IF NO
042506  005000                          CLR     R0                      ;INCREMENT TEST PATTERN
                                110$:   MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
042512  005200                          INC     R0                      ;SET NEXT PATTERN
042514  005302                          DEC     R2                      ;DONE?
042516  003374                          BGT     110$                    ;BR IF NO
042520  022737  000002  002314  115$:   CMP     #2,TSTFLAG              ;DECREMENT PATTERN THIS TIME THRU?
042526  001006                          BNE     125$                    ;BR IF NO
042530  012700  000377                  MOV     #377,R0                 ;DECREMENT TEST PATTERN
                                120$:   MOVB    R0,(R1)+                ;STORE DECREMENT TEST BYTE
042536  005300                          DEC     R0                      ;SET NEXT PATTERN
042540  005302                          DEC     R2                      ;DONE?
042542  003374                          BGT     120$                    ;BR IF NO
042544  022737  000003  002314  125$:   CMP     #3,TSTFLAG              ;TSTBLK PATTERNS THIS TIME THRU?
042552  001005                          BNE     135$                    ;BR IF NO
```

```
2699 042554  012700  002752          MOV     #TSTBLK,R0              ;FLOAT 1'S/0'S ETC. TEST TABLE
2700 042560  112021          130$:  MOVB    (R0)+,(R1)+             ;STORE A TSTBLK BYTE
2701 042562  005302                 DEC     R2                      ;DONE?
2702 042564  003375                 BGT     130$                    ;BR IF NO
2703 042566                  135$:
2704 042566  013700  002310          MOV     COUNT,R0                ;FIFO BYTE COUNT
2705 042572  012701  046504          MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
2706 042576  004737  050052          JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
2707 042602  012704  050350          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2708 042606  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2709 042612  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2710 042616                          FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
2711 042632  103407                  BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
2712 042634  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2713 042636                          NEXT.ERRNO
2714 042636                  142$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042636  104455                                                                  TRAP    C$ERDF
     042640  001153                                                                  .WORD   619
     042642  047036                                                                  .WORD   T175SSR
     042644  012046                                                                  .WORD   PKTSSR
2715 042646  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
2716 042652                  150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042652  104406                                                                  TRAP    C$CLP1
2717
2718                         ;       Do a Write Subsystem READ STATUS
2719 042654  004737  047764          JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2720 042660  012704  050350          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2721 042664  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2722 042670  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2723 042674                          FORCERROR       157$                    ;@@DFORCE ERROR IF FORCER=1
2724 042710  103407                  BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
2725 042712  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2726 042714                          NEXT.ERRNO
2727 042714                  157$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042714  104455                                                                  TRAP    C$ERDF
     042716  001154                                                                  .WORD   620
     042720  046726                                                                  .WORD   T173SSR
     042722  012046                                                                  .WORD   PKTSSR
2728 042724  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
2729 042730                  160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042730  104406                                                                  TRAP    C$CLP1
2730
2731                         ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2732 042732  004737  050146          JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2733 042736  012701  046402          MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2734 042742  012702  050242          MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2735 042746  012221                  MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2736 042750  011211                  MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2737 042752  052711  000020          BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
2738 042756  052711  000040          BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
2739 042762  042711  000200          BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
2740 042766  042711  000100          BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
2741                         ;       If Input Ready NOT=1 then Print Error
2742                         ;       If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
2743 042772  005000                  CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2744 042774  012701  050222          MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2745 043000  012702  046362          MOV     #T17EXP,R2              ;EXPD ADDRESS
```

```
2746 043004  012703  000024            MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2747 043010  004737  011500            JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2748 043014                            FORCERROR       162$,NOTSSR     ;@@D
2749 043024  103404                    BCS     170$                    ;BR IF YES
2750 043026                            NEXT.ERRNO
2751 043026                    162$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
     043026  104456                                                            TRAP    C$ERHRD
     043030  001155                                                            .WORD   621
     043032  047323                                                            .WORD   T173CMP
     043034  012350                                                            .WORD   MSGSTAT
2752 043036                    170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043036  104406                                                            TRAP    C$CLP1
2753
2754                           ;       Do Write Subsystem READ FIFO
2755 043040  013700  002310            MOV     COUNT,R0                ;SET READ BYTE COUNT
2756 043044  004737  050106            JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
2757 043050  012704  050350            MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2758 043054  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2759 043060  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2760 043064                            FORCERROR       172$                    ;@@DFORCE ERROR IF FORCER=1
2761 043100  103407                    BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
2762 043102  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2763 043104                            NEXT.ERRNO
2764 043104                    172$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     043104  104455                                                            TRAP    C$ERDF
     043106  001156                                                            .WORD   622
     043110  047102                                                            .WORD   T176SSR
     043112  012046                                                            .WORD   PKTSSR
2765 043114  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
2766 043120                    180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043120  104406                                                            TRAP    C$CLP1
2767
2768                           ;       If Data read from FIFO NOT= to Data sent Then Print Error
2769 043122  005000                    CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2770 043124  012702  046504            MOV     #T17WFDATA,R2           ;GET EXPECTED ADDRESS FOR CKMSG2
2771 043130  012701  050242            MOV     #T17BFSTA,R1            ;GET RECEIVED ADDRESS FOR CKMSG2
2772 043134  013703  002310            MOV     COUNT,R3                ;NUMBER OF BYTES TO COMPARE
2773 043140  004737  011500            JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2774 043144                            FORCERROR       192$,NOTSSR     ;@@D
2775 043154  103406                    BCS     200$                    ;BR IF YES
2776 043156                            NEXT.ERRNO
2777 043156  013701  002310    192$:   MOV     COUNT,R1                ;GET BYTE COUNT
2778 043162                            ERRHRD  ERRNO,T175CMP,FIFEXP    ;REPORT ERROR
     043162  104456                                                            TRAP    C$ERHRD
     043164  001157                                                            .WORD   623
     043166  047472                                                            .WORD   T175CMP
     043170  012170                                                            .WORD   FIFEXP
2779 043172                    200$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043172  104406                                                            TRAP    C$CLP1
2780
2781                           ;       Do a Write Subsystem READ STATUS
2782 043174  004737  047764            JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2783 043200  012704  050350            MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2784 043204  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2785 043210  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2786 043214                            FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
2787 043230  103407                    BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
```

```
2788 043232  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2789 043234                             NEXT.ERRNO
2790 043234                     212$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     043234  104455                                                              TRAP    C$ERDF
     043236  001160                                                              .WORD   624
     043240  046726                                                              .WORD   T173SSR
     043242  012046                                                              .WORD   PKTSSR
2791 043244  005237  002222             INC     FATFLG                  ;SET FATAL ERROR FLAG
2792 043250                     220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043250  104406                                                              TRAP    C$CLP1
2793                            ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2794 043252  004737  050146             JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2795 043256  012701  046402             MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2796 043262  012702  050242             MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2797 043266  012221                     MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2798 043270  011211                     MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2799 043272  052711  000020             BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
2800 043276  042711  000040             BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
2801 043302  042711  000200             BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
2802 043306  042711  000100             BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
2803                            ;       If Input Ready NOT=1 then Print Error
2804                            ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2805 043312  005000                     CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2806 043314  012701  050222             MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2807 043320  012702  046362             MOV     #T17EXP,R2              ;EXPD ADDRESS
2808 043324  012703  000024             MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2809 043330  004737  011500             JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2810 043334                             FORCERROR       232$,NOTSSR     ;a&D
2811 043344  103404                     BCS     240$                    ;BR IF YES
2812 043346                             NEXT.ERRNO
2813 043346                     232$:   ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
     043346  104456                                                              TRAP    C$ERHRD
     043350  001161                                                              .WORD   625
     043352  047407                                                              .WORD   T174CMP
     043354  012350                                                              .WORD   MSGSTAT
2814 043356                     240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043356  104406                                                              TRAP    C$CLP1
2815 043360                             FORCEXIT        250$            ;a&D
2816 043370  005237  002310             INC     COUNT                   ;GET NEXT BYTE COUNT
2817 043374  023727  002310  000077     CMP     COUNT,#77               ;DONE 0 TO 77
2818 043402  101002                     BHI     250$                    ;BR IF YES
2819 043404  000137  042400             JMP     100$                    ;DO ANOTHER BYTE COUNT
2820 043410  005237  002314     250$:   INC     TSTFLAG                 ;GET NEXT TEST PATTERN CODE
2821 043414  023727  002314  000003     CMP     TSTFLAG,#3              ;DONE INC,DEC,TSTBLK PATTERNS?
2822 043422  101002                     BHI     255$                    ;BR IF YES
2823 043424  000137  042372             JMP     95$                     ;DO ANOTHER TEST PATTERN
2824 043430                     255$:
2825 043430                             ENDSUB                          ;////////////// END SUBTEST //////////
     043430                                                                      L10061:
     043430  104403                                                              TRAP    C$ESUB
2826
2827 043432  005737  002222             TST     FATFLG                  ;ANY FATAL ERRORS ?
2828 043436  001402                     BEQ     260$                    ;BRANCH IF NOT
2829 043440  004737  017202             JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2830 043444                     260$:
2831
2832
```

```
2833
```

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

                        ;++
                        ; TEST 6: SUBTEST 4:
                        ;
                        ; SUBTEST DESCRIPTION:
                        ;
                        ;       This subtest verifies that reading the FIFO when it is
                        ;       empty causes the Last Word (ILW) status to assert.
                        ;
                        ; TEST STEPS:
                        ;
                        ; BEGIN
                        ;       Write to TSSR to soft initialize
                        ;       Do Write Subsystem READ FIFO with byte count equal to 1
                        ;       Do a Write Subsystem READ STATUS
                        ;       If Input Ready  NOT=1   Then Print Error
                        ;       If Output Ready NOT=0   Then Print Error
                        ;       If Data In Miss NOT=0   Then Print Error
                        ;       If Last Word (ILW) NOT=1 Then Print Error
                        ;    END
                        ;--
043444                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
043444                                                                  T6.4:
043444  104402                                                                  TRAP    C$BSUB

                        ;       Write to TSSR register to soft initialize the controller
043446                  5$:
043446  004737  015774          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
043452  103405                  BCS     10$                     ;BR IF SOFT INIT OKAY
043454  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043456                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
043456  104455                                                                  TRAP    C$ERDF
043460  001161                                                                  .WORD   625
043462  003652                                                                  .WORD   SFIERR
043464  012034                                                                  .WORD   SFIMSG

                        ;       Do a WRITE CHARACTERISTICS to setup a message buffer
043466  005037  002222  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
043472  012704  050200          MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
043476  004737  010662          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
043502                          FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
043516  103407                  BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
043520  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043522                          NEXT.ERRNO
043522                  42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
043522  104455                                                                  TRAP    C$ERDF
043524  001162                                                                  .WORD   626
043526  046625                                                                  .WORD   T17SSR
043530  012046                                                                  .WORD   PKTSSR
043532  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
043536                  50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
043536  104406                                                                  TRAP    C$CLP1

                        ;       Do Write Subsystem READ FIFO with byte count equal to 1
043540  012700  000001          MOV     #1,R0                   ;SET READ BYTE COUNT
043544  004737  050106          JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
043550  012704  050350          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
```

```
 2881 043554  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 2882 043560  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 2883 043564                                 FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
 2884 043600  103407                         BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
 2885 043602  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2886 043604                                 NEXT.ERRNO
 2887 043604                         142$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      043604  104455                                                                    TRAP    C$ERDF
      043606  001163                                                                    .WORD   627
      043610  047102                                                                    .WORD   T176SSR
      043612  012046                                                                    .WORD   PKTSSR
 2888 043614  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 2889 043620                         150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      043620  104406                                                                    TRAP    C$CLP1
 2890
 2891                                ;       Do a Write Subsystem READ STATUS
 2892 043622  004737  047764                 JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
 2893 043626  012704  050350                 MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 2894 043632  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 2895 043636  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 2896 043642                                 FORCERROR       162$                    ;@@DFORCE ERROR IF FORCER=1
 2897 043656  103407                         BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
 2898 043660  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2899 043662                                 NEXT.ERRNO
 2900 043662                         162$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      043662  104455                                                                    TRAP    C$ERDF
      043664  001164                                                                    .WORD   628
      043666  046726                                                                    .WORD   T173SSR
      043670  012046                                                                    .WORD   PKTSSR
 2901 043672  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 2902 043676                         170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      043676  104406                                                                    TRAP    C$CLP1
 2903                                ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
 2904 043700  004737  050146                 JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
 2905 043704  012701  046402                 MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
 2906 043710  012702  050242                 MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
 2907 043714  012221                         MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
 2908 043716  011211                         MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
 2909 043720  052711  000020                 BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
 2910 043724  042711  000040                 BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
 2911 043730  042711  000200                 BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
 2912 043734  052711  000100                 BIS     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=1
 2913                                ;       If Input Ready NOT=1 then Print Error
 2914                                ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
 2915                                ;       If Last Word (ILW) NOT=1 Then Print Error
 2916 043740  005000                         CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 2917 043742  012701  050222                 MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 2918 043746  012702  046362                 MOV     #T17EXP,R2              ;EXPD ADDRESS
 2919 043752  012703  000024                 MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 2920 043756  004737  011500                 JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 2921 043762                                 FORCERROR       172$,NOTSSR             ;@@D
 2922 043772  103404                         BCS     180$                    ;BR IF YES
 2923 043774                                 NEXT.ERRNO
 2924 043774                         172$:   ERRHRD  ERRNO,T176CMP,MSGSTAT   ;REPORT ERROR
      043774  104456                                                                    TRAP    C$ERHRD
      043776  001165                                                                    .WORD   629
      044000  047546                                                                    .WORD   T176CMP
```

```
            044002  012350                                                          .WORD   MSGSTAT
2925        044004                      180$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
            044004  104406                                                          TRAP    C$CLP1
2926
2927        044006                      ENDSUB                          ;////////////// END SUBTEST //////////
            044006                                                                L10062:
            044006  104403                                                          TRAP    C$ESUB
2928
2929        044010  C05737  002222              TST     FATFLG          ;ANY FATAL ERRORS ?
2930        044014  001402              BEQ     260$                    ;BRANCH IF NOT
2931        044016  004737  017202              JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
2932        044022                      260$:
2933
2934
```

```
                                .SBTTL  TEST  6: SUBTEST 5: FIFO Verify Input Ready

                        ;++
                        ; TEST 6: SUBTEST 5:
                        ;
                        ; SUBTEST DESCRIPTION:
                        ;
                        ;       This subtest verifies that writing 64. bytes into the FIFO
                        ;       without reading any out causes the Input Ready status to
                        ;       negate. The Subtest then verifies that writing a 65th byte
                        ;       into the FIFO causes the Data In Miss status to assert.
                        ;       Next it is verified that the original 64 bytes can be read
                        ;       out correctly and that the data has not been corrupted.
                        ;
                        ; TEST STEPS:
                        ;
                        ; BEGIN
                        ;       Write to TSSR to soft initialize
                        ;       Do a WRITE CHARACTERISTICS to setup a message buffer
                        ;       Do a Write Subsystem WRITE NPR to set tape direction out
                        ;       Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
                        ;       Do a Write Subsystem READ STATUS
                        ;       If Input Ready  NOT=0   Then Print Error
                        ;       If Output Ready NOT=1   Then Print Error
                        ;       If Data In Miss NOT=0   Then Print Error
                        ;       Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
                        ;       Do a Write Subsystem READ STATUS
                        ;       If Input Ready  NOT=0   Then Print Error
                        ;       If Output Ready NOT=1   Then Print Error
                        ;       If Data In Miss NOT=1   Then Print Error
                        ;       Do Write Subsystem READ FIFO
                        ;       If Data read from FIFO NOT= to Data sent Then Print Error
                        ;       Do a Write Subsystem READ STATUS
                        ;       If Input Ready  NOT=1   Then Print Error
                        ;       If Output Ready NOT=0   Then Print Error
                        ;       If Data In Miss NOT=1   Then Print Error
                        ;  END
                        ;--
044022                          BGNSUB                          ;////////////// BEGIN SUBTEST ///////////
044022                                                                  T6.5:
044022  104402                                                                  TRAP    C$BSUB

                        ;
                        ;       Write to TSSR register to soft initialize the controller
                        ;5$:
044024  004737  015774          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
044030  103405                  BCS     10$                     ;BR IF SOFT INIT OKAY
044032  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
044034                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
044034  104455                                                                  TRAP    C$ERDF
044036  001165                                                                  .WORD   629
044040  003652                                                                  .WORD   SFIERR
044042  012034                                                                  .WORD   SFIMSG

                        ;
                        ;       Do a WRITE CHARACTERISTICS to setup a message buffer
                        ;10$:
044044  005037  002222          CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
044050  012704  050200          MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
044054  004737  010662          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
044060                          FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
```

```
 2987 044074  103407                    BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
 2988 044076  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2989 044100                            NEXT.ERRNO
 2990 044100                    42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      044100  104455                                                            TRAP    C$ERDF
      044102  001166                                                            .WORD   630
      044104  046625                                                            .WORD   T17SSR
      044106  012046                                                            .WORD   PKTSSR
 2991 044110  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
 2992 044114                    50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      044114  104406                                                            TRAP    C$CLP1
 2993
 2994                           ;       Do a Write Subsystem WRITE NPR to set tape direction out
 2995 044116  012700  000100    100$:   MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
 2996 044122  004737  050026            JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
 2997 044126  012704  050350            MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 2998 044132  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 2999 044136  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 3000 044142                            FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
 3001 044156  103407                    BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
 3002 044160  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 3003 044162                            NEXT.ERRNO
 3004 044162                    102$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      044162  104455                                                            TRAP    C$ERDF
      044164  001167                                                            .WORD   631
      044166  046773                                                            .WORD   T174SSR
      044170  012046                                                            .WORD   PKTSSR
 3005 044172  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
 3006 044176                    105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      044176  104406                                                            TRAP    C$CLP1
 3007
 3008                           ;       Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
 3009 044200  012737  000100  002310    MOV     #64.,COUNT              ;WRITE 64 BYTES
 3010 044206  012701  046504            MOV     #T17WFDATA,R1           ;EXPD WRITE FIFO DATA BUFFER
 3011 044212  012702  000100            MOV     #64.,R2                 ;TEST PATTERN SIZE
 3012 044216  005000                    CLR     R0                      ;INCREMENT TEST PATTERN
 3013 044220  110021            110$:   MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
 3014 044222  005200                    INC     R0                      ;SET NEXT PATTERN
 3015 044224  005302                    DEC     R2                      ;DONE?
 3016 044226  003374                    BGT     110$                    ;BR IF NO
 3017 044230  013700  002310            MOV     COUNT,R0                ;FIFO BYTE COUNT
 3018 044234  012701  046504            MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
 3019 044240  004737  050052            JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
 3020 044244  012704  050350            MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 3021 044250  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 3022 044254  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 3023 044260                            FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
 3024 044274  103407                    BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
 3025 044276  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 3026 044300                            NEXT.ERRNO
 3027 044300                    142$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      044300  104455                                                            TRAP    C$ERDF
      044302  001170                                                            .WORD   632
      044304  047036                                                            .WORD   T175SSR
      044306  012046                                                            .WORD   PKTSSR
 3028 044310  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
 3029 044314                    150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

```
      044314  104406                                                    TRAP    C$CLP1
3030
3031                                  ;       Do a Write Subsystem READ STATUS
3032                                  ;       If Input Ready  NOT=0   Then Print Error
3033                                  ;       If Output Ready NOT=1   Then Print Error
3034                                  ;       If Data In Miss NOT=0   Then Print Error
3035 044316  004737  047764                   JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
3036 044322  012704  050350                   MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3037 044326  010465  000000                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3038 044332  004737  016336                   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3039 044336                                   FORCERROR       157$                    ;@@DFORCE ERROR IF FORCER=1
3040 044352  103407                          BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
3041 044354  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3042 044356                                   NEXT.ERRNO
3043 044356                           157$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      044356  104455                                                            TRAP    C$ERDF
      044360  001171                                                            .WORD   633
      044362  046726                                                            .WORD   T173SSR
      044364  012046                                                            .WORD   PKTSSR
3044 044366  005237  002222                   INC     FATFLG                  ;SET FATAL ERROR FLAG
3045 044372                           160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      044372  104406                                                            TRAP    C$CLP1
3046                                  ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
3047 044374  004737  050146                   JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
3048 044400  012701  046402                   MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
3049 044404  012702  050242                   MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
3050 044410  012221                           MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
3051 044412  011211                           MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
3052 044414  042711  000020                   BIC     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 0
3053 044420  052711  000040                   BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
3054 044424  042711  000200                   BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
3055 044430  005000                           CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
3056 044432  012701  050222                   MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
3057 044436  012702  046362                   MOV     #T17EXP,R2              ;EXPD ADDRESS
3058 044442  012703  000024                   MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
3059 044446  004737  011500                   JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3060 044452                                   FORCERROR       162$,NOTSSR     ;@@D
3061 044462  103404                          BCS     170$                    ;BR IF YES
3062 044464                                   NEXT.ERRNO
3063 044464                           162$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
      044464  104456                                                            TRAP    C$ERHRD
      044466  001172                                                            .WORD   634
      044470  047323                                                            .WORD   T173CMP
      044472  012350                                                            .WORD   MSGSTAT
3064 044474                           170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      044474  104406                                                            TRAP    C$CLP1
3065
3066
3067                                  ;       Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
3068 044476  012700  000001                   MOV     #1,R0                   ;FIFO BYTE COUNT
3069 044502  012701  046504                   MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
3070 044506  004737  050052                   JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
3071 044512  012704  050350                   MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3072 044516  010465  000000                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3073 044522  004737  016336                   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3074 044526                                   FORCERROR       172$                    ;@@DFORCE ERROR IF FORCER=1
3075 044542  103407                          BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
```

```
3076 044544  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3077 044546                            NEXT.ERRNO
3078 044546                    172$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044546  104455                                                            TRAP    C$ERDF
     044550  001173                                                            .WORD   635
     044552  047036                                                            .WORD   T175SSR
     044554  012046                                                            .WORD   PKTSSR
3079 044556  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
3080 044562                    180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044562  104406                                                            TRAP    C$CLP1
3081
3082                           ;       Do a Write Subsystem READ STATUS
3083                           ;       If Input Ready  NOT=0   Then Print Error
3084                           ;       If Output Ready NOT=1   Then Print Error
3085                           ;       If Data In Miss NOT=1   Then Print Error
3086 044564  004737  047764            JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
3087 044570  012704  050350            MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3088 044574  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3089 044600  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3090 044604                            FORCERROR       187$                    ;@@DFORCE ERROR I  FORCER=1
3091 044620  103407                    BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
3092 044622  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3093 044624                            NEXT.ERRNO
3094 044624                    187$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044624  104455                                                            TRAP    C$ERDF
     044626  001174                                                            .WORD   636
     044630  046726                                                            .WORD   T173SSR
     044632  012046                                                            .WORD   PKTSSR
3095 044634  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
3096 044640                    190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044640  104406                                                            TRAP    C$CLP1
3097
3098 044642  004737  050146    ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
                                       JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
3099 044646  012701  046402            MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
3100 044652  012702  050242            MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
3101 044656  012221                    MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
3102 044660  011211                    MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
3103 044662  042711  000020            BIC     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 0
3104 044666  052711  000040            BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
3105 044672  052711  000200            BIS     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 1
3106 044676  005000                    CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
3107 044700  012701  050222            MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
3108 044704  012702  046362            MOV     #T17EXP,R2              ;EXPD ADDRESS
3109 044710  012703  000024            MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
3110 044714  004737  011500            JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3111 044720                            FORCERROR       192$,NOTSSR     ;@@D
3112 044730  103404                    BCS     200$                    ;BR IF YES
3113 044732                            NEXT.ERRNO
3114 044732                    192$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
     044732  104456                                                            TRAP    C$ERHRD
     044734  001175                                                            .WORD   637
     044736  047323                                                            .WORD   T173CMP
     044740  012350                                                            .WORD   MSGSTAT
3115 044742                    200$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044742  104406                                                            TRAP    C$CLP1
3116
3117 044744  013700  002310    ;       Do Write Subsystem READ FIFO
                                       MOV     COUNT,R0                ;SET READ BYTE COUNT
```

```
3118 044750  004737  050106          JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
3119 044754  012704  050350          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3120 044760  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3121 044764  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3122 044770                          FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
3123 045004  103407                  BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
3124 045006  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3125 045010                          NEXT.ERRNO
3126 045010                  212$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045010  104455                                                              TRAP    C$ERDF
     045012  001176                                                              .WORD   638
     045014  047102                                                              .WORD   T176SSR
     045016  012046                                                              .WORD   PKTSSR
3127 045020  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
3128 045024                  220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045024  104406                                                              TRAP    C$CLP1
3129
3130                         ;       If Data read from FIFO NOT= to Data sent Then Print Error
3131 045026  005000                  CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
3132 045030  012702  046504          MOV     #T17WFDATA,R2           ;GET EXPECTED ADDRESS FOR CKMSG2
3133 045034  012701  050242          MOV     #T17BFSTA,R1            ;GET RECEIVED ADDRESS FOR CKMSG2
3134 045040  013703  002310          MOV     COUNT,R3                ;NUMBER OF BYTES TO COMPARE
3135 045044  004737  011500          JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3136 045050                          FORCERROR       232$,NOTSSR             ;@@D
3137 045060  103406                  BCS     240$                    ;BR IF YES
3138 045062                          NEXT.ERRNO
3139 045062  013701  002310  232$:   MOV     COUNT,R1                ;GET BYTE COUNT
3140 045066                          ERRHRD  ERRNO,T175CMP,FIFEXP    ;REPORT ERROR
     045066  104456                                                              TRAP    C$ERHRD
     045070  001177                                                              .WORD   639
     045072  047472                                                              .WORD   T175CMP
     045074  012170                                                              .WORD   FIFEXP
3141 045076                  240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045076  104406                                                              TRAP    C$CLP1
3142
3143                         ;       Do a Write Subsystem READ STATUS
3144                         ;       If Input Ready  NOT=1   Then Print Error
3145                         ;       If Output Ready NOT=0   Then Print Error
3146                         ;       If Data In Miss NOT=1   Then Print Error
3147 045100  004737  047764          JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
3148 045104  012704  050350          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3149 045110  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3150 045114  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3151 045120                          FORCERROR       252$                    ;@@DFORCE ERROR IF FORCER=1
3152 045134  103407                  BCS     260$                    ;BR IF CARRY SET (GOOD RETURN)
3153 045136  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3154 045140                          NEXT.ERRNO
3155 045140                  252$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045140  104455                                                              TRAP    C$ERDF
     045142  001200                                                              .WORD   640
     045144  046726                                                              .WORD   T173SSR
     045146  012046                                                              .WORD   PKTSSR
3156 045150  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
3157 045154                  260$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045154  104406                                                              TRAP    C$CLP1
3158                         ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
3159 045156  004737  050146          JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
```

```
3160 045162  012701  046402          MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
3161 045166  012702  050242          MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
3162 045172  012221                  MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
3163 045174  011211                  MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
3164 045176  052711  000020          BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
3165 045202  042711  000040          BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
3166 045206  052711  000200          BIS     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 1
3167 045212  005000                  CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
3168 045214  012701  050222          MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
3169 045220  012702  046362          MOV     #T17EXP,R2              ;EXPD ADDRESS
3170 045224  012703  000024          MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
3171 045230  004737  011500          JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3172 045234                          FORCERROR       272$,NOTSSR     ;@@D
3173 045244  103404                  BCS     280$                    ;BR IF YES
3174 045246                          NEXT.ERRNO
3175 045246                  272$:   ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
     045246  104456                                                          TRAP    C$ERHRD
     045250  001201                                                          .WORD   641
     045252  047407                                                          .WORD   T174CMP
     045254  012350                                                          .WORD   MSGSTAT
3176 045256                  280$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045256  104406                                                          TRAP    C$CLP1
3177
3178 045260                          ENDSUB                          ;////////////// END SUBTEST //////////
     045260                                                                  L10063:
     045260  104403                                                          TRAP    C$ESUB
3179
3180 045262  005737  002222          TST     FATFLG                  ;ANY FATAL ERRORS ?
3181 045266  001402                  BEQ     300$                    ;BRANCH IF NOT
3182 045270  004737  017202          JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
3183 045274                  300$:
3184
3185
3186
```

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

                ;++
                ; TEST 6: SUBTEST 6:
                ;
                ; SUBTEST DESCRIPTION:
                ;
                ;       This subtest verifies that the Reset FIFO function within
                ;       the Write Miscellaneous Control 1 function initializes
                ;       the FIFO to correct intial status. The following steps
                ;       are performed:
                ;               1. Reset an already initialized FIFO and check for
                ;                  proper status.
                ;               2. Write a varying number of bytes (1-65.) into the
                ;                  FIFO and verify that after each block of bytes is
                ;                  written the FIFO can be be reset to it's initial
                ;                  state.
                ;
                ; TEST STEPS:
                ;
                ; BEGIN
                ;       Write to TSSR to soft initialize
                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
                ;       Do a Write Subsystem Write Misc to Reset FIFO
                ;       Do a Write Subsystem READ STATUS
                ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
                ;         signals NOT=0 Then Print Error
                ;       Do a Write Subsystem WRITE NPR to set tape direction out
                ;
                ; REPEAT FOR BYTE COUNT 1 TO 65.
                ;   BEGIN
                ;       Do a Write Subsystem WRITE FIFO with the current byte count
                ;       Do a Write Subsystem Write Misc to Reset FIFO
                ;       Do a Write Subsystem READ STATUS
                ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
                ;         signals NOT=0 Then Print Error
                ;   END
                ;--
045274                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
045274                                                                          T6.6:
045274  104402                                                                  TRAP    C$BSUB

                ;       Write to TSSR register to soft initialize the controller
045276          5$:
045276  004737  015774  JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
045302  103405          BCS     10$                     ;BR IF SOFT INIT OKAY
045304  010001          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
045306                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
045306  104455                                                                  TRAP    C$ERDF
045310  001201                                                                  .WORD   641
045312  003652                                                                  .WORD   SFIERR
045314  012034                                                                  .WORD   SFIMSG

                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
045316  005037  002222  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
045322  012704  050200          MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
045326  004737  010662          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
045332                          FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
```

```
3239 045346  103407                    BCS     5$                      ;BR IF CARRY SET (GOOD RETURN)
3240 045350  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3241 045352                            NEXT.ERRNO
3242 045352                    42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     045352  104455                                                            TRAP    C$ERDF
     045354  001202                                                            .WORD   642
     045356  046625                                                            .WORD   T17SSR
     045360  012046                                                            .WORD   PKTSSR
3243 045362  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
3244 045366                    50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045366  104406                                                            TRAP    C$CLP1
3245                           ;       Do a Write Subsystem Write Misc to Reset FIFO
3246 045370  004737  050004            JSR     PC,T17RSFIF             ;SETUP PKT FOR WRITE MISC RESET FIFO
3247 045374  012704  050350            MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3248 045400  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3249 045404  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3250 045410                            FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
3251 045424  103407                    BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
3252 045426  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3253 045430                            NEXT.ERRNO
3254 045430                    62$:    ERRDF   ERRNO,T172SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045430  104455                                                            TRAP    C$ERDF
     045432  001203                                                            .WORD   643
     045434  046662                                                            .WORD   T172SSR
     045436  012046                                                            .WORD   PKTSSR
3255 045440  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
3256 045444                    70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045444  104406                                                            TRAP    C$CLP1
3257
3258                           ;       Do a Write Subsystem READ STATUS
3259                           ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
3260                           ;         signals NOT=0 Then Print Error
3261 045446  004737  047764            JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
3262 045452  012704  050350            MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3263 045456  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3264 045462  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3265 045466                            FORCERROR       77$                     ;@@DFORCE ERROR IF FORCER=1
3266 045502  103407                    BCS     80$                     ;BR IF CARRY SET (GOOD RETURN)
3267 045504  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3268 045506                            NEXT.ERRNO
3269 045506                    77$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045506  104455                                                            TRAP    C$ERDF
     045510  001204                                                            .WORD   644
     045512  046726                                                            .WORD   T173SSR
     045514  012046                                                            .WORD   PKTSSR
3270 045516  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
3271 045522                    80$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045522  104406                                                            TRAP    C$CLP1
3272 045524  004737  050146            JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
3273 045530  012701  046402            MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
3274 045534  012702  050242            MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
3275 045540  011211                    MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
3276 045542  042711  002000            BIC     #S1.ICER,(R1)           ;SET EXPD ICER  =0
3277 045546  042711  001000            BIC     #S1.IFMK,(R1)           ;SET EXPD IFMK  =0
3278 045552  042711  000400            BIC     #S1.IHER,(R1)           ;SET EXPD IHER  =0
3279 045556  016261  000002  000002    MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
3280 045564  005000                    CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
```

```
3281 045566  012701  050222          MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
3282 045572  012702  046362          MOV     #T17EXP,R2              ;EXPD ADDRESS
3283 045576  012703  000024          MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
3284 045602  004737  011500          JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3285 045606                          FORCERROR       92$,NOTSSR      ;@@D
3286 045616  103404                  BCS     100$                    ;BR IF YES
3287 045620                          NEXT.ERRNO
3288 045620                  92$:    ERRHRD  ERRNO,T177CMP,MSGSTAT   ;REPORT ERROR
     045620  104456                                                          TRAP    C$ERHRD
     045622  001205                                                          .WORD   645
     045624  047654                                                          .WORD   T177CMP
     045626  012350                                                          .WORD   MSGSTAT
3289 045630                  100$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045630  104406                                                          TRAP    C$CLP1
3290
3291                         ;       Do a Write Subsystem WRITE NPR to set tape direction out
3292 045632  012700  000100          MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
3293 045636  004737  050026          JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
3294 045642  012704  050350          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3295 045646  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3296 045652  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3297 045656                          FORCERROR       112$                    ;@@DFORCE ERROR IF FORCER=1
3298 045672  103407                  BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
3299 045674  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3300 045676                          NEXT.ERRNO
3301 045676                  112$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045676  104455                                                          TRAP    C$ERDF
     045700  001206                                                          .WORD   646
     045702  046773                                                          .WORD   T174SSR
     045704  012046                                                          .WORD   PKTSSR
3302 045706  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
3303 045712                  120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045712  104406                                                          TRAP    C$CLP1
3304
3305                         ;       Setup incrementing pattern in FIFO data buffer
3306 045714  012701  046402          MOV     #T17EXSTA,R1            ;EXPD WRITE FIFO DATA BUFFER
3307 045720  012702  000100          MOV     #64.,R2                 ;TEST PATTERN SIZE
3308 045724  005000                  CLR     R0                      ;INCREMENT TEST PATTERN
3309 045726  110021          130$:   MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
3310 045730  005200                  INC     R0                      ;SET NEXT PATTERN
3311 045732  005302                  DEC     R2                      ;DONE?
3312 045734  003374                  BGT     130$                    ;BR IF NO
3313
3314                         ;  REPEAT FOR BYTE COUNT 1 TO 65.
3315 045736  012737  000001  002310      MOV     #1,COUNT                ;GET FIRST BYTE COUNT
3316                         ;       Do a Write Subsystem WRITE FIFO with the current byte count
3317 045744                  150$:                                   ;REPEAT LOOP LABEL
3318 045744  013700  002310          MOV     COUNT,R0                ;FIFO BYTE COUNT
3319 045750  012701  046402          MOV     #T17EXSTA,R1            ;FIFO WRITE DATA ADDRESS
3320 045754  004737  050052          JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
3321 045760  012704  050350          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3322 045764  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3323 045770  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3324 045774                          FORCERROR       152$                    ;@@DFORCE ERROR IF FORCER=1
3325 046010  103407                  BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
3326 046012  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3327 046014                          NEXT.ERRNO
```

```
3328 046014                          152$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     046014  104455                                                                TRAP    C$ERDF
     046016  001207                                                                .WORD   647
     046020  047036                                                                .WORD   T175SSR
     046022  012046                                                                .WORD   PKTSSR
3329 046024  005237  002222                  INC     FATFLG                 ;SET FATAL ERROR FLAG
3330 046030                          160$:   CKLOOP                         ;LOOP ON ERROR, IF FLAG SET
     046030  104406                                                                TRAP    C$CLP1
3331
3332                                 ;       Do a Write Subsystem Write Misc to Reset FIFO
3333 046032  004737  050004                  JSR     PC,T17RSFIF            ;SETUP PKT FOR WRITE MISC RESET FIFO
3334 046036  012704  050350                  MOV     #T17PK2,R4             ;GET WRITE SUBSYSTEM COMMAND PACKET
3335 046042  010465  000000                  MOV     R4,TSDB(R5)            ;SET THE PACKET ADDRESS TO EXECUTE
3336 046046  004737  016336                  JSR     PC,CHKTSSR             ;WAIT FOR SSR TO SET
3337 046052                                  FORCERROR       162$                    ;@@DFORCE ERROR IF FORCER=1
3338 046066  103407                          BCS     170$                   ;BR IF CARRY SET (GOOD RETURN)
3339 046070  010001                          MOV     R0,R1                  ;SAVE CONTENTS OF TSSR
3340 046072                                  NEXT.ERRNO
3341 046072                          162$:   ERRDF   ERRNO,T172SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     046072  104455                                                                TRAP    C$ERDF
     046074  001210                                                                .WORD   648
     046076  046662                                                                .WORD   T172SSR
     046100  012046                                                                .WORD   PKTSSR
3342 046102  005237  002222                  INC     FATFLG                 ;SET FATAL ERROR FLAG
3343 046106                          170$:   CKLOOP                         ;LOOP ON ERROR, IF FLAG SET
     046106  104406                                                                TRAP    C$CLP1
3344
3345                                 ;       Do a Write Subsystem READ STATUS
3346                                 ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
3347                                 ;         signals NOT=0 Then Print Error
3348 046110  004737  047764                  JSR     PC,T17SRD              ;SETUP PACKET FOR READ STATUS
3349 046114  012704  050350                  MOV     #T17PK2,R4             ;GET WRITE SUBSYSTEM COMMAND PACKET
3350 046120  010465  000000                  MOV     R4,TSDB(R5)            ;SET THE PACKET ADDRESS TO EXECUTE
3351 046124  004737  016336                  JSR     PC,CHKTSSR             ;WAIT FOR SSR TO SET
3352 046130                                  FORCERROR       177$                    ;@@DFORCE ERROR IF FORCER=1
3353 046144  103407                          BCS     180$                   ;BR IF CARRY SET (GOOD RETURN)
3354 046146  010001                          MOV     R0,R1                  ;SAVE CONTENTS OF TSSR
3355 046150                                  NEXT.ERRNO
3356 046150                          177$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     046150  104455                                                                TRAP    C$ERDF
     046152  001211                                                                .WORD   649
     046154  046726                                                                .WORD   T173SSR
     046156  012046                                                                .WORD   PKTSSR
3357 046160  005237  002222                  INC     FATFLG                 ;SET FATAL ERROR FLAG
3358 046164                          180$:   CKLOOP                         ;LOOP ON ERROR, IF FLAG SET
     046164  104406                                                                TRAP    C$CLP1
3359 046166  004737  050146                  JSR     PC,T17SETEXP           ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
3360 046172  012701  046402                  MOV     #T17EXSTA,R1           ;GET EXPECTED READ STATUS
3361 046176  012702  050242                  MOV     #T17BFSTA,R2           ;GET RECV READ STATUS
3362 046202  011211                          MOV     (R2),(R1)              ;SET EXPD WORD #8 = RECV TEMP
3363 046204  042711  002000                  BIC     #S1.ICER,(R1)          ;SET EXPD ICER  =0
3364 046210  042711  001000                  BIC     #S1.IFMK,(R1)          ;SET EXPD IFMK  =0
3365 046214  042711  000400                  BIC     #S1.IHER,(R1)          ;SET EXPD IHER  =0
3366 046220  016261  000002  000002          MOV     2(R2),2(R1)            ;SET EXPD WORD #9 = RECV (NOT TESTING)
3367 046226  005000                          CLR     R0                     ;HIGH RECV ADDRESS FOR CKMSG2
3368 046230  012701  050222                  MOV     #T17BFR,R1             ;LOW RECV ADDRESS FOR CKMSG2
3369 046234  012702  046362                  MOV     #T17EXP,R2             ;EXPD ADDRESS
```

```
3370 046240  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
3371 046244  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3372 046250                                FORCERROR       192$,NOTSSR     ;@@D
3373 046260  103404                        BCS     200$                    ;BR IF YES
3374 046262                                NEXT.ERRNO
3375 046262                        192$:   ERRHRD  ERRNO,T177CMP,MSGSTAT   ;REPORT ERROR
     046262  104456                                                            TRAP    C$ERHRD
     046264  001212                                                            .WORD   650
     046266  047654                                                            .WORD   T177CMP
     046270  012350                                                            .WORD   MSGSTAT
3376 046272                        200$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     046272  104406                                                            TRAP    C$CLP1
3377
3378
3379 046274                        250$:
3380 046274                                FORCEXIT        260$
3381 046304  005237  002310                INC     COUNT                   ;GET NEXT BYTE COUNT
3382 046310  023727  002310  000101        CMP     COUNT,#65.              ;DONE ALL BYTES?
3383 046316  101002                        BHI     260$                    ;BR IF YES
3384 046320  000137  045744                JMP     150$                    ;DO ANOTHER BYTE COUNT
3385 046324                        260$:
3386
3387 046324                                ENDSUB                          ;////////////// END SUBTEST //////////
     046324                                                                 L10064:
     046324  104403                                                            TRAP    C$ESUB
3388
3389 046326  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
3390 046332  001402                        BEQ     300$                    ;BRANCH IF NOT
3391 046334  004737  017202                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
3392 046340  004737  016456        300$:   JSR     PC,TSTLOOP              ;DO ITERATIONS?
3393 046344  103002                        BCC     305$                    ;BR IF NO
3394 046346  000137  040404                JMP     T17LOOP                 ;LOOP UNTIL ITERATIONS DONE
3395 046352                        305$:
3396
3397 046352                                EXIT    TST                     ;////////// EXIT TEST //////////
     046352  104432                                                            TRAP    C$EXIT
     046354  002112                                                            .WORD   L10056-.
3398
3399
```

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


046356                                  T17MSK:                         ;MASK OF UNTESTED BITS IN READ STATUS BYTES
                                                                        ;UNTESTED BITS ARE SET TO 1
046356      377                                 .BYTE   ^C<000>         ;BYTE 0 MASK
046357      037                                 .BYTE   ^C<340>         ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
046360      360                                 .BYTE   ^C<017>         ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
046361      000                                 .BYTE   0               ;MAKE IT EVEN

046362                                  T17EXP:                         ;BEGIN EXPECTED DATA BUFFER
046362  000000                                  .WORD   0               ;MESSAGE TYPE
046364  000000                                  .WORD   0               ;DATA FIELD LENGTH
046366  000000                                  .WORD   0               ;RBPCR
046370  000000                                  .WORD   0               ;XST0
046372  000000                                  .WORD   0               ;XST1
046374  000000                                  .WORD   0               ;XST2
046376  000000                                  .WORD   0               ;XST3
046400  000000                                  .WORD   0               ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
046402                                  T17EXSTA: .BLKB 66.             ;EXPECTED READ STATUS AND WRITE FIFO DATA
046504                                  T17EXEND:                               ;END EXPECTED DATA BUFFER

046504                                  T17WFDATA: .BLKB 66.            ;WRITE FIFO EXPECTED DATA BUFFER


                                        ;+
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-

046606      106     111     106     TST17ID:        .ASCIZ  'FIFO Exerciser'
046625      127     122     111     T17SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
046662      127     122     111     T172SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
046726      127     122     111     T173SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
046773      127     122     111     T174SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
047036      127     122     111     T175SSR:.ASCIZ  'WRITE SUBSYSTEM (Write FIFO) Failed'
047102      127     122     111     T176SSR:.ASCIZ  'WRITE SUBSYSTEM (Read FIFO) Failed'
047145      106     111     106     T171CMP:.ASCIZ  'FIFO Status in WORD #9 Incorrect after Initialize'
047227      122     145     141     T172CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
047323      106     111     106     T173CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after WRITE FIFO'
047407      106     111     106     T174CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after READ FIFO'
047472      122     145     141     T175CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO Data'
047546      106     111     106     T176CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after READ FIFO from an Empty FIFO'
047654      106     111     106     T177CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after RESET FIFO'
                                                .EVEN


                                        ;+
                                        ; CLEAR MESSAGE BUFFER
                                        ;-
047740                                  T17CLRBUF:
047740                                          SAVREG                  ;
047744  012701  050222                          MOV     #T17BFR,R1      ;SAVE R1-R5 UNTIL NEXT RETURN
047750  012702  000120                          MOV     #T17BEND-T17BFR,R2 ;GET MESSAGE BUFFER ADDRESS
047754  105021                          10$:    CLRB    (R1)+           ;SIZE OF MESSAGE BUFFER IN BYTES
047756  005302                          DEC     R2                      ;CLEAR A BYTE
047760  003375                          BGT     10$                     ;DONE?
047762  000207                          RTS     PC                      ;BR IF NO
                                                                        ;RETURN
```

```
                                        ;+
                                        ; SETUP T17PK2 PACKET FOR READ STATUS
                                        ;-
047764                                  T17SRD:
047764  004737  047740                          JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
047770  012700  050360                          MOV     #T17DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
047774  112720  000005                          MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
050000  105010                                  CLRB    (R0)                    ;CLEAR BSEL1
050002  000207                                  RTS     PC                      ;RETURN

                                        ;+
                                        ; SETUP T17PK2 PACKET FOR WRITE MISC RESET FIFO
                                        ;-
050004                                  T17RSFIF:
050004  004737  047740                          JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
050010  012700  050360                          MOV     #T17DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
050014  112720  000010                          MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
050020  112710  000030                          MOVB    #MS.RSFIF!MS.RSTAP,(R0) ;STORE BSEL1 CLEAR FIFO CODES
050024  000207                                  RTS     PC                      ;RETURN

                                        ;+
                                        ; SETUP T17PK2 PACKET FOR WRITE NPR
                                        ;
                                        ; INPUT:
                                        ;       R0 CONTAINS BSEL1 NPR DATA
                                        ;
                                        ;       SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
                                        ;-
050026                                  T17SNPR:
050026  004737  047740                          JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
050032  012701  050360                          MOV     #T17DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
050036  112721  000011                          MOVB    #PW.WNPR,(R1)+          ;STORE WRITE NPR IN BSEL0
050042  052700  000020                          BIS     #NP.WRP,R0              ;DON'T WRITE WRONG PARITY
050046  110011                                  MOVB    R0,(R1)                 ;STORE NPR DATA IN BSEL1
050050  000207                                  RTS     PC                      ;RETURN

                                        ;+
                                        ; SETUP T17PK2 PACKET FOR WRITE FIFO
                                        ;
                                        ; INPUT:
                                        ;       R0 CONTAINS BYTE COUNT
                                        ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
                                        ;-
050052                                  T17WFIF:
050052                                          SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
050056  004737  047740                          JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
050062  012702  050360                          MOV     #T17DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
050066  112722  000004                          MOVB    #PW.WFIFO,(R2)+         ;STORE WRITE FIFO IN BSEL0
050072  110022                                  MOVB    R0,(R2)+                ;STORE BYTE COUNT IN BSEL1
050074  005022                                  CLR     (R2)+                   ;CLEAR SEL2 (UNUSED)
050076  112122                          10$:    MOVB    (R1)+,(R2)+             ;STORE DATA PATTERN BYTE
050100  005300                                  DEC     R0                      ;DONE ALL BYTES?
050102  003375                                  BGT     10$                     ;BR IF NO
050104  000207                                  RTS     PC                      ;RETURN

                                        ;+
```

```
                                ; SETUP T17PK2 PACKET FOR READ FIFO
                                ;
                                ; INPUT:
                                ;       R0 CONTAINS SEL2 BYTE COUNT
                                ;-
050106                          T17RFIF:
050106  004737  047740                  JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
050112  012701  050360                  MOV     #T17DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
050116  112721  000003                  MOVB    #PW.RFIFO,(R1)+         ;STORE READ FIFO IN BSEL0
050122  110021                          MOVB    R0,(R1)+                ;STORE BYTE COUNT IN BSEL1
050124  000207                          RTS     PC                      ;RETURN
                                ;+
                                ; CLEAR EXPECTED DATA MESSAGE BUFFER
                                ;-
050126                          T17CLEXP:
050126  012701  046362                  MOV     #T17EXP,R1              ;GET EXPD ADDRESS
050132  012700  000122                  MOV     #T17EXEND-T17EXP,R0     ;GET EXPD SIZE
050136  105021                  10$:    CLRB    (R1)+                   ;CLEAR A BYTE
050140  005300                          DEC     R0                      ;DONE?
050142  003375                          BGT     10$                     ;BR IF NO
050144  000207                          RTS     PC                      ;RETURN

                                ;+
                                ;Set WORDS 0-7 of expd message buffer = to recv since not testing
                                ;-
050146                          T17SETEXP:
050146  012702  046362                  MOV     #T17EXP,R2              ;GET EXPD
050152  012703  050222                  MOV     #T17BFR,R3              ;GET READ STATUS RECV BUFFER
050156  012700  000010                  MOV     #8.,R0                  ;SET WORDS 0-7 EXP=RECV
050162  012322                  5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
050164  005300                          DEC     R0                      ;DONE WORDS 0-7 WORDS?
050166  003375                          BGT     5$                      ;BR IF NO
050170  000207                          RTS     PC                      ;RETURN

        050200                          .=<.+10>&177770
                                ;
                                ;WRITE CHARACTERISTICS COMMAND PACKET
                                ;
050200                          T17PACKET:                              ;COMMAND PACKET FOR TEST
050200  100004                          .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
050202  050210                          .WORD   T17DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
050204  000000                          .WORD   0
050206  000012                          .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE

050210                          T17DATA:                                ;CHARACTERISTICS DATA BLOCK
050210  050222                          .WORD   T17BFR                  ;ADDRESS OF MESSAGE BUFFER
050212  000000                          .WORD   0
050214  000024                          .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
050216  000000                          .WORD   0                       ;ESS,ENB,EAI,ERI
050220  000000                          .WORD   0                       ;EXTENDED FEATURES UNIT NO. ETC.


                                ;MESSAGE BUFFER FOR ALL TEST 6 COMMANDS

050222                          T17BFR:                                 ;BEGIN MESSAGE BUFFER
050222  000000                          .WORD   0                       ;MESSAGE TYPE
050224  000000                          .WORD   0                       ;DATA FIELD LENGTH
```

```
3574 050226  000000                    .WORD   0                       ;RBPCR
3575 050230  000000                    .WORD   0                       ;XST0
3576 050232  000000                    .WORD   0                       ;XST1
3577 050234  000000                    .WORD   0                       ;XST2
3578 050236  000000                    .WORD   0                       ;XST3
3579 050240  000000                    .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
3580 050242                    T17BFSTA: .BLKB 64.                       ;READ STATUS AND WRITE FIFO BUFFER
3581 050342                    T17BEND:                                  ;END OF MESSAGE BUFFER
3582                           ;
3583                           ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
3584                           ;
3586         050350                    .=<.+10>&177770
3588 050350                    T17PK2:
3589 050350  100006                    .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
3590 050352  050360                    .WORD   T17DT2                  ;LOW ADDRESS OF DATA BLOCK
3591 050354  000000                    .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
3592 050356  000012                    .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
3593
3594 050360                    T17DT2:                                   ;DATA BLOCK
3595 050360     000                    .BYTE   0                       ;BSEL0
3596 050361     000                    .BYTE   0                       ;BSEL1
3597 050362  000000                    .WORD   0                       ;SEL2
3598 050364                            .BLKB   66.                     ;WRITE FIFO DATA OUTPUT BUFFER
3599
3600 050466                    ENDTST
     050466                                                    L10056:
     050466  104401                                                   TRAP    C$ETST
3601
```

```
                                        .SBTTL  TEST  7: STATIC TRANSPORT BUS INTERFACE TEST
                                ;+
                                ; TEST DESCRIPTION:
                                ;
                                ;
                                ; TEST STEPS:
                                ;
                                ;    REPEAT FOR LOOPCNT
                                ;       BEGIN
                                ;       Write to TSSR register to soft initialize the controller
                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
                                ;       If Extended Features Hardware Switch Clear then:
                                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
                                ;       Do WRITE CHARACTERISTICS to select reserved unit 7
                                ;       Do a Write Subsystem READ STATUS
                                ;       If any transport interface signals are asserted then Print Error
                                ;       END
                                ;--


050470                                  BGNTST
050470                                                                          T7::
050470  012700  051176                   MOV     #TST18ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
050474  004737  016510                   JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
050500  012737  000012  002216           MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
050506                          T18LOOP:
                                ;       Write to TSSR register to soft initialize the controller
050506                          5$:
050506  004737  015774                   JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
050512  103405                   BCS     10$                     ;BR IF SOFT INIT OKAY
050514  010001                   MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
050516                           ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
050516  104455                                                          TRAP    C$ERDF
050520  001274                                                          .WORD   700
050522  003652                                                          .WORD   SFIERR
050524  012034                                                          .WORD   SFIMSG

                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
050526  005037  002222           10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
050532  012704  051660           MOV     #T18PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
050536  004737  010662           JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
050542                           FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
050556  103407                   BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
050560  010001                   MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
050562                           NEXT.ERRNO
050562                  12$:     ERRDF   ERRNO,T18SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
050562  104455                                                          TRAP    C$ERDF
050564  001275                                                          .WORD   701
050566  051235                                                          .WORD   T18SSR
050570  012046                                                          .WORD   PKTSSR
050572  005237  002222           INC     FATFLG                  ;SET FATAL ERROR FLAG
050576                  15$:     CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
050576  104406                                                          TRAP    C$CLP1

                                ;       If Extended Features Hardware Switch Clear then:
                                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
050600  012701  051702           MOV     #T18BFR,R1              ;MESSAGE BUFFER ADDRESS
```

```
3654 050604  032761  000200  000012          BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
3655 050612  001026                           BNE     30$                     ;BR IF YES
3656 050614  004737  051526                   JSR     PC,T18SMISC             ;SETUP PACKET FOR WRITE MISCELLANEOUS
3657 050620  012704  051730                   MOV     #T18PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3658 050624  010465  000000                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3659 050630  004737  016336                   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3660 050634                                   FORCERROR       22$                     ;@@DFORCE ERROR IF FORCER=1
3661 050650  103407                           BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
3662 050652  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3663 050654                                   NEXT.ERRNO
3664 050654                           22$:    ERRDF   ERRNO,T182SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     050654  104455                                                                   TRAP    C$ERDF
     050656  001276                                                                   .WORD   702
     050660  051272                                                                   .WORD   T182SSR
     050662  012046                                                                   .WORD   PKTSSR
3665 050664  005237  002222                   INC     FATFLG                  ;SET FATAL ERROR FLAG
3666 050670                           30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     050670  104406                                                                   TRAP    C$CLP1
3667
3668
3669                                  ;       Do WRITE CHARACTERISTICS to select reserved unit 7
3670 050672  005037  002222                   CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
3671 050676  012704  051660                   MOV     #T18PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3672 050702  004737  010662                   JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
3673 050706                                   FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
3674 050722  103407                           BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
3675 050724  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3676 050726                                   NEXT.ERRNO
3677 050726                           42$:    ERRDF   ERRNO,T18SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     050726  104455                                                                   TRAP    C$ERDF
     050730  001277                                                                   .WORD   703
     050732  051235                                                                   .WORD   T18SSR
     050734  012046                                                                   .WORD   PKTSSR
3678 050736  005237  002222                   INC     FATFLG                  ;SET FATAL ERROR FLAG
3679 050742                           50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     050742  104406                                                                   TRAP    C$CLP1
3680
3681                                  ;       Clear message buffer
3682 050744  012701  051702                   MOV     #T18BFR,R1              ;GET MESSAGE BUFFER ADDRESS
3683 050750  013700  051674                   MOV     T18DATA+4,R0            ;SIZE OF MESSAGE BUFFER IN BYTES
3684 050754  105021                   60$:    CLRB    (R1)+                   ;CLEAR A BYTE
3685 050756  005300                           DEC     R0                      ;DONE?
3686 050760  003375                           BGT     60$                     ;BR IF NO
3687
3688 050762  004737  051506                   JSR     PC,T18SRD               ;SETUP PACKET FOR READ STATUS
3689 050766  012704  051730                   MOV     #T18PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3690 050772  010465  000000                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3691 050776  004737  016336                   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3692 051002                                   FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
3693 051016  103407                           BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
3694 051020  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3695 051022                                   NEXT.ERRNO
3696 051022                           62$:    ERRDF   ERRNO,T183SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     051022  104455                                                                   TRAP    C$ERDF
     051024  001300                                                                   .WORD   704
     051026  051336                                                                   .WORD   T183SSR
     051030  012046                                                                   .WORD   PKTSSR
```

```
3697 051032  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
3698 051036                            70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     051036  104406                                                                    TRAP    C$CLP1
3699
3700
3701                                    ;       Set first 8 words of expd message buffer = to recv since not testing
3702                                    ;       Set unused bits in Read Status expd equal rcvd
3703 051040  004737  051550                    JSR     PC,T18SETEXP            ;SET SOME EXPD TO RECV
3704                                    ;       If any transport interface signals are asserted then Print Error
3705 051044  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
3706 051046  012701  051702                    MOV     #T18BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
3707 051052  012702  051146                    MOV     #T18EXP,R2              ;EXPD ADDRESS
3708 051056  012703  000012                    MOV     #10.,R3                 ;NUMBER OF WORDS TO COMPARE
3709 051062  004737  011500                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3710 051066                                    FORCERROR       82$,NOTSSR      ;@@D
3711 051076  103404                            BCS     90$                     ;BR IF YES
3712 051100                                    NEXT.ERRNO
3713 051100                            82$:    ERRHRD  ERRNO,T18CMP,MSGSTAT    ;REPORT ERROR
     051100  104456                                                                    TRAP    C$ERHRD
     051102  001301                                                                    .WORD   705
     051104  051403                                                                    .WORD   T18CMP
     051106  012350                                                                    .WORD   MSGSTAT
3714 051110                            90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     051110  104406                                                                    TRAP    C$CLP1
3715
3716 051112  005737  002222                    TST     FATFLG                  ;ANY FATAL ERRORS ?
3717 051116  001402                            BEQ     160$                    ;BRANCH IF NOT
3718 051120  004737  017202                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
3719 051124  004737  016456            160$:   JSR     PC,TSTLOOP              ;DO ITERATIONS?
3720 051130  103002                            BCC     165$                    ;BR IF NO
3721 051132  000137  050506                    JMP     T18LOOP                 ;LOOP UNTIL ITERATIONS DONE
3722 051136                            165$:
3723 051136                                    EXIT    TST
     051136  104432                                                                    TRAP    C$EXIT
     051140  000606                                                                    .WORD   L10065-.
3724
3725
```

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

051142                                  T18MSK:                         ;MASK OF UNUSED BITS IN READ STATUS BYTES
051142  377                                     .BYTE   ^C<000>         ;BYTE 0 MASK
051143  037                                     .BYTE   ^C<340>         ;BYTE 1
051144  100                                     .BYTE   ^C<277>         ;BYTE 2
051145  000                                     .BYTE   0               ;MAKE IT EVEN

051146                                  T18EXP:                         ;EXPECTED DATA BUFFER
051146  000000                                  .WORD   0               ;MESSAGE TYPE
051150  000000                                  .WORD   0               ;DATA FIELD LENGTH
051152  000000                                  .WORD   0               ;RBPCR
051154  000000                                  .WORD   0               ;XST0
051156  000000                                  .WORD   0               ;XST1
051160  000000                                  .WORD   0               ;XST2
051162  000000                                  .WORD   0               ;XST3
051164  000000                                  .WORD   0               ;XST4 (ALWAYS PRESENT FOR WRITE SUB)
051166  000000                                  .WORD   0               ;READ STATUS BYTE 1/0
051170  000000                                  .WORD   0               ;READ STATUS BYTE 2

051172  377     020             T18XS:  .BYTE   377,020                 ;READ STATUS BYTE 0/1 EXPECTED BASE
051174  000000                          .WORD   0                       ;READ STATUS BYTE 2 EXPECTED BASE

                                        ;+
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-

051176  123    164    141      TST18ID:        .ASCIZ  'Static Transport Bus Interface'
051235  127    122    111      T18SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
051272  127    122    111      T182SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
051336  127    122    111      T183SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
051403  124    162    141      T18CMP: .ASCIZ  'Transport Bus Interface Signals NOT Negated After Unit 7 Selected'
                                        .EVEN

                                        ;+
                                        ; SETUP T18PK2 PACKET FOR READ STATUS
                                        ;-
051506                                  T18SRD:
051506                                          SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
051512  012700  051740                          MOV     #T18DT2,R0      ;WRITE SUBSYSTEM DATA BUFFER
051516  112720  000005                          MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
051522  105010                                  CLRB    (R0)            ;CLEAR BSEL1
051524  000207                                  RTS     PC              ;RETURN

                                        ;+
                                        ; SETUP T18PK2 PACKET FOR WRITE MISC.
                                        ;-
051526                                  T18SMISC:
051526                                          SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
051532  012700  051740                          MOV     #T18DT2,R0      ;WRITE SUBSYSTEM DATA BUFFER
051536  112720  000010                          MOVB    #PW.WMISC,(R0)+ ;STORE WRITE MISCELLANEOUS IN BSEL0
051542  112710  000200                          MOVB    #MS.EXT,(R0)    ;STORE INVERT EXTENDED FEATURES IN BSEL1
051546  000207                                  RTS     PC              ;RETURN

                                        ;+
```

```
                                        ;Set first 8 words of expd message buffer = to recv since not testing
                                        ;       Set unused bits in Read Status expd equal rcvd
                                        ;-
051550                                  T18SETEXP:
051550  012702  051146                          MOV     #T18EXP,R2              ;GET EXPD
051554  012703  051702                          MOV     #T18BFR,R3              ;GET READ STATUS RECV BUFFER
051560  012700  000010                          MOV     #8.,R0                  ;SET FIRST 8 WORDS EXP=RECV
051564  012322                          5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
051566  005300                          DEC     R0                              ;DONE FIRST 8 WORDS?
051570  003375                          BGT     5$                              ;BR IF NO
051572  012701  051142                          MOV     #T18MSK,R1              ;GET UNUSED BIT MASK
051576  013712  051172                          MOV     T18XS,(R2)              ;SETUP BASE EXPECTED BYTE 1/0
051602  013762  051174  000002                  MOV     T18XS+2,2(R2)           ;SETUP BASE EXPECTED BYTE 2
051610  011300                          MOV     (R3),R0                         ;GET RECV BYTE 1 AND BYTE 0
051612  041100                          BIC     (R1),R0                         ;CLEAR ALL BUT UNUSED
051614  040012                          BIC     R0,(R2)                         ;CLEAR UNUSED IN EXP
051616  050012                          BIS     R0,(R2)                         ;SET UNUSED EXPD=RECV FOR COMPARE
051620  016300  000002                  MOV     2(R3),R0                        ;GET RECV BYTE 2
051624  046100  000002                  BIC     2(R1),R0                        ;CLEAR ALL BUT UNUSED
051630  040062  000002                  BIC     R0,2(R2)                        ;CLEAR UNUSED IN EXPD
051634  050062  000002                  BIS     R0,2(R2)                        ;SET UNUSED EXPD=RECV FOR COMPARE
051640  105062  000003                  CLRB    3(R2)                           ;CLEAR EXPD BYTE 3 (UNUSED)
051644  105063  000003                  CLRB    3(R3)                           ;CLEAR RECV BYTE 3 (UNUSED)
051650  000207                          RTS     PC                              ;RETURN

051660                                  .=<.+10>&177770
                                        ;
                                        ;WRITE CHARARTERISTICS COMMAND PACKET
                                        ;
051660                                  T18PACKET:                              ;COMMAND PACKET FOR TEST
051660  100004                          .WORD   100004                          ;WRITE CHARACTERISTICS COMMAND, WITH ACK
051662  051670                          .WORD   T18DATA                         ;ADDRESS OF CHARACTERISTICS BLOCK
051664  000000                          .WORD   0
051666  000012                          .WORD   10.                             ;MESSAGE PACKET MINIMUM SIZE

051670                                  T18DATA:                                ;CHARACTERISTICS DATA BLOCK
051670  051702                          .WORD   T18BFR                          ;ADDRESS OF MESSAGE BUFFER
051672  000000                          .WORD   0
051674  000024                          .WORD   20.                             ;LENGTH OF MESSAGE BUFFER
051676  000000                          .WORD   0                               ;ESS,ENB,EAI,ERI
051700  000007                          .WORD   7                               ;SELECT RESERVED UNIT 7

051702                                  T18BFR:                                 ;MESSAGE BUFFER
051702  000000                          .WORD   0                               ;MESSAGE TYPE
051704  000000                          .WORD   0                               ;DATA FIELD LENGTH
051706  000000                          .WORD   0                               ;RBPCR
051710  000000                          .WORD   0                               ;XST0
051712  000000                          .WORD   0                               ;XST1
051714  000000                          .WORD   0                               ;XST2
051716  000000                          .WORD   0                               ;XST3
051720  000000                          .WORD   0                               ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
051722  000000                          .WORD   0                               ;READ STATUS BYTE 1/0 RETURNED
051724  000000                          .WORD   0                               ;READ STATUS BYTE 2
                                        ;
                                        ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
                                        ;
```

```
3844            051730                  .=<.+10>&177770
3846 051730                     T18PK2:
3847 051730  100006                     .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
3848 051732  051740                     .WORD   T18DT2                  ;LOW ADDRESS OF DATA BLOCK
3849 051734  000000                     .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
3850 051736  000010                     .WORD   8.                      ;BUFFER EXTENT
3851
3852 051740                     T18DT2:                                 ;DATA BLOCK
3853 051740     000                     .BYTE   0                       ;BSEL0
3854 051741     000                     .BYTE   0                       ;BSEL1
3855 051742  000000                     .WORD   0                       ;SEL2
3856 051744  000000                     .WORD   0                       ;DATA
3857
3858
3859 051746                     ENDTST
     051746                                                     L10065:
     051746  104401                                                     TRAP    C$ETST
```

```
                                .SBTTL  TEST  8: TRANSPORT BUS INTERFACE LOOPBACK TEST
                        ;++
                        ; TEST DESCRIPTION:
                        ;
                        ;       This test verifies the controller's Transport Bus
                        ;       drivers, receivers, and signal loopback logic. Note
                        ;       that the Static Transport Bus test must have run
                        ;       correctly for this test to provide meaningful results.
                        ;
                        ; TEST STEPS:
                        ;
                        ;  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
                        ;--


051750                          BGNTST
051750                                                                  T8::
051750  012700  060162          MOV     #TST19ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
051754  004737  016510          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
051760  012737  000012  002216  MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
051766                  T19LOOP:
```

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

                        ;++
                        ; TEST 8: SUBTEST 1:
                        ;
                        ; SUBTEST DESCRIPTION:
                        ;
                        ;       This subtest verifies the Transport Control loopback
                        ;       path can transmit and receive correctly. The
                        ;       control signals are all loopback signals other
                        ;       than the read/write data (IW<7:0> and IR<7:0>).
                        ;
                        ; TEST STEPS:
                        ;
                        ;       The loopback signals IFAD,ITAD0,ITAD1 are the tape unit select
                        ;       lines. Since reserved unit 7 must remain selected these signals
                        ;       are always set low. This further means the signals they drive
                        ;       (ISPEED,IRDY,IONL) are only tested in the low state.
                        ;
                        ; BEGIN
                        ;       Write to TSSR register to soft initialize the controller
                        ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
                        ;       If Extended Features Hardware Switch Clear then:
                        ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
                        ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
                        ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
                        ;       Do Write Subsystem Write Control to CLEAR loopback signals group 1.
                        ;       Do Write Subsystem Write Format to CLEAR loopback signals group 2.
                        ;        (the loopback signals have to be cleared here due to the flip-flops
                        ;         that are set on a 1 to 0 transition (IHER,IFMK,ICER))
                        ;       Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
                        ;       Do a Write Subsystem READ STATUS
                        ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
                        ;         signals NOT=0 Then Print Error
                        ;
                        ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
                        ;   BEGIN
                        ;       Do Write Subsystem Write Control to Drive loopback signals group 1.
                        ;       Do Write Subsystem Write Format to Drive loopback signals group 2.
                        ;       Do a Write Subsystem READ STATUS
                        ;       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
                        ;--
051766                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
051766                                                                          T8.1:
051766  104402                                                                  TRAP    C$BSUB

                        ;       Write to TSSR register to soft initialize the controller
                        5$:
051770                          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
051770  004737  015774        BCS     10$                     ;BR IF SOFT INIT OKAY
051774  103405                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
051776  010001                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
052000
```

```
        052000  104455                                                          TRAP    C$ERDF
        052002  001440                                                          .WORD   800
        052004  003652                                                          .WORD   SFIERR
        052006  012034                                                          .WORD   SFIMSG
 3948                           ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
 3949 052010  005037  002222    10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 3950 052014  012704  062310            MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 3951 052020  004737  010662            JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
 3952 052024                            FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
 3953 052040  103407                    BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 3954 052042  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 3955 052044                            NEXT.ERRNO
 3956 052044                    12$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        052044  104455                                                          TRAP    C$ERDF
        052046  001441                                                          .WORD   801
        052050  060223                                                          .WORD   T19SSR
        052052  012046                                                          .WORD   PKTSSR
 3957 052054  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
 3958 052060                    15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        052060  104406                                                          TRAP    C$CLP1
 3959                           ;       If Extended Features Hardware Switch Clear then:
 3960                           ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
 3961 052062  012701  062332            MOV     #T19BFR,R1              ;MESSAGE BUFFER ADDRESS
 3962 052066  032761  000200  000012    BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
 3963 052074  001026                    BNE     30$                     ;BR IF YES
 3964 052076  004737  062162            JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
 3965 052102  012704  062460            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 3966 052106  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 3967 052112  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 3968 052116                            FORCERROR       22$                     ;@@DFORCE ERROR IF FORCER=1
 3969 052132  103407                    BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
 3970 052134  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 3971 052136                            NEXT.ERRNO
 3972 052136                    22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        052136  104455                                                          TRAP    C$ERDF
        052140  001442                                                          .WORD   802
        052142  060260                                                          .WORD   T192SSR
        052144  012046                                                          .WORD   PKTSSR
 3973 052146  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
 3974 052152                    30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        052152  104406                                                          TRAP    C$CLP1
 3975                           ;       Do WRITE CHARACTERISTICS to select reserved unit 7
 3976 052154  005037  002222            CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 3977 052160  012704  062310            MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 3978 052164  004737  010662            JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
 3979 052170                            FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
 3980 052204  103407                    BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
 3981 052206  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 3982 052210                            NEXT.ERRNO
 3983 052210                    42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        052210  104455                                                          TRAP    C$ERDF
        052212  001443                                                          .WORD   803
        052214  060223                                                          .WORD   T19SSR
        052216  012046                                                          .WORD   PKTSSR
 3984 052220  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
 3985 052224                    50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        052224  104406                                                          TRAP    C$CLP1
```

```
                                ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
052226  012700  000100          62$:    MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
052232  052700  000040                  BIS     #NP.LOOP,R0             ;SET LOOPBACK ENABLE
052236  004737  062022                  JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
052242  012704  062460                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
052246  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
052252  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
052256                                  FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
052272  103407                          BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
052274  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
052276                                  NEXT.ERRNO
052276                          62$:    ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
052276  104455                                                                  TRAP    C$ERDF
052300  001444                                                                  .WORD   804
052302  060371                                                                  .WORD   T194SSR
052304  012046                                                                  .WORD   PKTSSR
052306  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
052312                          70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
052312  104406                                                                  TRAP    C$CLP1
                                ;       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))
052314  005000                          CLR     R0                      ;WRITE 0'S
052316  042700  000200                  BIC     #WC.IFAD,R0             ;IFAD MUST  ALWAYS =0
052322  042700  000100                  BIC     #WC.IOTAD,R0            ;ITAD0 MUST ALWAYS =0
052326  042700  000040                  BIC     #WC.I1TAD,R0            ;ITAD1 MUST ALWAYS =0
052332  004737  062122                  JSR     PC,T19WCTL              ;SETUP PACKET FOR WRITE CONTROL
052336  012704  062460                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
052342  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
052346  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
052352                                  FORCERROR       82$                     ;@@DFORCE ERROR IF FORCER=1
052366  103407                          BCS     90$                     ;BR IF CARRY SET (GOOD RETURN)
052370  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
052372                                  NEXT.ERRNO
052372                          82$:    ERRDF   ERRNO,T197SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
052372  104455                                                                  TRAP    C$ERDF
052374  001445                                                                  .WORD   805
052376  060543                                                                  .WORD   T197SSR
052400  012046                                                                  .WORD   PKTSSR
052402  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
052406                          90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
052406  104406                                                                  TRAP    C$CLP1
052410  005000                          CLR     R0                      ;SET FORMAT DRIVE DATA=0
052412  004737  062142                  JSR     PC,T19WFMT              ;SETUP PACKET FOR WRITE FORMAT
052416  012704  062460                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
052422  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
052426  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
052432                                  FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
052446  103407                          BCS     110$                    ;BR IF CARRY SET (GOOD RETURN)
052450  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
052452                                  NEXT.ERRNO
052452                          102$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
052452  104455                                                                  TRAP    C$ERDF
052454  001446                                                                  .WORD   806
052456  060612                                                                  .WORD   T198SSR
052460  012046                                                                  .WORD   PKTSSR
```

```
4029 052462  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4030 052466                    110$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     052466  104406                                                     TRAP    C$CLP1
4031                           ;        Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
4032 052470  004737  062000            JSR     PC,T19RSFIF             ;SETUP PKT FOR WRITE MISC Reset Tape Status F-FLOPS
4033 052474  012704  062460            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4034 052500  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4035 052504  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4036 052510                            FORCERROR       122$                    ;@@DFORCE ERROR IF FORCER=1
4037 052524  103407                    BCS     130$                    ;BR IF CARRY SET (GOOD RETURN)
4038 052526  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4039 052530                            NEXT.ERRNO
4040 052530                    122$:   ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     052530  104455                                                     TRAP    C$ERDF
     052532  001447                                                     .WORD   807
     052534  060260                                                     .WORD   T192SSR
     052536  012046                                                     .WORD   PKTSSR
4041 052540  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4042 052544                    130$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     052544  104406                                                     TRAP    C$CLP1
4043                           ;        Do a Write Subsystem READ STATUS
4044                           ;        If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
4045                           ;          signals NOT=0 Then Print Error
4046 052546  004737  061760            JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
4047 052552  012704  062460            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4048 052556  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4049 052562  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4050 052566                            FORCERROR       132$                    ;@@DFORCE ERROR IF FORCER=1
4051 052602  103407                    BCS     140$                    ;BR IF CARRY SET (GOOD RETURN)
4052 052604  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4053 052606                            NEXT.ERRNO
4054 052606                    132$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     052606  104455                                                     TRAP    C$ERDF
     052610  001450                                                     .WORD   808
     052612  060324                                                     .WORD   T193SSR
     052614  012046                                                     .WORD   PKTSSR
4055 052616  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4056 052622                    140$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     052622  104406                                                     TRAP    C$CLP1
4057 052624  004737  062220            JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4058 052630  012701  060062            MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4059 052634  012702  062352            MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
4060 052640  011211                    MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
4061 052642  042711  002000            BIC     #S1.ICER,(R1)           ;SET EXPD ICER  =0
4062 052646  042711  001000            BIC     #S1.IFMK,(R1)           ;SET EXPD IFMK  =0
4063 052652  042711  000400            BIC     #S1.IHER,(R1)           ;SET EXPD IHER  =0
4064 052656  016261  000002  000002    MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
4065 052664  005000                    CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4066 052666  012701  062332            MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4067 052672  012702  060042            MOV     #T19EXP,R2              ;EXPD ADDRESS
4068 052676  012703  000024            MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
4069 052702  004737  011500            JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4070 052706                            FORCERROR       152$,NOTSSR             ;@@D
4071 052716  103404                    BCS     160$                    ;BR IF YES
4072 052720                            NEXT.ERRNO
4073 052720                    152$:   ERRHRD  ERRNO,T197CMP,MSGLOOP   ;REPORT ERROR
     052720  104456                                                     TRAP    C$ERHRD
```

```
        052722  001451                                                          .WORD   809
        052724  061263                                                          .WORD   T197CMP
        052726  013064                                                          .WORD   MSGLOOP
  4074  052730                  160$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
        052730  104406                                                          TRAP    C$CLP1
  4075
  4076  052732  005037  057774          CLR     T19PREV         ;INIT 1-0 TRANSITION FLAG
  4077  052736  012703  002752          MOV     #TSTBLK,R3      ;GET FIRST PATTERN ADDRESS
  4078  052742  012300          200$:   MOV     (R3)+,R0        ;GET A TEST PATTERN
  4079  052744  010337  002316          MOV     R3,TSTPTR       ;SAVE POINTER INTO TSTBLK
  4080  052750  042700  000200          BIC     #WC.IFAD,R0     ;IFAD MUST  ALWAYS =0
  4081  052754  042700  000100          BIC     #WC.IOTAD,R0    ;ITAD0 MUST ALWAYS =0
  4082  052760  042700  000040          BIC     #WC.I1TAD,R0    ;ITAD1 MUST ALWAYS =0
  4083  052764  010037  002312          MOV     R0,DATA         ;SET DATA PATTERN
  4084                                  ;       Do Write Subsystem Write Control to Drive loopback signals group 1.
  4085                                  ;@@D    CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
  4086  052770  013700  002312          MOV     DATA,R0         ;GET TEST PATTERN
  4087  052774  004737  062244          JSR     PC,T19CNVT      ;CONVERT PATTERN TO CONTROL DRIVE MASK
  4088                                                          ;R0 CONTAINS WRITE CONTROL DATA HERE
  4089  053000  004737  062122          JSR     PC,T19WCTL      ;SETUP PACKET FOR WRITE CONTROL
  4090  053004  012704  062460          MOV     #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
  4091  053010  010465  000000          MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS TO EXECUTE
  4092  053014  004737  016336          JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
  4093  053020                          FORCERROR       212$            ;@@DFORCE ERROR IF FORCER=1
  4094  053034  103407                  BCS     220$            ;BR IF CARRY SET (GOOD RETURN)
  4095  053036  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
  4096  053040                          NEXT.ERRNO
  4097  053040                  212$:   ERRDF   ERRNO,T197SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        053040  104455                                                          TRAP    C$ERDF
        053042  001452                                                          .WORD   810
        053044  060543                                                          .WORD   T197SSR
        053046  012046                                                          .WORD   PKTSSR
  4098  053050  005237  002222          INC     FATFLG          ;SET FATAL ERROR FLAG
  4099  053054                  220$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
        053054  104406                                                          TRAP    C$CLP1
  4100
  4101                                  ;       Do Write Subsystem Write Format to Drive loopback signals group 2.
  4102                                  ;@@D    CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
  4103  053056  013700  002312          MOV     DATA,R0         ;GET TEST PATTERN
  4104  053062  004737  062244          JSR     PC,T19CNVT      ;CONVERT PATTERN TO FORMAT DRIVE MASK
  4105  053066  000300                  SWAB    R0              ;WRITE FORMAT DATA RETURNED IN HIGH BYTE
  4106  053070  004737  062142          JSR     PC,T19WFMT      ;SETUP PACKET FOR WRITE FORMAT
  4107  053074  012704  062460          MOV     #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
  4108  053100  010465  000000          MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS TO EXECUTE
  4109  053104  004737  016336          JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
  4110  053110                          FORCERROR       232$            ;@@DFORCE ERROR IF FORCER=1
  4111  053124  103407                  BCS     240$            ;BR IF CARRY SET (GOOD RETURN)
  4112  053126  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
  4113  053130                          NEXT.ERRNO
  4114  053130                  232$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        053130  104455                                                          TRAP    C$ERDF
        053132  001453                                                          .WORD   811
        053134  060612                                                          .WORD   T198SSR
        053136  012046                                                          .WORD   PKTSSR
  4115  053140  005237  002222          INC     FATFLG          ;SET FATAL ERROR FLAG
  4116  053144                  240$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
        053144  104406                                                          TRAP    C$CLP1
```

```
4117                                    ;       Do a Write Subsystem READ STATUS
4118 053146  004737  061760                    JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
4119 053152  012704  062460                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4120 053156  010465  000000                    MOV     R4,SDB(R5)              ;SET THE PACKET ADDRESS TO EXECUTE
4121 053162  004737  016336                    JSR     PC,HKTSSR               ;WAIT FOR SSR TO SET
4122 053166                                    FORCERROR       252$                    ;@@DFORCE ERROR IF FORCER=1
4123 053202  103407                            BCS     260$                    ;BR IF CARRY SET (GOOD RETURN)
4124 053204  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4125 053206                                    NEXT.ERRNO
4126 053206                             252$:  ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053206  104455                                                                    TRAP    C$ERDF
     053210  001454                                                                    .WORD   812
     053212  060324                                                                    .WORD   T193SSR
     053214  012046                                                                    .WORD   PKTSSR
4127 053216  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4128 053222                             260$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053222  104406                                                                    TRAP    C$CLP1
4129                                    ;       If loopback data NOT= data sent Then Print Error
4130 053224  004737  062220                    JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4131 053230  012701  060062                    MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4132 053234  012702  062352                    MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
4133 053240  013711  002312                    MOV     DATA,(R1)               ;SET EXPD WORD #8 TO TEST DATA FIRST
4134 053244  013700  057774                    MOV     T19PREV,R0              ;GET PREVIOUS DATA PATTERN
4135 053250  013703  002312                    MOV     DATA,R3                 ;GET CURRENT PATTERN
4136 053254  012704  000400                    MOV     #S1.IHER,R4             ;SETUP IHER EXPECTED
4137 053260  040411                            BIC     R4,(R1)                 ;SET EXPD IHER  =0
4138 053262  030400                            BIT     R4,R0                   ;PREVIOUS =1?
4139 053264  001403                            BEQ     275$                    ;BR IF NO
4140 053266  030403                            BIT     R4,R3                   ;CURRENT  =0?
4141 053270  001001                            BNE     275$                    ;BR IF NO
4142 053272  050411                            BIS     R4,(R1)                 ;SET EXPD IHER  =1
4143 053274  012704  001000             275$:  MOV     #S1.IFMK,R4             ;SETUP IFMK EXPECTED
4144 053300  040411                            BIC     R4,(R1)                 ;SET EXPD IFMK  =0
4145 053302  030400                            BIT     R4,R0                   ;PREVIOUS =1?
4146 053304  001403                            BEQ     280$                    ;BR IF NO
4147 053306  030403                            BIT     R4,R3                   ;CURRENT  =0?
4148 053310  001001                            BNE     280$                    ;BR IF NO
4149 053312  050411                            BIS     R4,(R1)                 ;SET EXPD IFMK  =1
4150 053314  012704  002000             280$:  MOV     #S1.ICER,R4             ;SETUP ICER EXPECTED
4151 053320  040411                            BIC     R4,(R1)                 ;SET EXPD ICER  =0
4152 053322  030400                            BIT     R4,R0                   ;PREVIOUS =1?
4153 053324  001403                            BEQ     285$                    ;BR IF NO
4154 053326  030403                            BIT     R4,R3                   ;CURRENT  =0?
4155 053330  001001                            BNE     285$                    ;BR IF NO
4156 053332  050411                            BIS     R4,(R1)                 ;SET EXPD ICER  =1
4157 053334  011100                     285$:  MOV     (R1),R0                 ;GET EXPD WORD
4158                                    ;       If previous IIDENT=1 and current is IIDENT=1 then EXPD= 0 else 1
4159 053336  012704  004000                    MOV     #S1.IIDENT,R4           ;IIDENT
4160 053342  050400                            BIS     R4,R0                   ;ASSUME EXPD=1
4161 053344  030437  057774                    BIT     R4,T19PREV              ;PREVIOUS IIDENT=1?
4162 053350  001403                            BEQ     288$                    ;BR IF NO
4163 053352  030403                            BIT     R4,R3                   ;IS CURRENT IIDENT=1?
4164 053354  001401                            BEQ     288$                    ;BR IF NO
4165 053356  040400                            BIC     R4,R0                   ;SET EXPD=0
4166 053360  052700  040000             288$:  BIS     #S1.I2RES,R0            ;IRESV2 EXPD ALWAYS=1
4167 053364  052700  020000                    BIS     #S1.I1RES,R0            ;IRESV1 EXPD ALWAYS=1
4168 053370  042700  100000                    BIC     #S1.PARERR,R0           ;IGNORE PARERR
```

```
4169 053374  032712  100000          BIT     #S1.PARERR,(R2)         ;IS PARERR SET IN RECV?
4170 053400  001402                  BEQ     290$                    ;BR IF NO
4171 053402  052700  100000          BIS     #S1.PARERR,R0           ;SET IN EXPD
4172 053406  010011          290$:   MOV     R0,(R1)                 ;SETUP FINAL EXPD IN WORD #8
4173 053410  016261  000002  000002  MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
4174 053416  005000                  CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4175 053420  012701  062332          MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4176 053424  012702  060042          MOV     #T19EXP,R2              ;EXPD ADDRESS
4177 053430  012703  000024          MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
4178 053434  004737  011500          JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4179 053440                          FORCERROR       302$,NOTSSR     ;@@D
4180 053450  103404                  BCS     310$                    ;BR IF YES
4181 053452                          NEXT.ERRNO
4182 053452                  302$:   ERRHRD  ERRNO,T198CMP,MSGLOOP   ;REPORT ERROR
     053452  104456                                                          TRAP    C$ERHRD
     053454  001455                                                          .WORD   813
     053456  061351                                                          .WORD   T198CMP
     053460  013064                                                          .WORD   MSGLOOP
4183 053462                  310$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053462  104406                                                          TRAP    C$CLP1
4184                         ;       Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
4185 053464  004737  062000          JSR     PC,T19RSFIF             ;SETUP PKT FOR WRITE MISC Reset STATUS
4186 053470  012704  062460          MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4187 053474  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4188 053500  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4189 053504                          FORCERROR       322$                    ;@@DFORCE ERROR IF FORCER=1
4190 053520  103407                  BCS     330$                    ;BR IF CARRY SET (GOOD RETURN)
4191 053522  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4192 053524                          NEXT.ERRNO
4193 053524                  322$:   ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053524  104455                                                          TRAP    C$ERDF
     053526  001456                                                          .WORD   814
     053530  060260                                                          .WORD   T192SSR
     053532  012046                                                          .WORD   PKTSSR
4194 053534  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
4195 053540                  330$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053540  104406                                                          TRAP    C$CLP1
4196                         ;       Do a Write Subsystem READ STATUS
4197 053542  004737  061760          JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
4198 053546  012704  062460          MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4199 053552  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4200 053556  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4201 053562                          FORCERROR       342$                    ;@@DFORCE ERROR IF FORCER=1
4202 053576  103407                  BCS     350$                    ;BR IF CARRY SET (GOOD RETURN)
4203 053600  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4204 053602                          NEXT.ERRNO
4205 053602                  342$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053602  104455                                                          TRAP    C$ERDF
     053604  001457                                                          .WORD   815
     053606  060324                                                          .WORD   T193SSR
     053610  012046                                                          .WORD   PKTSSR
4206 053612  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
4207 053616                  350$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053616  104406                                                          TRAP    C$CLP1
4208 053620  004737  062220          JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4209 053624  012701  060062          MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4210 053630  012702  062352          MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
```

```
 4211 053634  011211                          MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
 4212 053636  042711  002000                  BIC     #S1.ICER,(R1)           ;SET EXPD ICER  =0
 4213 053642  042711  001000                  BIC     #S1.IFMK,(R1)           ;SET EXPD IFMK  =0
 4214 053646  042711  000400                  BIC     #S1.IHER,(R1)           ;SET EXPD IHER  =0
 4215 053652  016261  000002  000002          MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
 4216 053660  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 4217 053662  012701  062332                  MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 4218 053666  012702  060042                  MOV     #T19EXP,R2              ;EXPD ADDRESS
 4219 053672  012703  000024                  MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
 4220 053676  004737  011500                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 4221 053702                                  FORCERROR       362$,NOTSSR     ;a@D
 4222 053712  103404                          BCS     370$                    ;BR IF YES
 4223 053714                                  NEXT.ERRNO
 4224 053714                          362$:   ERRHRD  ERRNO,T197CMP,MSGSTAT   ;REPORT ERROR
      053714  104456                                                                  TRAP    C$ERHRD
      053716  001460                                                                  .WORD   816
      053720  061263                                                                  .WORD   T197CMP
      053722  012350                                                                  .WORD   MSGSTAT
 4225 053724                          370$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      053724  104406                                                                  TRAP    C$CLP1
 4226
 4227 053726  013737  002312  057774          MOV     DATA,T19PREV            ;SETUP PREVIOUS DATA FOR EXPD CALC.
 4228 053734  013703  002316                  MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
 4229 053740  020327  003062                  CMP     R3,#TBLEND              ;END OF TSTBLK?
 4230 053744  103002                          BHIS    400$                    ;BR IF YES
 4231 053746  000137  052742                  JMP     200$                    ;DO NEXT TSTBLK PATTERN
 4232 053752                          400$:
 4233
 4234 053752                                  ENDSUB                          ;////////////// END SUBTEST //////////
      053752                                                                          L10067:
      053752  104403                                                                  TRAP    C$ESUB
 4235
 4236 053754  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
 4237 053760  001402                          BEQ     460$                    ;BRANCH IF NOT
 4238 053762  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 4239 053766                          460$:
 4240
 4241
 4242
 4243
 4244
```

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

                ;++
                ; TEST 8: SUBTEST 2:
                ;
                ; SUBTEST DESCRIPTION:
                ;
                ;       This subtest verifies the Read/Write data loopback path.
                ;       The Read/Write data signals are IR<7:0> and IW<7:0>
                ;       respectively.
                ;
                ; TEST STEPS:
                ;
                ;
                ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
                ;    BEGIN
                ;       Write to TSSR register to soft initialize the controller
                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
                ;       If Extended Features Hardware Switch Clear then:
                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
                ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
                ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
                ;       Do a WRITE NPR to set loopback and tape direction OUT
                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
                ;       Do a READ FIFO with tape direction OUT to load tape out write latch
                ;       Do a WRITE NPR to set loopback and tape direction IN
                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
                ;         to strobe loopback data into FIFO.
                ;       Do a READ FIFO with tape direction IN to read data
                ;       If Data read from FIFO NOT= to Data sent Then Print Error
                ;       Do a Write Subsystem READ STATUS
                ;       If Input Ready  NOT=1   Then Print Error
                ;       If Output Ready NOT=0   Then Print Error
                ;       If Data In Miss NOT=0   Then Print Error
                ;    END
                ;--
053766                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
053766                                                                      T8.2:
053766  104402                                                              TRAP    C$BSUB
                ;
                ;       Write to TSSR register to soft initialize the controller
                ;
053770          5$:
053770  004737  015774  000000G      JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
053774  103405                      BCS     10$                     ;BR IF SOFT INIT OKAY
053776  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
054000                              ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
054000  104455                                                              TRAP    C$ERDF
054002  001460                                                              .WORD   816
054004  003652                                                              .WORD   SFIERR
054006  012034                                                              .WORD   SFIMSG
                ;
                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
                ;
054010  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
054014  012704  062310                  MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
054020  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
054024                                  FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
054040  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
054042  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
054044                                  NEXT.ERRNO
```

```
4297 054044                          12$:   ERRDF   ERRNO,T19SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054044  104455                                                                TRAP    C$ERDF
     054046  001461                                                                .WORD   817
     054050  060223                                                                .WORD   T19SSR
     054052  012046                                                                .WORD   PKTSSR
4298 054054  005237  002222                 INC     FATFLG                     ;SET FATAL ERROR FLAG
4299 054060                          15$:   CKLOOP                             ;LOOP ON ERROR, IF FLAG SET
     054060  104406                                                                TRAP    C$CLP1
4300                                 ;      If Extended Features Hardware Switch Clear then:
4301                                 ;        Do Write Subsystem Write Miscellaneous to Set Extended Features.
4302 054062  012701  062332                 MOV     #T19BFR,R1                 ;MESSAGE BUFFER ADDRESS
4303 054066  032761  000200  000012         BIT     #X2.EXTF,XST2(R1)          ;EXTENDED FEATURES SWITCH SET?
4304 054074  001026                         BNE     30$                        ;BR IF YES
4305 054076  004737  062162                 JSR     PC,T19SEXT                 ;SETUP PACKET FOR WRITE MISC INVERT
4306 054102  012704  062460                 MOV     #T19PK2,R4                 ;GET WRITE SUBSYSTEM COMMAND PACKET
4307 054106  010465  000000                 MOV     R4,TSDB(R5)                ;SET THE PACKET ADDRESS TO EXECUTE
4308 054112  004737  016336                 JSR     PC,CHKTSSR                 ;WAIT FOR SSR TO SET
4309 054116                                 FORCERROR       22$                        ;@@DFORCE ERROR IF FORCER=1
4310 054132  103407                         BCS     30$                        ;BR IF CARRY SET (GOOD RETURN)
4311 054134  010001                         MOV     R0,R1                      ;SAVE CONTENTS OF TSSR
4312 054136                                 NEXT.ERRNO
4313 054136                          22$:   ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054136  104455                                                                TRAP    C$ERDF
     054140  001462                                                                .WORD   818
     054142  060260                                                                .WORD   T192SSR
     054144  012046                                                                .WORD   PKTSSR
4314 054146  005237  002222                 INC     FATFLG                     ;SET FATAL ERROR FLAG
4315 054152                          30$:   CKLOOP                             ;LOOP ON ERROR, IF FLAG SET
     054152  104406                                                                TRAP    C$CLP1
4316                                 ;      Do WRITE CHARACTERISTICS to select reserved unit 7
4317 054154  012704  062310                 MOV     #T19PACKET,R4              ;GET THE ADDRESS OF COMMAND PACKET
4318 054160  004737  010662                 JSR     PC,WRTCHR                  ;DO WRITE CHARACTERISTICS COMMAND
4319 054164                                 FORCERROR       42$                        ;@@DFORCE ERROR IF FORCER=1
4320 054200  103407                         BCS     50$                        ;BR IF CARRY SET (GOOD RETURN)
4321 054202  010001                         MOV     R0,R1                      ;SAVE CONTENTS OF TSSR
4322 054204                                 NEXT.ERRNO
4323 054204                          42$:   ERRDF   ERRNO,T19SSR.PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     054204  104455                                                                TRAP    C$ERDF
     054206  001463                                                                .WORD   819
     054210  060223                                                                .WORD   T19SSR
     054212  012046                                                                .WORD   PKTSSR
4324 054214  005237  002222                 INC     FATFLG                     ;SET FATAL ERROR FLAG
4325 054220                          50$:   CKLOOP                             ;LOOP ON ERROR, IF FLAG SET
     054220  104406                                                                TRAP    C$CLP1
4326
4327
4328                                 ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4329 054222  012703  002752                 MOV     #TSTBLK,R3                 ;GET FIRST PATTERN ADDRESS
4330 054226  012337  002312          100$:  MOV     (R3)+,DATA                 ;GET A TEST PATTERN
4331 054232  042737  177400  002312         BIC     #^C<377>,DATA              ;DATA IS BYTE
4332 054240  010337  002316                 MOV     R3,TSTPTR                  ;SETUP CURRENT TSTBLK POINTER
4333                                 ;      Do a WRITE NPR to set loopback and tape direction OUT
4334 054244  012700  000100                 MOV     #NP.OUT,R0                 ;SET TAPE DIRECTION OUT
4335 054250  052700  000040                 BIS     #NP.LOOP,R0                ;SET LOOPBACK
4336 054254  004737  062022                 JSR     PC,T19SNPR                 ;SETUP T19PK2 FOR WRITE NPR
4337 054260  012704  062460                 MOV     #T19PK2,R4                 ;GET WRITE SUBSYSTEM COMMAND PACKET
4338 054264  010465  000000                 MOV     R4,TSDB(R5)                ;SET THE PACKET ADDRESS TO EXECUTE
```

```
4339 054270  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4340 054274                          FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
4341 054310  103407                  BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
4342 054312  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4343 054314                          NEXT.ERRNO
4344 054314                  102$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054314  104455                                                          TRAP    C$ERDF
     054316  001464                                                          .WORD   820
     054320  060371                                                          .WORD   T194SSR
     054322  012046                                                          .WORD   PKTSSR
4345 054324  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
4346 054330                  105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054330  104406                                                          TRAP    C$CLP1
4347                         ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4348 054332  012700  000001          MOV     #1,R0                   ;WRITE 1 BYTE
4349 054336  012701  002312          MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
4350 054342  004737  062066          JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
4351 054346  012704  062460          MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4352 054352  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4353 054356  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4354 054362                          FORCERROR       107$                    ;@@DFORCE ERROR IF FORCER=1
4355 054376  103407                  BCS     110$                    ;BR IF CARRY SET (GOOD RETURN)
4356 054400  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4357 054402                          NEXT.ERRNO
4358 054402                  107$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054402  104455                                                          TRAP    C$ERDF
     054404  001465                                                          .WORD   821
     054406  060434                                                          .WORD   T195SSR
     054410  012046                                                          .WORD   PKTSSR
4359 054412  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
4360 054416                  110$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054416  104406                                                          TRAP    C$CLP1
4361                         ;       Do a READ FIFO with tape direction OUT to load tape out write latch
4362 054420  012700  000001          MOV     #1,R0                   ;SET READ BYTE COUNT
4363 054424  004737  062046          JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
4364 054430  012704  062460          MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4365 054434  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4366 054440  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4367 054444                          FORCERROR       122$                    ;@@DFORCE ERROR IF FORCER=1
4368 054460  103407                  BCS     130$                    ;BR IF CARRY SET (GOOD RETURN)
4369 054462  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4370 054464                          NEXT.ERRNO
4371 054464                  122$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054464  104455                                                          TRAP    C$ERDF
     054466  001466                                                          .WORD   822
     054470  060500                                                          .WORD   T196SSR
     054472  012046                                                          .WORD   PKTSSR
4372 054474  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
4373 054500                  130$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054500  104406                                                          TRAP    C$CLP1
4374                         ;       Do a WRITE NPR to set loopback and tape direction IN
4375 054502  005000                  CLR     R0                      ;CLR NP.OUT TO SET TAPE DIRECTION IN
4376 054504  052700  000040          BIS     #NP.LOOP,R0             ;SET LOOPBACK
4377 054510  004737  062022          JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
4378 054514  012704  062460          MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4379 054520  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4380 054524  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
```

```
4381 054530                                     FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
4382 054544  103407                            BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
4383 054546  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4384 054550                                     NEXT.ERRNO
4385 054550                             142$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054550  104455                                                                    TRAP    C$ERDF
     054552  001467                                                                    .WORD   823
     054554  060371                                                                    .WORD   T194SSR
     054556  012046                                                                    .WORD   PKTSSR
4386 054560  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4387 054564                             150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054564  104406                                                                    TRAP    C$CLP1
4388                                    ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
4389 054566  012700  000001                    MOV     #1,R0                   ;WRITE 1 BYTE
4390 054572  012701  002312                    MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
4391 054576  004737  062066                    JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
4392 054602  012704  062460                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4393 054606  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4394 054612  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4395 054616                                     FORCERROR       162$                    ;@@DFORCE ERROR IF FORCER=1
4396 054632  103407                            BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
4397 054634  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4398 054636                                     NEXT.ERRNO
4399 054636                             162$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054636  104455                                                                    TRAP    C$ERDF
     054640  001470                                                                    .WORD   824
     054642  060434                                                                    .WORD   T195SSR
     054644  012046                                                                    .WORD   PKTSSR
4400 054646  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4401 054652                             170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054652  104406                                                                    TRAP    C$CLP1
4402                                    ;       Do a READ FIFO with tape direction IN to read data
4403                                    ;       If Data read from FIFO NOT= to Data sent Then Print Error
4404 054654  012700  000001                    MOV     #1,R0                   ;SET READ BYTE COUNT
4405 054660  004737  062046                    JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
4406 054664  012704  062460                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4407 054670  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4408 054674  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4409 054700                                     FORCERROR       182$                    ;@@DFORCE ERROR IF FORCER=1
4410 054714  103407                            BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
4411 054716  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4412 054720                                     NEXT.ERRNO
4413 054720                             182$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054720  104455                                                                    TRAP    C$ERDF
     054722  001471                                                                    .WORD   825
     054724  060500                                                                    .WORD   T196SSR
     054726  012046                                                                    .WORD   PKTSSR
4414 054730  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4415 054734                             190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054734  104406                                                                    TRAP    C$CLP1
4416 054736  004737  062220                    JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4417 054742  012701  060062                    MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4418 054746  012702  062352                    MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
4419 054752  013711  002312                    MOV     DATA,(R1)               ;SET EXPD WORD #8 = DATA
4420 054756  016261  000002  000002            MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
4421 054764  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4422 054766  012701  062332                    MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
```

```
4423 054772  012702  060042              MOV     #T19EXP,R2              ;EXPD ADDRESS
4424 054776  012703  000022              MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
4425 055002  004737  011500              JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4426 055006                              FORCERROR       202$,NOTSSR     ;@@D
4427 055016  103404                      BCS     210$                    ;BR IF YES
4428 055020                              NEXT.ERRNO
4429 055020                      202$:   ERRHRD  ERRNO,T199CMP,MSGSUB    ;REPORT ERROR
     055020  104456                                                              TRAP    C$ERHRD
     055022  001472                                                              .WORD   826
     055024  061440                                                              .WORD   T199CMP
     055026  013742                                                              .WORD   MSGSUB
4430 055030                      210$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055030  104406                                                              TRAP    C$CLP1
4431                             ;       Do a Write Subsystem READ STATUS
4432                             ;       If Input Ready  NOT=1   Then Print Error
4433                             ;       If Output Ready NOT=0   Then Print Error
4434                             ;       If Data In Miss NOT=0   Then Print Error
4435 055032  004737  061760              JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
4436 055036  012704  062460              MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4437 055042  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4438 055046  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4439 055052                              FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
4440 055066  103407                      BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
4441 055070  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4442 055072                              NEXT.ERRNO
4443 055072                      212$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     055072  104455                                                              TRAP    C$ERDF
     055074  001473                                                              .WORD   827
     055076  060324                                                              .WORD   T193SSR
     055100  012046                                                              .WORD   PKTSSR
4444 055102  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
4445 055106                      220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055106  104406                                                              TRAP    C$CLP1
4446 055110  004737  062220              JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4447 055114  012701  060062              MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4448 055120  012702  062352              MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
4449 055124  012221                      MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
4450 055126  011211                      MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
4451 055130  052711  000020              BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
4452 055134  042711  000040              BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
4453 055140  042711  000200              BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
4454 055144  005000                      CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4455 055146  012701  062332              MOV     #T196FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4456 055152  012702  060042              MOV     #T19EXP,R2              ;EXPD ADDRESS
4457 055156  012703  000024              MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
4458 055162  004737  011500              JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4459 055166                              FORCERROR       232$,NOTSSR     ;@@D
4460 055176  103404                      BCS     240$                    ;BR IF YES
4461 055200                              NEXT.ERRNO
4462 055200                      232$:   ERRHRD  ERRNO,T196CMP,MSGSTAT   ;REPORT ERROR
     055200  104456                                                              TRAP    C$ERHRD
     055202  001474                                                              .WORD   828
     055204  061200                                                              .WORD   T196CMP
     055206  012350                                                              .WORD   MSGSTAT
4463 055210                      240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055210  104406                                                              TRAP    C$CLP1
4464
```

```
4465
4466
4467 055212                                    FORCEXIT        255$            ;a@D
4468 055222  013703  002316                    MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
4469 055226  020327  003062                    CMP     R3,#TBLEND              ;END OF TSTBLK?
4470 055232  103002                           BHIS    255$                    ;BR IF YES
4471 055234  000137  054226                    JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
4472 055240                            255$:
4473
4474 055240                                    ENDSUB                          ;////////////// END SUBTEST //////////
     055240                                                                                    L10070:
     055240  104403                                                                                    TRAP    C$ESUB
4475
4476 055242  005737  002222                    TST     FATFLG                  ;ANY FATAL ERRORS ?
4477 055246  001402                           BEQ     260$                    ;BRANCH IF NOT
4478 055250  004737  017202                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4479 055254                            260$:
4480
4481
```

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

                        ;++
                        ; TEST 8: SUBTEST 3:
                        ;
                        ; SUBTEST DESCRIPTION:
                        ;
                        ;       This subtest verifies the Write Strobe loopback path
                        ;       can strobe data from the FIFO to the Data lines.
                        ;       The signal IRESV3 drives IWSTR (write strobe) to write
                        ;       data from the FIFO to the tape data out latch.
                        ;
                        ; TEST STEPS:
                        ;
                        ;
                        ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
                        ;    BEGIN
                        ;       Write to TSSR register to soft initialize the controller
                        ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
                        ;       If Extended Features Hardware Switch Clear then:
                        ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
                        ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
                        ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
                        ;       Do a WRITE NPR to set loopback and tape direction OUT
                        ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
                        ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
                        ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 0 to load write data latch
                        ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
                        ;       Do a WRITE NPR to set loopback and tape direction IN
                        ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
                        ;         to strobe loopback data into FIFO.
                        ;       Do a READ FIFO with tape direction IN to read data
                        ;       If Data read from FIFO NOT= to Data sent Then Print Error
                        ;    END
                        ;--
055254                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
055254                                                                  T8.3:
055254  104402                                                          TRAP    C$BSUB
                        ;
                        ;       Write to TSSR register to soft initialize the controller
                        ;5$:
055256                          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
055262  103405                  BCS     10$                     ;BR IF SOFT INIT OKAY
055264  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
055266                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
055266  104455                                                          TRAP    C$ERDF
055270  001474                                                          .WORD   828
055272  003652                                                          .WORD   SFIERR
055274  012034                                                          .WORD   SFIMSG
                        ;
                        ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
055276  005037  002222  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
055302  012704  062310          MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
055306  004737  010662          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
055312                          FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
055326  103407                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
055330  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
055332                          NEXT.ERRNO
055332                  12$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
```

```
                055332  104455                                                          TRAP    C$ERDF
                055334  001475                                                          .WORD   829
                055336  060223                                                          .WORD   T19SSR
                055340  012046                                                          .WORD   PKTSSR
4534            055342  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
4535            055346                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                055346  104406                                                          TRAP    C$CLP1
4536                                            ;       If Extended Features Hardware Switch Clear then:
4537                                            ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
4538            055350  012701  062332                  MOV     #T19BFR,R1              ;MESSAGE BUFFER ADDRESS
4539            055354  032761  000200  000012          BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
4540            055362  001026                          BNE     30$                     ;BR IF YES
4541            055364  004737  062162                  JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
4542            055370  012704  062460                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4543            055374  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4544            055400  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4545            055404                                  FORCERROR       22$                     ;@@DFORCE ERROR IF FORCER=1
4546            055420  103407                          BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
4547            055422  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4548            055424                                  NEXT.ERRNO
4549            055424                          22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                055424  104455                                                          TRAP    C$ERDF
                055426  001476                                                          .WORD   830
                055430  060260                                                          .WORD   T192SSR
                055432  012046                                                          .WORD   PKTSSR
4550            055434  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
4551            055440                          30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                055440  104406                                                          TRAP    C$CLP1
4552                                            ;       Do WRITE CHARACTERISTICS to select reserved unit 7
4553            055442  012704  062310                  MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
4554            055446  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
4555            055452                                  FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
4556            055466  103407                          BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
4557            055470  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4558            055472                                  NEXT.ERRNO
4559            055472                          42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                055472  104455                                                          TRAP    C$ERDF
                055474  001477                                                          .WORD   831
                055476  060223                                                          .WORD   T19SSR
                055500  012046                                                          .WORD   PKTSSR
4560            055502  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
4561            055506                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                055506  104406                                                          TRAP    C$CLP1
4562
4563                                            ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4564            055510  012703  002752                  MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
4565            055514  012337  002312          100$:   MOV     (R3)+,DATA              ;GET A TEST PATTERN
4566            055520  042737  177400  002312          BIC     #^C<377>,DATA           ;DATA IS BYTE
4567            055526  010337  002316                  MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
4568                                            ;       Do a WRITE NPR to set loopback and tape direction OUT
4569            055532  012700  000100                  MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
4570            055536  052700  000040                  BIS     #NP.LOOP,R0             ;SET LOOPBACK
4571            055542  004737  062022                  JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
4572            055546  012704  062460                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4573            055552  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4574            055556  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4575            055562                                  FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
```

```
4576 055576  103407                    BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
4577 055600  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4578 055602                            NEXT.ERRNO
4579 055602                    102$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     055602  104455                                                            TRAP    C$ERDF
     055604  001500                                                            .WORD   832
     055606  060371                                                            .WORD   T194SSR
     055610  012046                                                            .WORD   PKTSSR
4580 055612  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4581 055616                    105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055616  104406                                                            TRAP    C$CLP1
4582                           ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
4583 055620  012700  000002            MOV     #WF.I3RES,R0            ;IRESV3==>IWSTR=1
4584 055624  004737  062142            JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
4585 055630  012704  062460            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4586 055634  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4587 055640  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4588 055644                            FORCERROR       112$                    ;@@DFORCE ERROR IF FORCER=1
4589 055660  103407                    BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
4590 055662  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4591 055664                            NEXT.ERRNO
4592 055664                    112$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     055664  104455                                                            TRAP    C$ERDF
     055666  001501                                                            .WORD   833
     055670  060612                                                            .WORD   T198SSR
     055672  012046                                                            .WORD   PKTSSR
4593 055674  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4594 055700                    120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055700  104406                                                            TRAP    C$CLP1
4595                           ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4596 055702  012700  000001            MOV     #1,R0                   ;WRITE 1 BYTE
4597 055706  012701  002312            MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
4598 055712  004737  062066            JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
4599 055716  012704  062460            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4600 055722  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4601 055726  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4602 055732                            FORCERROR       132$                    ;@@DFORCE ERROR IF FORCER=1
4603 055746  103407                    BCS     140$                    ;BR IF CARRY SET (GOOD RETURN)
4604 055750  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4605 055752                            NEXT.ERRNO
4606 055752                    132$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     055752  104455                                                            TRAP    C$ERDF
     055754  001502                                                            .WORD   834
     055756  060434                                                            .WORD   T195SSR
     055760  012046                                                            .WORD   PKTSSR
4607 055762  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4608 055766                    140$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055766  104406                                                            TRAP    C$CLP1
4609                           ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 0
4610 055770  005000                    CLR     R0                      ;SET IRESV3==>IWSTR=0
4611 055772  004737  062142            JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
4612 055776  012704  062460            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4613 056002  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4614 056006  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4615 056012                            FORCERROR       152$                    ;@@DFORCE ERROR IF FORCER=1
4616 056026  103407                    BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
4617 056030  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

```
                                        NEXT.ERRNO
                                152$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                                                                                TRAP    C$ERDF
                                                                                .WORD   835
                                                                                .WORD   T198SSR
                                                                                .WORD   PKTSSR
                                        INC     FATFLG                  ;SET FATAL ERROR FLAG
                                160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                TRAP    C$CLP1
                                ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
                                        MOV     #WF.I3RES,R0            ;IRESV3==>IWSTR=1
                                        JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
                                        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       172$                    ;@@DFORCE ERROR IF FORCER=1
                                        BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        NEXT.ERRNO
                                172$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                                                                                TRAP    C$ERDF
                                                                                .WORD   836
                                                                                .WORD   T198SSR
                                                                                .WORD   PKTSSR
                                        INC     FATFLG                  ;SET FATAL ERROR FLAG
                                180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                TRAP    C$CLP1

                                ;       Do a WRITE NPR to set loopback and tape direction IN
                                        CLR     R0                      ;CLR NP.OUT TO SET TAPE DIRECTION IN
                                        BIS     #NP.LOOP,R0             ;SET LOOPBACK
                                        JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
                                        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       182$                    ;@@DFORCE ERROR IF FORCER=1
                                        BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        NEXT.ERRNO
                                182$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                                                                                TRAP    C$ERDF
                                                                                .WORD   837
                                                                                .WORD   T194SSR
                                                                                .WORD   PKTSSR
                                        INC     FATFLG                  ;SET FATAL ERROR FLAG
                                190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                TRAP    C$CLP1

                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
                                        MOV     #1,R0                   ;WRITE 1 BYTE
                                        MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
                                        JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
                                        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       202$                    ;@@DFORCE ERROR IF FORCER=1
                                        BCS     210$                    ;BR IF CARRY SET (GOOD RETURN)
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

```
                                        NEXT.ERRNO
                                202$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                                                                                TRAP    C$ERDF
                                                                                .WORD   838
                                                                                .WORD   T195SSR
                                                                                .WORD   PKTSSR
                                        INC     FATFLG                  ;SET FATAL ERROR FLAG
                                210$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                TRAP    C$CLP1
                                ;       Do a READ FIFO with tape direction IN to read data
                                        MOV     #1,R0                   ;SET READ BYTE COUNT
                                        JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
                                        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       222$                    ;@@DFORCE ERROR IF FORCER=1
                                        BCS     230$                    ;BR IF CARRY SET (GOOD RETURN)
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        NEXT.ERRNO
                                222$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                                                                                TRAP    C$ERDF
                                                                                .WORD   839
                                                                                .WORD   T196SSR
                                                                                .WORD   PKTSSR
                                        INC     FATFLG                  ;SET FATAL ERROR FLAG
                                230$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                TRAP    C$CLP1
                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
                                        JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
                                        MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
                                        MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
                                        MOV     DATA,(R1)               ;SET EXPD WORD #8 = DATA
                                        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
                                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
                                        MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
                                        MOV     #T19EXP,R2              ;EXPD ADDRESS
                                        MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
                                        JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
                                        FORCERROR       242$,NOTSSR     ;@@D
                                        BCS     250$                    ;BR IF YES
                                        NEXT.ERRNO
                                242$:   ERRHRD  ERRNO,T19WSTR,MSGSUB    ;REPORT ERROR
                                                                                TRAP    C$ERHRD
                                                                                .WORD   840
                                                                                .WORD   T19WSTR
                                                                                .WORD   MSGSUB
                                250$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                TRAP    C$CLP1


                                        FORCEXIT        255$            ;@@D
                                        MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
                                        CMP     R3,#TBLEND              ;END OF TSTBLK?
                                        BHIS    255$                    ;BR IF YES
                                        JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
                                255$:
```

```
4702 056510                                ENDSUB                    ;////////////// END SUBTEST //////////
     056510                                                                    L10071:
     056510  104403                                                                TRAP    C$ESUB
4703
4704 056512  005737  002222                TST     FATFLG            ;ANY FATAL ERRORS ?
4705 056516  001402                        BEQ     260$              ;BRANCH IF NOT
4706 056520  004737  017202                JSR     PC,CKDROP         ;TRY TO DROP THE UNIT
4707 056524                        260$:
```

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

                        ;++
                        ; TEST 8: SUBTEST 4:
                        ;
                        ; SUBTEST DESCRIPTION:
                        ;
                        ;       This subtest verifies the Read Strobe loopback path
                        ;       can strobe the data from the Data lines to the FIFO.
                        ;       The signal IRESV4 drives IRSTR (read strobe) to write
                        ;       from the data lines to the FIFO.
                        ;
                        ; TEST STEPS:
                        ;
                        ;
                        ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
                        ;    BEGIN
                        ;       Write to TSSR register to soft initialize the controller
                        ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
                        ;       If Extended Features Hardware Switch Clear then:
                        ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
                        ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
                        ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
                        ;       Do a WRITE NPR to set loopback and tape direction OUT
                        ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
                        ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
                        ;       Do a READ FIFO with tape direction OUT to load tape out write latch
                        ;       Do a WRITE NPR to set loopback and tape direction IN
                        ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 to write loop data to FIFO
                        ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
                        ;         (to strobe loopback data into FIFO.)
                        ;       Do a READ FIFO with tape direction IN to read data
                        ;       If Data read from FIFO NOT= to Data sent Then Print Error
                        ;    END
                        ;--
056524                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
056524                                                                  T8.4:
056524  104402                                                          TRAP    C$BSUB
                        ;
                        ;       Write to TSSR register to soft initialize the controller
                        5$:
056526  004737  015774          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
056532  103405                  BCS     10$                     ;BR IF SOFT INIT OKAY
056534  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
056536                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
056536  104455                                                          TRAP    C$ERDF
056540  001510                                                          .WORD   840
056542  003652                                                          .WORD   SFIERR
056544  012034                                                          .WORD   SFIMSG
                        ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
056546  005037  002222  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
056552  012704  062310          MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
056556  004737  010662          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
056562                          FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
056576  103407                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
056600  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
056602                          NEXT.ERRNO
056602                  12$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
```

```
                056602  104455                                                          TRAP    C$ERDF
                056604  001511                                                          .WORD   841
                056606  060223                                                          .WORD   T19SSR
                056610  012046                                                          .WORD   PKTSSR
 4760           056612  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
 4761           056616                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                056616  104406                                                          TRAP    C$CLP1
 4762                                           ;       If Extended Features Hardware Switch Clear then:
 4763                                           ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
 4764           056620  012701  062332                  MOV     #T19BFR,R1              ;MESSAGE BUFFER ADDRESS
 4765           056624  032761  000200  000012          BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
 4766           056632  001026                          BNE     30$                     ;BR IF YES
 4767           056634  004737  062162                  JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
 4768           056640  012704  062460                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 4769           056644  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 4770           056650  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 4771           056654                                  FORCERROR       22$                     ;@@FORCE ERROR IF FORCER=1
 4772           056670  103407                          BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
 4773           056672  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4774           056674                                  NEXT.ERRNO
 4775           056674                          22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                056674  104455                                                          TRAP    C$ERDF
                056676  001512                                                          .WORD   842
                056700  060260                                                          .WORD   T192SSR
                056702  012046                                                          .WORD   PKTSSR
 4776           056704  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
 4777           056710                          30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                056710  104406                                                          TRAP    C$CLP1
 4778                                           ;       Do WRITE CHARACTERISTICS to select reserved unit 7
 4779           056712  012704  062310                  MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 4780           056716  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
 4781           056722                                  FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
 4782           056736  103407                          BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
 4783           056740  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4784           056742                                  NEXT.ERRNO
 4785           056742                          42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                056742  104455                                                          TRAP    C$ERDF
                056744  001513                                                          .WORD   843
                056746  060223                                                          .WORD   T19SSR
                056750  012046                                                          .WORD   PKTSSR
 4786           056752  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
 4787           056756                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                056756  104406                                                          TRAP    C$CLP1
 4788
 4789                                           ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
 4790           056760  012703  002752                  MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
 4791           056764  012337  002312          100$:   MOV     (R3)+,DATA              ;GET A TEST PATTERN
 4792           056770  042737  177400  002312          BIC     #^C<377>,DATA           ;DATA IS BYTE
 4793           056776  010337  002316                  MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
 4794                                           ;       Do a WRITE NPR to set loopback and tape direction OUT
 4795           057002  012700  000100                  MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
 4796           057006  052700  000040                  BIS     #NP.LOOP,R0             ;SET LOOPBACK
 4797           057012  004737  062022                  JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
 4798           057016  012704  062460                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 4799           057022  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 4800           057026  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 4801           057032                                  FORCERROR       102$                    ;@@FORCE ERROR IF FORCER=1
```

```
 4802 057046  103407                    BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
 4803 057050  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4804 057052                            NEXT.ERRNO
 4805 057052                    102$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      057052  104455                                                            TRAP    C$ERDF
      057054  001514                                                            .WORD   844
      057056  060371                                                            .WORD   T194SSR
      057060  012046                                                            .WORD   PKTSSR
 4806 057062  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
 4807 057066                    105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      057066  104406                                                            TRAP    C$CLP1
 4808                           ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
 4809 057070  012700  000001            MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR=1
 4810 057074  004737  062142            JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
 4811 057100  012704  062460            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 4812 057104  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 4813 057110  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 4814 057114                            FORCERROR       112$                    ;@@DFORCE ERROR IF FORCER=1
 4815 057130  103407                    BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
 4816 057132  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4817 057134                            NEXT.ERRNO
 4818 057134                    112$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      057134  104455                                                            TRAP    C$ERDF
      057136  001515                                                            .WORD   845
      057140  060612                                                            .WORD   T198SSR
      057142  012046                                                            .WORD   PKTSSR
 4819 057144  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
 4820 057150                    120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      057150  104406                                                            TRAP    C$CLP1
 4821                           ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
 4822 057152  012700  000001            MOV     #1,R0                   ;WRITE 1 BYTE
 4823 057156  012701  002312            MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
 4824 057162  004737  062066            JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRIiE FIFO
 4825 057166  012704  062460            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 4826 057172  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 4827 057176  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 4828 057202                            FORCERROR       132$                    ;@@DFORCE ERROR IF FORCER=1
 4829 057216  103407                    BCS     140$                    ;BR IF CARRY SET (GOOD RETURN)
 4830 057220  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4831 057222                            NEXT.ERRNO
 4832 057222                    132$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      057222  104455                                                            TRAP    C$ERDF
      057224  001516                                                            .WORD   846
      057226  060434                                                            .WORD   T195SSR
      057230  012046                                                            .WORD   PKTSSR
 4833 057232  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
 4834 057236                    140$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      057236  104406                                                            TRAP    C$CLP1
 4835                           ;       Do a READ FIFO with tape direction OUT to load tape out write latch
 4836 057240  012700  000001            MOV     #1,R0                   ;SET READ BYTE COUNT
 4837 057244  004737  062046            JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
 4838 057250  012704  062460            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 4839 057254  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 4840 057260  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 4841 057264                            FORCERROR       152$                    ;@@DFORCE ERROR IF FORCER=1
 4842 057300  103407                    BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
 4843 057302  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

```
4844 057304                                    NEXT.ERRNO
4845 057304                            152$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057304  104455                                                             TRAP    C$ERDF
     057306  001517                                                             .WORD   847
     057310  060500                                                             .WORD   T196SSR
     057312  012046                                                             .WORD   PKTSSR
4846 057314  005237  002222                       INC     FATFLG                ;SET FATAL ERROR FLAG
4847 057320                            160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057320  104406                                                             TRAP    C$CLP1
4848                                   ;       Do a WRITE NPR to set loopback and tape direction IN
4849 057322  005000                            CLR     R0                      ;CLR NP.OUT TO SET TAPE DIRECTION IN
4850 057324  052700  000040                    BIS     #NP.LOOP,R0             ;SET LOOPBACK
4851 057330  004737  062022                    JSR     PC,T19$NPR              ;SETUP T19PK2 FOR WRITE NPR
4852 057334  012704  062460                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4853 057340  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4854 057344  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4855 057350                                    FORCERROR       182$                    ;@@DFORCE ERROR IF FORCER=1
4856 057364  103407                            BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
4857 057366  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4858 057370                                    NEXT.ERRNO
4859 057370                            182$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057370  104455                                                             TRAP    C$ERDF
     057372  001520                                                             .WORD   848
     057374  060371                                                             .WORD   T194SSR
     057376  012046                                                             .WORD   PKTSSR
4860 057400  005237  002222                       INC     FATFLG                ;SET FATAL ERROR FLAG
4861 057404                            190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057404  104406                                                             TRAP    C$CLP1
4862                                   ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 0
4863 057406  005000                            CLR     R0                      ;SET IRESV4==>IRSTR=0
4864 057410  004737  062142                    JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
4865 057414  012704  062460                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4866 057420  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4867 057424  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4868 057430                                    FORCERROR       202$                    ;@@DFORCE ERROR IF FORCER=1
4869 057444  103407                            BCS     210$                    ;BR IF CARRY SET (GOOD RETURN)
4870 057446  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4871 057450                                    NEXT.ERRNO
4872 057450                            202$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057450  104455                                                             TRAP    C$ERDF
     057452  001521                                                             .WORD   849
     057454  060612                                                             .WORD   T198SSR
     057456  012046                                                             .WORD   PKTSSR
4873 057460  005237  002222                       INC     FATFLG                ;SET FATAL ERROR FLAG
4874 057464                            210$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057464  104406                                                             TRAP    C$CLP1
4875                                   ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
4876 057466  012700  000001                    MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR=1
4877 057472  004737  062142                    JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
4878 057476  012704  062460                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4879 057502  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4880 057506  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4881 057512                                    FORCERROR       222$                    ;@@DFORCE ERROR IF FORCER=1
4882 057526  103407                            BCS     230$                    ;BR IF CARRY SET (GOOD RETURN)
4883 057530  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4884 057532                                    NEXT.ERRNO
4885 057532                            222$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
```

```
        057532  104455                                                          TRAP    C$ERDF
        057534  001522                                                          .WORD   850
        057536  060612                                                          .WORD   T198SSR
        057540  012046                                                          .WORD   PKTSSR
 4886   057542  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 4887   057546                         230$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        057546  104406                                                          TRAP    C$CLP1
 4888                                   ;       Do a READ FIFO with tape direction IN to read data
 4889   057550  012700  000001                 MOV     #1,R0                   ;SET READ BYTE COUNT
 4890   057554  004737  062046                 JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
 4891   057560  012704  062460                 MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 4892   057564  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 4893   057570  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 4894   057574                                 FORCERROR       282$                    ;@@DFORCE ERROR IF FORCER=1
 4895   057610  103407                         BCS     290$                    ;BR IF CARRY SET (GOOD RETURN)
 4896   057612  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4897   057614                                 NEXT.ERRNO
 4898   057614                         282$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        057614  104455                                                          TRAP    C$ERDF
        057616  001523                                                          .WORD   851
        057620  060500                                                          .WORD   T196SSR
        057622  012046                                                          .WORD   PKTSSR
 4899   057624  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 4900   057630                         290$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        057630  104406                                                          TRAP    C$CLP1
 4901                                   ;       If Data read from FIFO NOT= to Data sent Then Print Error
 4902   057632  004737  062220                 JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
 4903   057636  012701  060062                 MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
 4904   057642  012702  062352                 MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
 4905   057646  013711  002312                 MOV     DATA,(R1)               ;SET EXPD WORD #8 = DATA
 4906   057652  016261  000002  000002         MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
 4907   057660  005000                         CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
 4908   057662  012701  062332                 MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
 4909   057666  012702  060042                 MOV     #T19EXP,R2              ;EXPD ADDRESS
 4910   057672  012703  000022                 MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
 4911   057676  004737  011500                 JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
 4912   057702                                 FORCERROR       302$,NOTSSR     ;@@D
 4913   057712  103404                         BCS     310$                    ;BR IF YES
 4914   057714                                 NEXT.ERRNO
 4915   057714                         302$:   ERRHRD  ERRNO,T19RSTR,MSGSUB    ;REPORT ERROR
        057714  104456                                                          TRAP    C$ERHRD
        057716  001524                                                          .WORD   852
        057720  061630                                                          .WORD   T19RSTR
        057722  013742                                                          .WORD   MSGSUB
 4916   057724                         310$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        057724  104406                                                          TRAP    C$CLP1
 4917
 4918
 4919   057726                                 FORCEXIT        355$            ;@@D
 4920   057736  013703  002316                 MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
 4921   057742  020327  003062                 CMP     R3,#TBLEND              ;END OF TSTBLK?
 4922   057746  103002                         BHIS    355$                    ;BR IF YES
 4923   057750  000137  056764                 JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
 4924   057754                         355$:
 4925
 4926   057754                                 ENDSUB                          ;////////////// END SUBTEST //////////
        057754                                                                                  L10072:
```

```
        057754  104403                                                  TRAP    C$ESUB
 4927
 4928 057756  005737  002222          TST     FATFLG          ;ANY FATAL ERRORS ?
 4929 057762  001402                  BEQ     360$            ;BRANCH IF NOT
 4930 057764  004737  017202          JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
 4931 057770                  360$:
 4932
 4933 057770                  EXIT    TST                     ;////////// EXIT TEST //////////
        057770  104432                                                  TRAP    C$EXIT
        057772  002602                                                  .WORD   L10066-.
 4934
 4935
```

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

057774  000000                          T19PREV:        .WORD   0               ;DRIVE SIGNAL 1-0 TRANSITION FLAG
                                        ;+
                                        ; LOOPBACK DRIVE SIGNAL TABLE
                                        ; THIS TABLE IS USED BY T19CNVT TO SETUP
                                        ; A DRIVE PATTERN FROM THE TEST DATA INPUT PATTERN.
                                        ;
                                        ; WRITE CONTROL SIGNALS ARE OF FORM WC.XXX
                                        ; WRITE FORMAT SIGNALS ARE OF FORM WF.XXXX
                                        ;-
057776                                  T19BFCTL:                               ;WRITE CONTROL DRIVE SIGNALS
057776  000001                                    WC.IGO                      ;IGO==>IFPT     DATA<0>
060000  000002                                    WC.IFEN                     ;IFEN==>IFBY    DATA<1>
060002  000004                                    WC.IRWU                     ;IRWU==>IRWD    DATA<2>
060004  000010                                    WC.IREW                     ;IREW==>IDBY    DATA<3>
060006  002000                                    WF.IERASE*256.              ;IFAD==>ILDP    DATA<4>
060010  000040                                    WC.I1TAD                    ;ITAD1==>IONL   DATA<5>
060012  000100                                    WC.I0TAD                    ;ITAD0==>IRDY   DATA<6>
060014  000200                                    WC.IFAD                     ;IERASE==>ISPEED DATA<7>
060016  004000                                    WF.IEDIT*256.               ;IEDIT==>IHER   DATA<8>
060020  010000                                    WF.IWFM*256.                ;IWFM==>IFMK    DATA<9>
060022  020000                                    WF.IREV*256.                ;IREV==>ICER    DATA<10>
060024  040000                                    WF.IWRT*256.                ;IWRT==>IIDENT  DATA<11>
060026  100000                                    WF.IHISP*256.               ;IHISP==>IEOT   DATA<12>
060030  000000                                    .WORD   0                   ;IRESV2 (UNUSED)DATA<13>
060032  000000                                    .WORD   0                   ;IRESV1 (UNUSED)DATA<14>
060034  000000                                    .WORD   0                   ;PARERR (UNTESTED)DATA<15>

060036                                  T19MSK:                                 ;MASK OF UNTESTED BITS IN READ STATUS BYTES
                                                                                ;UNTESTED BITS ARE SEi TO 1
060036      377                                 .BYTE   ^C<000>                 ;BYTE 0 MASK
060037      037                                 .BYTE   ^C<340>                 ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
060040      360                                 .BYTE   ^C<017>                 ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
060041      000                                 .BYTE   0                       ;MAKE IT EVEN

060042                                  T19EXP:                                 ;BEGIN EXPECTED DATA BUFFER
060042  000000                                  .WORD   0                       ;MESSAGE TYPE
060044  000000                                  .WORD   0                       ;DATA FIELD LENGTH
060046  000000                                  .WORD   0                       ;RBPCR
060050  000000                                  .WORD   0                       ;XST0
060052  000000                                  .WORD   0                       ;XST1
060054  000000                                  .WORD   0                       ;XST2
060056  000000                                  .WORD   0                       ;XST3
060060  000000                                  .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
060062                                  T19EXSTA: .BLKB 64.                     ;EXPECTED READ STATUS AND WRITE FIFO DATA
060162                                  T19EXEND:                                       ;END EXPECTED DATA BUFFER
                                        ;+
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-

060162      124     162     141         TST19ID:        .ASCIZ  'Transport Bus Interface Loopback'
060223      127     122     111         T19SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
060260      127     122     111         T192SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
060324      127     122     111         T193SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
```

```
 4994 060371      127      122      111  T194SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
 4995 060434      127      122      111  T195SSR:.ASCIZ  'WRITE SUBSYSTEM (Write FIFO) Failed'
 4996 060500      127      122      111  T196SSR:.ASCIZ  'WRITE SUBSYSTEM (Read FIFO) Failed'
 4997 060543      127      122      111  T197SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Control) Failed'
 4998 060612      127      122      111  T198SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Format) Failed'
 4999 060660      106      111      106  T191CMP:.ASCIZ  'FIFO Status in WORD #9 Incorrect after Initialize'
 5000 060742      122      145      141  T192CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
 5001 061036      124      141      160  T193CMP:.ASCIZ  'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
 5002 061124      122      145      141  T195CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO Data'
 5003 061200      106      111      106  T196CMP:.ASCIZ  'FIFO Status (in WORD #9) Incorrect after READ FIFO'
 5004 061263      124      141      160  T197CMP:.ASCIZ  'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
 5005 061351      103      157      156  T198CMP:.ASCIZ  'Control Signal Loopback Data Error, Data is in WORD #8'
 5006 061440      122      145      141  T199CMP:.ASCIZ  'Read/Write Loopback Data Error, Data is in WORD #8'
 5007 061523      114      157      157  T19WSTR:.ASCIZ  'Loopback Data Error when strobed by Write strobe, Data is in WORD #8'
 5008 061630      114      157      157  T19RSTR:.ASCIZ  'Loopback Data Error when strobed by Read Strobe, Data is in WORD #8'
 5009
 5010                                            .EVEN
 5011
 5012                                    ;+
 5013                                    ; CLEAR MESSAGE BUFFER
 5014                                    ;-
 5015 061734                             T19CLRBUF:
 5016 061734                                     SAVREG                          ;
 5017 061740  012701  062332                     MOV     #T19BFR,R1              ;SAVE R1-R5 UNTIL NEXT RETURN
 5018 061744  012702  000120                     MOV     #T19BEND-T19BFR,R2      ;GET MESSAGE BUFFER ADDRESS
 5019 061750  105021                     10$:    CLRB    (R1)+                   ;SIZE OF MESSAGE BUFFER IN BYTES
 5020 061752  005302                             DEC     R2                      ;CLEAR A BYTE
 5021 061754  003375                             BGT     10$                     ;DONE?
 5022 061756  000207                             RTS     PC                      ;BR IF NO
 5023                                                                            ;RETURN
 5024                                    ;+
 5025                                    ; SETUP T19PK2 PACKET FOR READ STATUS
 5026                                    ;-
 5027 061760                             T19SRD:
 5028 061760  004737  061734                     JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
 5029 061764  012700  062470                     MOV     #T19DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
 5030 061770  112720  000005                     MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
 5031 061774  105010                             CLRB    (R0)                    ;CLEAR BSEL1
 5032 061776  000207                             RTS     PC                      ;RETURN
 5033
 5034                                    ;+
 5035                                    ; SETUP T19PK2 PACKET FOR WRITE MISC Reset Tape Status F-FLOPS
 5036                                    ;-
 5037 062000                             T19RSFIF:
 5038 062000  004737  061734                     JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
 5039 062004  012700  062470                     MOV     #T19DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
 5040 062010  112720  000010                     MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
 5041 062014  112710  000030                     MOVB    #MS.RSFIF!MS.RSTAP,(R0) ;STORE BSEL1 CLEAR FIFO CODES
 5042 062020  000207                             RTS     PC                      ;RETURN
 5043
 5044                                    ;+
 5045                                    ; SETUP T19PK2 PACKET FOR WRITE NPR
 5046                                    ;
 5047                                    ; INPUT:
 5048                                    ;       R0 CONTAINS BSEL1 NPR DATA
 5049                                    ;
 5050                                    ;       SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
```

```
5051                                    ;-
5052 062022                             T19SNPR:
5053 062022  004737  061734                    JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
5054 062026  012701  062470                    MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5055 062032  112721  000011                    MOVB    #PW.WNPR,(R1)+          ;STORE WRITE NPR IN BSEL0
5056 062036  052700  000020                    BIS     #NP.WRP,R0              ;DON'T WRITE WRONG PARITY
5057 062042  110011                            MOVB    R0,(R1)                 ;STORE NPR DATA IN BSEL1
5058 062044  000207                            RTS     PC                      ;RETURN
5059
5060                                    ;+
5061                                    ; SETUP T19PK2 PACKET FOR READ FIFO
5062                                    ;
5063                                    ; INPUT:
5064                                    ;       R0 CONTAINS SEL2 BYTE COUNT
5065                                    ;-
5066 062046                             T19RFIF:
5067 062046  004737  061734                    JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
5068 062052  012701  062470                    MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5069 062056  112721  000003                    MOVB    #PW.RFIFO,(R1)+         ;STORE READ FIFO IN BSEL0
5070 062062  110021                            MOVB    R0,(R1)+                ;STORE BYTE COUNT IN BSEL1
5071 062064  000207                            RTS     PC                      ;RETURN
5072                                    ;+
5073                                    ; SETUP T19PK2 PACKET FOR WRITE FIFO
5074                                    ;
5075                                    ; INPUT:
5076                                    ;       R0 CONTAINS BYTE COUNT
5077                                    ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
5078                                    ;-
5079 062066                             T19WFIF:
5080 062066                                    SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
5081 062072  004737  061734                    JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
5082 062076  012702  062470                    MOV     #T19DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
5083 062102  112722  000004                    MOVB    #PW.WFIFO,(R2)+         ;STORE WRITE FIFO IN BSEL0
5084 062106  110022                            MOVB    R0,(R2)+                ;STORE BYTE COUNT IN BSEL1
5085 062110  005022                            CLR     (R2)+                   ;CLEAR SEL2 (UNUSED)
5086 062112  112122                    10$:    MOVB    (R1)+,(R2)+             ;STORE DATA PATTERN BYTE
5087 062114  005300                            DEC     R0                      ;DONE ALL BYTES?
5088 062116  003375                            BGT     10$                     ;BR IF NO
5089 062120  000207                            RTS     PC                      ;RETURN
5090                                    ;+
5091                                    ; SETUP T19PK2 FOR WRITE CONTROL
5092                                    ;
5093                                    ; INPUT:
5094                                    ;       R0 CONTAINS DRIVING DATA PATTERN
5095                                    ;-
5096 062122                             T19WCTL:
5097 062122  004737  061734                    JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
5098 062126  012701  062470                    MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5099 062132  112721  000006                    MOVB    #PW.WCTL,(R1)+          ;STORE WRITE CONTROL IN BSEL0
5100 062136  110021                            MOVB    R0,(R1)+                ;STORE DATA WORD IN BSEL1
5101 062140  000207                            RTS     PC                      ;RETURN
5102                                    ;+
5103                                    ; SETUP T19PK2 FOR WRITE FORMAT TRANSPORT REGISTER
5104                                    ;
5105                                    ; INPUT:
5106                                    ;       R0 CONTAINS DRIVING DATA PATTERN
5107                                    ;-
```

```
5108 062142
5109 062142  004737  061734          T19WFMT:
                                             JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
5110 062146  012701  062470                    MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5111 062152  112721  000007                    MOVB    #PW.WFMT,(R1)+          ;STORE WRITE FORMAT IN BSEL0
5112 062156  110021                            MOVB    R0,(R1)+                ;STORE DATA WORD IN BSEL1
5113 062160  000207                            RTS     PC                      ;RETURN
5114                                        ;+
5115                                        ; SETUP T19PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
5116                                        ;-
5117 062162                                 T19SEXT:
5118 062162  012700  062470                    MOV     #T19DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
5119 062166  112720  000010                    MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
5120 062172  112710  000200                    MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
5121 062176  000207                            RTS     PC                      ;RETURN
5122                                        ;+
5123                                        ; CLEAR EXPECTED DATA MESSAGE BUFFER
5124                                        ;-
5125 062200                                 T19CLEXP:
5126 062200  012701  060042                    MOV     #T19EXP,R1              ;GET EXPD ADDRESS
5127 062204  012700  000120                    MOV     #T19EXEND-T19EXP,R0     ;GET EXPD SIZE
5128 062210  105021                     10$:   CLRB    (R1)+                   ;CLEAR A BYTE
5129 062212  005300                            DEC     R0                      ;DONE?
5130 062214  003375                            BGT     10$                     ;BR IF NO
5131 062216  000207                            RTS     PC                      ;RETURN
5132
5133                                        ;+
5134                                        ;Set WORDS 0-7 of expd message BUFFER = to recv since not testing
5135                                        ;-
5136 062220                                 T19SETEXP:
5137 062220  012702  060042                    MOV     #T19EXP,R2              ;GET EXPD
5138 062224  012703  062332                    MOV     #T19BFR,R3              ;GET READ STATUS RECV BUFFER
5139 062230  012700  000010                    MOV     #8.,R0                  ;SET WORDS 0-7 EXP=RECV
5140 062234  012322                     5$:    MOV     (R3)+,(R2)+             ;SET EXPD=RECV
5141 062236  005300                            DEC     R0                      ;DONE WORDS 0-7 WORDS?
5142 062240  003375                            BGT     5$                      ;BR IF NO
5143 062242  000207                            RTS     PC                      ;RETURN
5144
5145                                        ;+
5146                                        ; CONVERT A TEST PATTERN DATA WORD TO LOOPBACK DRIVE SIGNALS
5147                                        ;
5148                                        ; INPUTS:
5149                                        ;
5150                                        ;       R0      TEST PATTERN
5151                                        ;
5152                                        ; IMPLICIT INPUTS:
5153                                        ;
5154                                        ;       T19BFCTL  - CONTAINS WRITE CONTROL / WRITE FORMAT CONVERSION BITS
5155                                        ;
5156                                        ; OUTPUTS:
5157                                        ;
5158                                        ;       R0      - LOW BYTE CONTAINS WRITE CONTROL DATA
5159                                        ;               - HIGH BYTE CONTAINS WRITE FORMAT DATA
5160                                        ;-
5161 062244                                 T19CNVT:
5162 062244                                    SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
5163 062250  012701  057776                    MOV     #T19BFCTL,R1            ;CONVERSION TABLE ADDRESS
5164 062254  005002                            CLR     R2                      ;INIT RESULT OF CONVERSION
```

```
5165 062256  012703  000020          MOV     #16.,R3         ;BIT COUNT
5166 062262  006000          10$:    ROR     R0              ;IS THIS BIT EQUAL TO 1?
5167 062264  103001                  BCC     20$             ;BR IF NO
5168 062266  051102                  BIS     (R1),R2         ;SET CONVERTED BIT
5169 062270  005721          20$:    TST     (R1)+           ;POINT TO NEXT BIT IN CONVERSION TABLE
5170 062272  005303                  DEC     R3              ;DONE?
5171 062274  003372                  BGT     10$             ;BR IF NO
5172 062276  010200                  MOV     R2,R0           ;COPY RESULT
5173 062300  000207                  RTS     PC              ;RETURN
5174
5175
5176
5178         062310                  .=<.+10>&177770
5180                                 ;
5181                                 ;WRITE CHARACTERISTICS COMMAND PACKET
5182                                 ;
5183 062310                          T19PACKET:                              ;COMMAND PACKET FOR TEST
5184 062310  100004                          .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
5185 062312  062320                          .WORD   T19DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
5186 062314  000000                          .WORD   0
5187 062316  000012                          .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
5188
5189 062320                          T19DATA:                                ;CHARACTERISTICS DATA BLOCK
5190 062320  062332                          .WORD   T19BFR                  ;ADDRESS OF MESSAGE BUFFER
5191 062322  000000                          .WORD   0
5192 062324  000024                          .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
5193 062326  000000                          .WORD   0                       ;ESS,ENB,EAI,ERI
5194 062330  000007                          .WORD   7                       ;EXTENDED FEATURES UNIT NO.
5195
5196
5197                                 ;MESSAGE BUFFER FOR ALL TEST 8 COMMANDS
5198
5199 062332                          T19BFR:                                 ;BEGIN MESSAGE BUFFER
5200 062332  000000                          .WORD   0                       ;MESSAGE TYPE
5201 062334  000000                          .WORD   0                       ;DATA FIELD LENGTH
5202 062336  000000                          .WORD   0                       ;RBPCR
5203 062340  000000                          .WORD   0                       ;XST0
5204 062342  000000                          .WORD   0                       ;XST1
5205 062344  000000                          .WORD   0                       ;XST2
5206 062346  000000                          .WORD   0                       ;XST3
5207 062350  000000                          .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
5208 062352                          T19BFSTA: .BLKB 64.                     ;READ STATUS AND WRITE FIFO BUFFER
5209 062452                          T19BEND:                                ;END OF MESSAGE BUFFER
5210                                 ;
5211                                 ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
5212                                 ;
5214         062460                  .=<.+10>&177770
5216 062460                          T19PK2:
5217 062460  100006                          .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
5218 062462  062470                          .WORD   T19DT2                  ;LOW ADDRESS OF DATA BLOCK
5219 062464  000000                          .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
5220 062466  000012                          .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
5221
5222 062470                          T19DT2:                                 ;DATA BLOCK
5223 062470     000                          .BYTE   0                       ;BSEL0
5224 062471     000                          .BYTE   0                       ;BSEL1
5225 062472  000000                          .WORD   0                       ;SEL2
```

```
 5226 062474                            .BLKB   64.             ;WRITE FIFO DATA OUTPUT BUFFER
 5227
 5228
 5229 062574                            ENDTST
      062574                                                    L10066:
      062574  104401                                                    TRAP    C$ETST
```

```
                        .SBTTL  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 STEPS:
                ;
                ;  REPEAT FOR LOOPCNT
                ;    BEGIN
                ;       Write to TSSR register to soft initialize the controller
                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
                ;       If Extended Features Hardware Switch Clear then:
                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
                ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
                ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
                ;    BEGIN
                ;       (* Verify Write Wrong Parity Sets Parity Error *)
                ;       Do a WRITE NPR to set loopback and tape direction OUT
                ;               and SET Write Wrong Parity.
                ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
                ;       Do a READ FIFO with tape direction OUT to load tape out write latch
                ;         (this is when wrong parity (IWP) is set)
                ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 0 (sets read strobe low)
                ;            (Read Strobe sets PAR IN H [Parity Error])
                ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
                ;       Do a Write Subsystem READ STATUS
                ;       If Read Data parity error NOT=1  Then Print Error
                ;       Do a Write Misc to RESET FIFO
                ;       Do a Write Subsystem READ STATUS
                ;       If Read Data parity error NOT=0  Then Print Error
                ;
                ;       (* Verify Data can be transferred without a Parity Error *)
                ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
                ;       Do a WRITE NPR to set loopback and tape direction OUT
                ;               and CLEAR Write Wrong Parity.
                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
                ;       Do a READ FIFO with tape direction OUT to load tape out write latch
                ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 0 (sets read strobe low)
                ;            (Read Strobe should NOT set PAR IN H [Parity Error] here)
                ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
                ;       Do a Write Subsystem READ STATUS
                ;       If Read Data parity error NOT=0  Then Print Error
                ;
```

```
                                ;    END
                                ;--


                                        BGNTST                                                  T9::
                                T20LOOP:
                                        BGNSUB                          ;////////////// BEGIN SUBTEST //////////
                                                                                                T9.1:
                                ;       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
                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
                                10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
                                        MOV     #T20PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
                                        JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
                                        FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
                                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        NEXT.ERRNO
                                12$:    ERRDF   ERRNO,T20SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                                        INC     FATFLG                  ;SET FATAL ERROR FLAG
                                15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                ;       If Extended Features Hardware Switch Clear then:
                                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
                                        MOV     #T20BFR,R1              ;MESSAGE BUFFER ADDRESS
                                        BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
                                        BNE     30$                     ;BR IF YES
                                        JSR     PC,T20SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
                                        MOV     #T20PK2,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,T202SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
```

```
062576                                  TRAP    C$BSUB
062616                                  TRAP    C$ERDF
062630  001604                          .WORD   900
062632  003652                          .WORD   SFIERR
062634  012034                          .WORD   SFIMSG
062672  104455                          TRAP    C$ERDF
062674  001605                          .WORD   901
062676  065211                          .WORD   T20SSR
062700  012046                          .WORD   PKTSSR
062706  104406                          TRAP    C$CLP1
062764  104455                          TRAP    C$ERDF
062766  001606                          .WORD   902
062770  065246                          .WORD   T202SSR
```

```
                062772  012046                                                          .WORD   PKTSSR
5334            062774  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
5335            063000                          30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                063000  104406                                                          TRAP    C$CLP1
5336                                            ;       Do WRITE CHARACTERISTICS to select reserved unit 7
5337            063002  012704  066360                  MOV     #T20PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
5338            063006  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
5339            063012                                  FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
5340            063026  103407                          BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
5341            063030  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5342            063032                                  NEXT.ERRNO
5343            063032                          42$:    ERRDF   ERRNO,T20SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                063032  104455                                                          TRAP    C$ERDF
                063034  001607                                                          .WORD   903
                063036  065211                                                          .WORD   T20SSR
                063040  012046                                                          .WORD   PKTSSR
5344            063042  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
5345            063046                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                063046  104406                                                          TRAP    C$CLP1
5346
5347
5348                                            ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
5349            063050  012703  002752                  MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
5350            063054  012337  002312          100$:   MOV     (R3)+,DATA              ;GET A TEST PATTERN
5351            063060  042737  177400  002312          BIC     #^C<377>,DATA           ;DATA IS BYTE
5352            063066  010337  002316                  MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
5353                                            ;       Do a WRITE NPR to set loopback and tape direction OUT and
5354                                            ;               and SET Write Wrong Parity.
5355            063072  012700  000100                  MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
5356            063076  052700  000040                  BIS     #NP.LOOP,R0             ;SET LOOPBACK
5357            063102  042700  000020                  BIC     #NP.WRP,R0              ;SET WRITE WRONG PARITY (INVERTED)
5358            063106  004737  066146                  JSR     PC,T20WNPR              ;SETUP T20PK2 FOR WRITE NPR
5359            063112  012704  066530                  MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5360            063116  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5361            063122  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5362            063126                                  FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
5363            063142  103407                          BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
5364            063144  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5365            063146                                  NEXT.ERRNO
5366            063146                          102$:   ERRDF   ERRNO,T204SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                063146  104455                                                          TRAP    C$ERDF
                063150  001610                                                          .WORD   904
                063152  065357                                                          .WORD   T204SSR
                063154  012046                                                          .WORD   PKTSSR
5367            063156  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
5368            063162                          105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                063162  104406                                                          TRAP    C$CLP1
5369                                            ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
5370            063164  012700  000001                  MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR = 1
5371            063170  004737  066242                  JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
5372            063174  012704  066530                  MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5373            063200  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5374            063204  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5375            063210                                  FORCERROR       112$                    ;@@DFORCE ERROR IF FORCER=1
5376            063224  103407                          BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
5377            063226  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5378            063230                                  NEXT.ERRNO
```

```
5379 063230                    112$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063230  104455                                                            TRAP    C$ERDF
     063232  001611                                                            .WORD   905
     063234  065531                                                            .WORD   T208SSR
     063236  012046                                                            .WORD   PKTSSR
5380 063240  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
5381 063244                    120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063244  104406                                                            TRAP    C$CLP1
5382                           ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
5383 063246  012700  000001            MOV     #1,R0                   ;WRITE 1 BYTE
5384 063252  012701  002312            MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
5385 063256  004737  066206            JSR     PC,T20WFIF              ;SETUP T20PK2 FOR WRITE FIFO
5386 063262  012704  066530            MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5387 063266  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5388 063272  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5389 063276                            FORCERROR       152$                    ;@@DFORCE ERROR IF FORCER=1
5390 063312  103407                    BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
5391 063314  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5392 063316                            NEXT.ERRNO
5393 063316                    152$:   ERRDF   ERRNO,T205SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063316  104455                                                            TRAP    C$ERDF
     063320  001612                                                            .WORD   906
     063322  065422                                                            .WORD   T205SSR
     063324  012046                                                            .WORD   PKTSSR
5394 063326  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
5395 063332                    160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063332  104406                                                            TRAP    C$CLP1
5396                           ;       Do a READ FIFO with tape direction OUT to load tape out write latch
5397                           ;         (this is when wrong parity (IWP) is set)
5398 063334  012700  000001            MOV     #1,R0                   ;SET READ BYTE COUNT
5399 063340  004737  066166            JSR     PC,T20RFIF              ;SETUP T20PK2 FOR READ FIFO
5400 063344  012704  066530            MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5401 063350  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5402 063354  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5403 063360                            FORCERROR       172$                    ;@@DFORCE ERROR IF FORCER=1
5404 063374  103407                    BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
5405 063376  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5406 063400                            NEXT.ERRNO
5407 063400                    172$:   ERRDF   ERRNO,T206SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063400  104455                                                            TRAP    C$ERDF
     063402  001613                                                            .WORD   907
     063404  065466                                                            .WORD   T206SSR
     063406  012046                                                            .WORD   PKTSSR
5408 063410  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
5409 063414                    180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063414  104406                                                            TRAP    C$CLP1
5410                           ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 0 (sets read strobe low)
5411                           ;               (Read Strobe sets PAR IN H [Parity Error])
5412 063416  005000                    CLR     R0                      ;IRESV4==>IRSTR = 0
5413 063420  004737  066242            JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
5414 063424  012704  066530            MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5415 063430  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5416 063434  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5417 063440                            FORCERROR       192$                    ;@@DFORCE ERROR IF FORCER=1
5418 063454  103407                    BCS     200$                    ;BR IF CARRY SET (GOOD RETURN)
5419 063456  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5420 063460                            NEXT.ERRNO
```

```
5421 063460                          192$:  ERRDF   ERRNO,T208SSR,PKTSSR   ;DEVICE FATAL SSR FAILED TO SET
     063460  104455                                                              TRAP    C$ERDF
     063462  001614                                                              .WORD   908
     063464  065531                                                              .WORD   T208SSR
     063466  012046                                                              .WORD   PKTSSR
5422 063470  005237  002222                INC     FATFLG                      ;SET FATAL ERROR FLAG
5423 063474                          200$:  CKLOOP                              ;LOOP ON ERROR, IF FLAG SET
     063474  104406                                                              TRAP    C$CLP1
5424                                 ;      Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
5425 063476  012700  000001                MOV     #WF.I4RES,R0                ;IRESV4==>IRSTR = 1
5426 063502  004737  066242                JSR     PC,T20WFMT                  ;SETUP T20PK2 FOR WRITE FORMAT
5427 063506  012704  066530                MOV     #T20PK2,R4                  ;GET WRITE SUBSYSTEM COMMAND PACKET
5428 063512  010465  000000                MOV     R4,TSDB(R5)                 ;SET THE PACKET ADDRESS TO EXECUTE
5429 063516  004737  016336                JSR     PC,CHKTSSR                  ;WAIT FOR SSR TO SET
5430 063522                                FORCERROR       212$                        ;@@DFORCE ERROR IF FORCER=1
5431 063536  103407                        BCS     220$                        ;BR IF CARRY SET (GOOD RETURN)
5432 063540  010001                        MOV     R0,R1                       ;SAVE CONTENTS OF TSSR
5433 063542                                NEXT.ERRNO
5434 063542                          212$:  ERRDF   ERRNO,T208SSR,PKTSSR   ;DEVICE FATAL SSR FAILED TO SET
     063542  104455                                                              TRAP    C$ERDF
     063544  001615                                                              .WORD   909
     063546  065531                                                              .WORD   T208SSR
     063550  012046                                                              .WORD   PKTSSR
5435 063552  005237  002222                INC     FATFLG                      ;SET FATAL ERROR FLAG
5436 063556                          220$:  CKLOOP                              ;LOOP ON ERROR, IF FLAG SET
     063556  104406                                                              TRAP    C$CLP1
5437                                 ;      Do a Write Subsystem READ STATUS
5438 063560  004737  066126                JSR     PC,T20SRD                   ;SETUP PACKET FOR READ STATUS
5439 063564  012704  066530                MOV     #T20PK2,R4                  ;GET WRITE SUBSYSTEM COMMAND PACKET
5440 063570  010465  000000                MOV     R4,TSDB(R5)                 ;SET THE PACKET ADDRESS TO EXECUTE
5441 063574  004737  016336                JSR     PC,CHKTSSR                  ;WAIT FOR SSR TO SET
5442 063600                                FORCERROR       232$                        ;@@DFORCE ERROR IF FORCER=1
5443 063614  103407                        BCS     240$                        ;BR IF CARRY SET (GOOD RETURN)
5444 063616  010001                        MOV     R0,R1                       ;SAVE CONTENTS OF TSSR
5445 063620                                NEXT.ERRNO
5446 063620                          232$:  ERRDF   ERRNO,T203SSR,PKTSSR   ;DEVICE FATAL SSR FAILED TO SET
     063620  104455                                                              TRAP    C$ERDF
     063622  001616                                                              .WORD   910
     063624  065312                                                              .WORD   T203SSR
     063626  012046                                                              .WORD   PKTSSR
5447 063630  005237  002222                INC     FATFLG                      ;SET FATAL ERROR FLAG
5448 063634                          240$:  CKLOOP                              ;LOOP ON ERROR, IF FLAG SET
     063634  104406                                                              TRAP    C$CLP1
5449                                 ;      If Read Data parity error NOT=1  Then Print Error
5450 063636  004737  066334                JSR     PC,T20SETEXP                ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
5451 063642  012701  065062                MOV     #T20EXSTA,R1                ;GET EXPECTED READ STATUS
5452 063646  012702  066422                MOV     #T20BFSTA,R2                ;GET RECV READ STATUS
5453 063652  011211                        MOV     (R2),(R1)                   ;SET EXPD WORD #8 = RECV TEMP
5454 063654  016261  000002  000002        MOV     2(R2),2(R1)                 ;SET EXPD WORD #9 = RECV (NOT TESTED)
5455 063662  052711  100000                BIS     #S1.PARERR,(R1)             ;SET EXP PAR ERR  =1
5456 063666  005000                        CLR     R0                          ;HIGH RECV ADDRESS FOR CKMSG2
5457 063670  012701  066402                MOV     #T20BFR,R1                  ;LOW RECV ADDRESS FOR CKMSG2
5458 063674  012702  065042                MOV     #T20EXP,R2                  ;EXPD ADDRESS
5459 063700  012703  000024                MOV     #20.,R3                     ;NUMBER OF BYTES TO COMPARE
5460 063704  004737  011500                JSR     PC,CKMSG2                   ;EXPD EQUAL RECV?
5461 063710                                FORCERROR       252$,NOTSSR                 ;@@D
5462 063720  103404                        BCS     260$                        ;BR IF YES
```

```
5463 063722                                    NEXT.ERRNO
5464 063722                            252$:   ERRHRD  ERRNO,T20SWP,MSGSTAT    ;REPORT ERROR
     063722  104456                                                            TRAP    C$ERHRD
     063724  001617                                                            .WORD   911
     063726  065577                                                            .WORD   T20SWP
     063730  012350                                                            .WORD   MSGSTAT
5465 063732                            260$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063732  104406                                                            TRAP    C$CLP1
5466                                   ;       Do a Write Misc to RESET FIFO
5467 063734  012700  000020                      MOV     #MS.RSFIF,R0           ;SET RESET FIFO COMMAND
5468 063740  004737  066262                      JSR     PC,T20WMISC            ;SETUP T20PK2 FOR WRITE MISC
5469 063744  012704  066530                      MOV     #T20PK2,R4             ;GET WRITE SUBSYSTEM COMMAND PACKET
5470 063750  010465  000000                      MOV     R4,TSDB(R5)            ;SET THE PACKET ADDRESS TO EXECUTE
5471 063754  004737  016336                      JSR     PC,CHKTSSR             ;WAIT FOR SSR TO SET
5472 063760                                      FORCERROR       282$                    ;@@DFORCE ERROR IF FORCER=1
5473 063774  103407                              BCS     290$                   ;BR IF CARRY SET (GOOD RETURN)
5474 063776  010001                              MOV     R0,R1                  ;SAVE CONTENTS OF TSSR
5475 064000                                      NEXT.ERRNO
5476 064000                            282$:   ERRDF   ERRNO,T202SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064000  104455                                                            TRAP    C$ERDF
     064002  001620                                                            .WORD   912
     064004  065246                                                            .WORD   T202SSR
     064006  012046                                                            .WORD   PKTSSR
5477 064010  005237  002222                      INC     FATFLG                 ;SET FATAL ERROR FLAG
5478 064014                            290$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064014  104406                                                            TRAP    C$CLP1
5479                                   ;       Do a Write Subsystem READ STATUS
5480                                   ;       If Read Data parity error NOT=0  Then Print Error
5481 064016  004737  066334                      JSR     PC,T20SETEXP           ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
5482 064022  012701  065062                      MOV     #T20EXSTA,R1           ;GET EXPECTED READ STATUS
5483 064026  012702  066422                      MOV     #T20BFSTA,R2           ;GET RECV READ STATUS
5484 064032  011211                              MOV     (R2),(R1)              ;SET EXPD WORD #8 = RECV TEMP
5485 064034  016261  000002  000002              MOV     2(R2),2(R1)            ;SET EXPD WORD #9 = RECV (NOT TESTED)
5486 064042  042711  100000                      BIC     #S1.PARERR,(R1)        ;SET EXP PAR ERR  =0
5487 064046  005000                              CLR     R0                     ;HIGH RECV ADDRESS FOR CKMSG2
5488 064050  012701  066402                      MOV     #T20BFR,R1             ;LOW RECV ADDRESS FOR CKMSG2
5489 064054  012702  065042                      MOV     #T20EXP,R2             ;EXPD ADDRESS
5490 064060  012703  000024                      MOV     #20.,R3                ;NUMBER OF BYTES TO COMPARE
5491 064064  004737  011500                      JSR     PC,CKMSG2              ;EXPD EQUAL RECV?
5492 064070                                      FORCERROR       302$,NOTSSR    ;@@D
5493 064100  103404                              BCS     320$                   ;BR IF YES
5494 064102                                      NEXT.ERRNO
5495 064102                            302$:   ERRHRD  ERRNO,T20RSF,MSGSTAT    ;REPORT ERROR
     064102  104456                                                            TRAP    C$ERHRD
     064104  001621                                                            .WORD   913
     064106  065706                                                            .WORD   T20RSF
     064110  012350                                                            .WORD   MSGSTAT
5496 064112                            320$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064112  104406                                                            TRAP    C$CLP1
5497                                   ;       (* Verify Data can be transferred without a Parity Error *)
5498                                   ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
5499 064114  012700  000001                      MOV     #WF.I4RES,R0           ;IRESV4==>IRSTR = 1
5500 064120  004737  066242                      JSR     PC,T20WFMT             ;SETUP T20PK2 FOR WRITE FORMAT
5501 064124  012704  066530                      MOV     #T20PK2,R4             ;GET WRITE SUBSYSTEM COMMAND PACKET
5502 064130  010465  000000                      MOV     R4,TSDB(R5)            ;SET THE PACKET ADDRESS TO EXECUTE
5503 064134  004737  016336                      JSR     PC,CHKTSSR             ;WAIT FOR SSR TO SET
5504 064140                                      FORCERROR       332$                    ;@@DFORCE ERROR IF FORCER=1
```

```
 5505 064154  103407                    BCS     340$                    ;BR IF CARRY SET (GOOD RETURN)
 5506 064156  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5507 064160                            NEXT.ERRNO
 5508 064160                    332$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      064160  104455                                                            TRAP    C$ERDF
      064162  001622                                                            .WORD   914
      064164  065531                                                            .WORD   T208SSR
      064166  012046                                                            .WORD   PKTSSR
 5509 064170  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
 5510 064174                    340$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      064174  104406                                                            TRAP    C$CLP1
 5511                           ;       Do a WRITE NPR to set loopback and tape direction OUT and
 5512                           ;                and CLEAR Write Wrong Parity.
 5513 064176  012700  000100            MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
 5514 064202  052700  000040            BIS     #NP.LOOP,R0             ;SET LOOPBACK
 5515 064206  052700  000020            BIS     #NP.WRP,R0              ;CLEAR WRITE WRONG PARITY (INVERTED)
 5516 064212  004737  066146            JSR     PC,T20WNPR              ;SETUP T20PK2 FOR WRITE NPR
 5517 064216  012704  066530            MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 5518 064222  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 5519 064226  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 5520 064232                            FORCERROR       352$                    ;@@DFORCE ERROR IF FORCER=1
 5521 064246  103407                    BCS     360$                    ;BR IF CARRY SET (GOOD RETURN)
 5522 064250  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5523 064252                            NEXT.ERRNO
 5524 064252                    352$:   ERRDF   ERRNO,T204SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      064252  104455                                                            TRAP    C$ERDF
      064254  001623                                                            .WORD   915
      064256  065357                                                            .WORD   T204SSR
      064260  012046                                                            .WORD   PKTSSR
 5525 064262  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
 5526 064266                    360$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      064266  104406                                                            TRAP    C$CLP1
 5527                           ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
 5528 064270  012700  000001            MOV     #1,R0                   ;WRITE 1 BYTE
 5529 064274  012701  002312            MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
 5530 064300  004737  066206            JSR     PC,T20WFIF              ;SETUP T20PK2 FOR WRITE FIFO
 5531 064304  012704  066530            MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 5532 064310  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 5533 064314  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 5534 064320                            FORCERROR       372$                    ;@@DFORCE ERROR IF FORCER=1
 5535 064334  103407                    BCS     380$                    ;BR IF CARRY SET (GOOD RETURN)
 5536 064336  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5537 064340                            NEXT.ERRNO
 5538 064340                    372$:   ERRDF   ERRNO,T205SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      064340  104455                                                            TRAP    C$ERDF
      064342  001624                                                            .WORD   916
      064344  065422                                                            .WORD   T205SSR
      064346  012046                                                            .WORD   PKTSSR
 5539 064350  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
 5540 064354                    380$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      064354  104406                                                            TRAP    C$CLP1
 5541                           ;       Do a READ FIFO with tape direction OUT to load tape out write latch
 5542 064356  012700  000001            MOV     #1,R0                   ;SET READ BYTE COUNT
 5543 064362  004737  066166            JSR     PC,T20RFIF              ;SETUP T20PK2 FOR READ FIFO
 5544 064366  012704  066530            MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 5545 064372  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 5546 064376  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
```

```
5547 064402                                FORCERROR       392$                    ;@@DFORCE ERROR IF FORCER=1
5548 064416  103407                        BCS     400$                    ;BR IF CARRY SET (GOOD RETURN)
5549 064420  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5550 064422                                NEXT.ERRNO
5551 064422                        392$:   ERRDF   ERRNO,T206SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064422  104455                                                                TRAP    C$ERDF
     064424  001625                                                                .WORD   917
     064426  065466                                                                .WORD   T206SSR
     064430  012046                                                                .WORD   PKTSSR
5552 064432  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
5553 064436                        400$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064436  104406                                                                TRAP    C$CLP1
5554                               ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 0 (sets read strobe low)
5555                               ;               (Read Strobe sets PAR IN H [Parity Error])
5556 064440  005000                        CLR     R0                      ;IRESV4==>IRSTR = 0
5557 064442  004737  066242                JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
5558 064446  012704  066530                MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5559 064452  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5560 064456  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5561 064462                                FORCERROR       412$                    ;@@DFORCE ERROR IF FORCER=1
5562 064476  103407                        BCS     420$                    ;BR IF CARRY SET (GOOD RETURN)
5563 064500  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5564 064502                                NEXT.ERRNO
5565 064502                        412$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064502  104455                                                                TRAP    C$ERDF
     064504  001626                                                                .WORD   918
     064506  065531                                                                .WORD   T208SSR
     064510  012046                                                                .WORD   PKTSSR
5566 064512  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
5567 064516                        420$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064516  104406                                                                TRAP    C$CLP1
5568                               ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
5569 064520  012700  000001                MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR = 1
5570 064524  004737  066242                JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
5571 064530  012704  066530                MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5572 064534  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5573 064540  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5574 064544                                FORCERROR       432$                    ;@@DFORCE ERROR IF FORCER=1
5575 064560  103407                        BCS     440$                    ;BR IF CARRY SET (GOOD RETURN)
5576 064562  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5577 064564                                NEXT.ERRNO
5578 064564                        432$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064564  104455                                                                TRAP    C$ERDF
     064566  001627                                                                .WORD   919
     064570  065531                                                                .WORD   T208SSR
     064572  012046                                                                .WORD   PKTSSR
5579 064574  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
5580 064600                        440$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064600  104406                                                                TRAP    C$CLP1
5581
5582                               ;       Do a Write Subsystem READ STATUS
5583 064602  004737  066126                JSR     PC,T20SRD               ;SETUP PACKET FOR READ STATUS
5584 064606  012704  066530                MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5585 064612  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5586 064616  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5587 064622                                FORCERROR       452$                    ;@@DFORCE ERROR IF FORCER=1
5588 064636  103407                        BCS     460$                    ;BR IF CARRY SET (GOOD RETURN)
```

```
5589 064640  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5590 064642                                NEXT.ERRNO
5591 064642                        452$:   ERRDF   ERRNO,T203SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064642  104455                                                        TRAP    C$ERDF
     064644  001630                                                        .WORD   920
     064646  065312                                                        .WORD   T203SSR
     064650  012046                                                        .WORD   PKTSSR
5592 064652  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
5593 064656                        460$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064656  104406                                                        TRAP    C$CLP1
5594                               ;       If Read Data parity error NOT=0  Then Print Error
5595 064660  004737  066334                JSR     PC,T20SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
5596 064664  012701  065062                MOV     #T20EXSTA,R1            ;GET EXPECTED READ STATUS
5597 064670  012702  066422                MOV     #T20BFSTA,R2            ;GET RECV READ STATUS
5598 064674  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
5599 064676  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTED)
5600 064704  042711  100000                BIC     #S1.PARERR,(R1)         ;SET EXP PAR ERR  =0
5601 064710  005000                        .LR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
5602 064712  012701  066402                MOV     #T20BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
5603 064716  012702  065042                MOV     #T20EXP,R2              ;EXPD ADDRESS
5604 064722  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
5605 064726  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
5606 064732                                FORCERROR       472$,NOTSSR     ;@@D
5607 064742  103404                        BCS     480$                    ;BR IF YES
5608 064744                                NEXT.ERRNO
5609 064744                        472$:   ERRHRD  ERRNO,T20CWP,MSGSTAT    ;REPORT ERROR
     064744  104456                                                        TRAP    C$ERHRD
     064746  001631                                                        .WORD   921
     064750  066007                                                        .WORD   T20CWP
     064752  012350                                                        .WORD   MSGSTAT
5610 064754                        480$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064754  104406                                                        TRAP    C$CLP1
5611
5612 064756                                FORCEXIT        555$            ;@@D
5613 064766  013703  002316                MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
5614 064772  020327  003062                CMP     R3,#TBLEND              ;END OF TSTBLK?
5615 064776  103002                        BHIS    555$                    ;BR IF YES
5616 065000  000137  063054                JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
5617 065004                        555$:
5618
5619 065004                                ENDSUB                          ;////////////// END SUBTEST //////////
     065004                                                                L10074:
     065004  104403                                                        TRAP    C$ESUB
5620
5621 065006  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
5622 065012  001402                        BEQ     560$                    ;BRANCH IF NOT
5623 065014  004737  017202                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
5624 065020                        560$:
5625 065020  004737  016456                JSR     PC,TSTLOOP              ;DO ITERATIONS?
5626 065024  103002                        BCC     565$                    ;BR IF NO
5627 065026  000137  050506                JMP     T18LOOP                 ;LOOP UNTIL ITERATIONS DONE
5628 065032                        565$:
5629 065032                                EXIT    TST                     ;////////// EXIT TEST //////////
     065032  104432                                                        TRAP    C$EXIT
     065034  001610                                                        .WORD   L10073-.
5630
5631
```

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


065036                                  T20MSK:                         ;MASK OF UNTESTED BITS IN READ STATUS
                                                                        ;UNTESTED BITS ARE SET TO 1
065036      377                                 .BYTE   ^C<000>         ;BYTE 0 MASK
065037      037                                 .BYTE   ^C<340>         ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
065040      360                                 .BYTE   ^C<017>         ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
065041      000                                 .BYTE   0               ;MAKE IT EVEN

065042                                  T20EXP:                         ;BEGIN EXPECTED DATA BUFFER
065042  000000                              .WORD   0                   ;MESSAGE TYPE
065044  000000                              .WORD   0                   ;DATA FIELD LENGTH
065046  000000                              .WORD   0                   ;RBPCR
065050  000000                              .WORD   0                   ;XST0
065052  000000                              .WORD   0                   ;XST1
065054  000000                              .WORD   0                   ;XST2
065056  000000                              .WORD   0                   ;XST3
065060  000000                              .WORD   0                   ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
065062                                  T20EXSTA: .BLKB 64.             ;EXPECTED READ STATUS AND WRITE FIFO DATA
065162                                  T20EXEND:                               ;END EXPECTED DATA BUFFER
                                        ;+
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-

065162      122     145     141     TST20ID:        .ASCIZ  'Read/Write Data Parity'
065211      127     122     111     T20SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
065246      127     122     111     T202SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
065312      127     122     111     T203SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
065357      127     122     111     T204SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
065422      127     122     111     T205SSR:.ASCIZ  'WRITE SUBSYSTEM (Write FIFO) Failed'
065466      127     122     111     T206SSR:.ASCIZ  'WRITE SUBSYSTEM (Read FIFO) Failed'
065531      127     122     111     T208SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Format) Failed'
065577      122     145     141     T20SWP: .ASCIZ  'Read Data Parity Error (PARERR) Failed to Set after Write Wrong Parity'
065706      122     145     141     T20RSF: .ASCIZ  'Read Data Parity Error (PARERR) Failed to Clear after RESET FIFO'
066007      122     145     141     T20CWP: .ASCIZ  'Read Data Parity Error (PARERR) occurred in Data Loopback'
                                            .EVEN

                                        ;+
                                        ; CLEAR MESSAGE BUFFER
                                        ;-
066102                                  T20CLRBUF:                      ;
066102                                      SAVREG                      ;SAVE R1-R5 UNTIL NEXT RETURN
066106  012701  066402                      MOV     #T20BFR,R1          ;GET MESSAGE BUFFER ADDRESS
066112  012702  000120                      MOV     #T20BEND-T20BFR,R2  ;SIZE OF MESSAGE BUFFER IN BYTES
066116  105021                      10$:    CLRB    (R1)+               ;CLEAR A BYTE
066120  005302                              DEC     R2                  ;DONE?
066122  003375                              BGT     10$                 ;BR IF NO
066124  000207                              RTS     PC                  ;RETURN

                                        ;+
                                        ; SETUP T20PK2 PACKET FOR READ STATUS
                                        ;-
066126                                  T20SRD:
066126  004737  066102                      JSR     PC,T20CLRBUF        ;CLEAR MESSAGE BUFFER
```

```
5690 066132  012700  066540                MOV     #T20DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
5691 066136  112720  000005                MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
5692 066142  105010                        CLRB    (R0)                    ;CLEAR BSEL1
5693 066144  000207                        RTS     PC                      ;RETURN
5694
5695
5696                                ;+
5697                                ; SETUP T20PK2 PACKET FOR WRITE NPR
5698                                ;
5699                                ; INPUT:
5700                                ;       R0 CONTAINS BSEL1 NPR DATA
5701                                ;
5702                                ;-
5703 066146                        T20WNPR:
5704 066146  004737  066102                JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
5705 066152  012701  066540                MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5706 066156  112721  000011                MOVB    #PW.WNPR,(R1)+          ;STORE WRITE NPR IN BSEL0
5707 066162  110011                        MOVB    R0,(R1)                 ;STORE NPR DATA IN BSEL1
5708 066164  000207                        RTS     PC                      ;RETURN
5709
5710                                ;+
5711                                ; SETUP T20PK2 PACKET FOR READ FIFO
5712                                ;
5713                                ; INPUT:
5714                                ;       R0 CONTAINS SEL2 BYTE COUNT
5715                                ;-
5716 066166                        T20RFIF:
5717 066166  004737  066102                JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
5718 066172  012701  066540                MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5719 066176  112721  000003                MOVB    #PW.RFIFO,(R1)+         ;STORE READ FIFO IN BSEL0
5720 066202  110021                        MOVB    R0,(R1)+                ;STORE BYTE COUNT IN BSEL1
5721 066204  000207                        RTS     PC                      ;RETURN
5722                                ;+
5723                                ; SETUP T20PK2 PACKET FOR WRITE FIFO
5724                                ;
5725                                ; INPUT:
5726                                ;       R0 CONTAINS BYTE COUNT
5727                                ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
5728                                ;-
5729 066206                        T20WFIF:
5730 066206                                SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
5731 066212  004737  066102                JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
5732 066216  012702  066540                MOV     #T20DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
5733 066222  112722  000004                MOVB    #PW.WFIFO,(R2)+         ;STORE WRITE FIFO IN BSEL0
5734 066226  110022                        MOVB    R0,(R2)+                ;STORE BYTE COUNT IN BSEL1
5735 066230  005022                        CLR     (R2)+                   ;CLEAR SEL2 (UNUSED)
5736 066232  112122                10$:    MOVB    (R1)+,(R2)+             ;STORE DATA PATTERN BYTE
5737 066234  005300                        DEC     R0                      ;DONE ALL BYTES?
5738 066236  003375                        BGT     10$                     ;BR IF NO
5739 066240  000207                        RTS     PC                      ;RETURN
5740
5741                                ;+
5742                                ; SETUP T20PK2 FOR WRITE FORMAT TRANSPORT REGISTER
5743                                ;
5744                                ; INPUT:
5745                                ;       R0 CONTAINS DRIVING DATA PATTERN
5746                                ;-
```

```
5747 066242                            T20WFMT:
5748 066242  004737  066102                    JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
5749 066246  012701  066540                    MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5750 066252  112721  000007                    MOVB    #PW.WFMT,(R1)+          ;STORE WRITE FORMAT IN BSEL0
5751 066256  110021                            MOVB    R0,(R1)+                ;STORE DATA WORD IN BSEL1
5752 066260  000207                            RTS     PC                      ;RETURN
5753                                   ;+
5754                                   ; SETUP T20PK2 PACKET FOR WRITE MISC.
5755                                   ;
5756                                   ;       R0 CONTAINS WRITE MISC DATA
5757                                   ;-
5758 066262                            T20WMISC:
5759 066262  012701  066540                    MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5760 066266  112721  000010                    MOVB    #PW.WMISC,(R1)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
5761 066272  110011                            MOVB    R0,(R1)                 ;STORE INVERT EXTENDED FEATURES IN BSEL1
5762 066274  000207                            RTS     PC                      ;RETURN
5763                                   ;+
5764                                   ; SETUP T20PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
5765                                   ;-
5766 066276                            T20SEXT:
5767 066276  012700  066540                    MOV     #T20DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
5768 066302  112720  000010                    MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
5769 066306  112710  000200                    MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
5770 066312  000207                            RTS     PC                      ;RETURN
5771                                   ;+
5772                                   ; CLEAR EXPECTED DATA MESSAGE BUFFER
5773                                   ;-
5774 066314                            T20CLEXP:
5775 066314  012701  065042                    MOV     #T20EXP,R1              ;GET EXPD ADDRESS
5776 066320  012700  000120                    MOV     #T20EXEND-T20EXP,R0     ;GET EXPD SIZE
5777 066324  105021                    10$:    CLRB    (R1)+                   ;CLEAR A BYTE
5778 066326  005300                            DEC     R0                      ;DONE?
5779 066330  003375                            BGT     10$                     ;BR IF NO
5780 066332  000207                            RTS     PC                      ;RETURN
5781
5782                                   ;+
5783                                   ;Set WORDS 0-7 of expd message BUFFER = to recv since not testing
5784                                   ;-
5785 066334                            T20SETEXP:
5786 066334  012702  065042                    MOV     #T20EXP,R2              ;GET EXPD
5787 066340  012703  066402                    MOV     #T20BFR,R3              ;GET READ STATUS RECV BUFFER
5788 066344  012700  000010                    MOV     #8.,R0                  ;SET WORDS 0-7 EXP=RECV
5789 066350  012322                    5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
5790 066352  005300                            DEC     R0                      ;DONE WORDS 0-7 WORDS?
5791 066354  003375                            BGT     5$                      ;BR IF NO
5792 066356  000207                            RTS     PC                      ;RETURN
5793
5794
5795
5799                                   ;
5800                                   ;WRITE CHARACTERISTICS COMMAND PACKET
5801                                   ;
5802 066360                            T20PACKET:                              ;COMMAND PACKET FOR TEST
5803 066360  100004                            .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
5804 066362  066370                            .WORD   T20DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
5805 066364  000000                            .WORD   0
5806 066366  000012                            .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
```

```
5807
5808 066370                    T20DATA:                        ;CHARACTERISTICS DATA BLOCK
5809 066370  066402                    .WORD   T20BFR          ;ADDRESS OF MESSAGE BUFFER
5810 066372  000000                    .WORD   0
5811 066374  000024                    .WORD   20.             ;LENGTH OF MESSAGE BUFFER
5812 066376  000000                    .WORD   0               ;ESS,ENB,EAI,ERI
5813 066400  000007                    .WORD   7               ;EXTENDED FEATURES UNIT NO.
5814
5815
5816                            ;MESSAGE BUFFER FOR ALL TEST 17 COMMANDS
5817
5818 066402                    T20BFR:                         ;BEGIN MESSAGE BUFFER
5819 066402  000000                    .WORD   0               ;MESSAGE TYPE
5820 066404  000000                    .WORD   0               ;DATA FIELD LENGTH
5821 066406  000000                    .WORD   0               ;RBPCR
5822 066410  000000                    .WORD   0               ;XST0
5823 066412  000000                    .WORD   0               ;XST1
5824 066414  000000                    .WORD   0               ;XST2
5825 066416  000000                    .WORD   0               ;XST3
5826 066420  000000                    .WORD   0               ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
5827 066422                    T20BFSTA: .BLKB 64.             ;READ STATUS AND WRITE FIFO BUFFER
5828 066522                    T20BEND:                        ;END OF MESSAGE BUFFER
5829                            ;
5830                            ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
5831                            ;
5833         066530                    .=<.+10>&177770
5835 066530                    T20PK2:
5836 066530  100006                    .WORD   P.WRTSUB!P.ACK  ;WRITE SUBSYSTEM WITH ACK
5837 066532  066540                    .WORD   T20DT2          ;LOW ADDRESS OF DATA BLOCK
5838 066534  000000                    .WORD   0               ;HIGH ADDRESS OF DATA BLOCK
5839 066536  000012                    .WORD   10.             ;MINIMUM MESSAGE PACKET SIZE
5840
5841 066540                    T20DT2:                         ;DATA BLOCK
5842 066540      000                   .BYTE   0               ;BSEL0
5843 066541      000                   .BYTE   0               ;BSEL1
5844 066542  000000                    .WORD   0               ;SEL2
5845 066544                            .BLKB   64.             ;WRITE FIFO DATA OUTPUT BUFFER
5846
5847
5848 066644                            ENDTST
     066644                                                    L10073:
     066644  104401                                                    TRAP    CSETST
```

```
                        .SBTTL  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 7.
;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     INITIATE TRANSPORT SERVO EXERCISER
;         6     PRINT EXTENDED TRANSPORT STATUS
;         7     EXIT (RETURN TO SUPERVISOR)
;
;EACH MENU ITEM CORRESPONDS TO A SUBTEST, AS FOLLOWS:
;
;
;PRINTS OUT THE MENU ON THE CONSOLE TERMINAL.
;
;
;CAUSES ALL THREE LED INDICATORS ON THE M7196 MODULE
;TO BE ILLUMINATED. AFTER INITIATING THIS ROUTINE, THE OPERATOR
;SHOULD OBSERVE THE LED'S AND VERIFY THAT THEY ARE INDEED ALL LIT.
;THIS ROUTINE FIRST USES THE WRITE SUBSYSTEM MEMORY COMMAND TO
;SET THE FORCE WRONG PARITY FLIP-FLOP, WHICH SERVES TO DRIVE THE
;"PROCESSOR NOT OK" LED. THEN IT ENTERS A LOOP THAT CONTINUALLY
;WRITES THE LOW BYTE OF TSDB AND READS THE TSSR. THESE LATTER TWO
;OPERATIONS WILL CAUSE THE "NOT SSR" AND "DRIVING BUS" LED'S TO
;GLOW -- THEY ARE NOT REALLY LIT AT ALL TIMES BUT SHOULD APPEAR
;REASONABLY VISIBLE.
;
;
;INITIALIZES THE CONTROLLER TO CAUSE ALL LED'S TO
;EXTINGUISH.
;
;
;
;
;THIS ROUTINE INITIALIZES THE CONTROLLER, ISSUES A
;WRITE CHARACTERISTICS COMMAND TO ENABLE ATTENTION INTERRUPTS,
;ISSUES A MESSAGE BUFFER RELEASE COMMAND, PRINTS A MESSAGE ON THE
;CONSOLE TERMINAL INSTRUCTING THE OPERATOR TO TOGGLE THE ON-LINE
;SWITCH ON THE TRANSPORT, THEN WAITS FOR AN ATTENTION INTERRUPT.
;EACH TIME THE TRANSPORT TRANSITIONS FROM ON-LINE TO OFF-LINE OR
;VICE-VERSA, AN ATTENTION INTERRUPT SHOULD BE GENERATED. THE PROGRAM
;WILL REPORT THE INTERRUPT AND THE CURRENT STATE OF THE TRANSPORT.
```

```
                                ;THE OPERATOR SHOULD VERIFY THAT THE REPORTED STATE MATCHES THE
                                ;STATE INDICATED BY THE LED ON THE FRONT PANEL OF THE TRANSPORT.
                                ;IN ADDITION, WHEN THE TRANSPORT IS PLACED OFF-LINE, THE PROGRAM
                                ;ISSUES A SEQUENCE OF TAPE-MOTION COMMANDS (READ, WRITE, POSITION, ETC.
                                ;AND VERIFIES THAT, FOR EACH COMMAND, FUNCTION REJECT TERMINATION
                                ;RESULTS, ALONG WITH THE NON-EXECUTABLE FUNCTION (NEF) ERROR BIT BEING
                                ;SET.
                                ;
                                ;THIS ROUTINE INSTRUCTS THE OPERATOR TO MOUNT A SCRATCH
                                ;TAPE REEL THAT DOES NOT HAVE A WRITE-ENABLE RING INSTALLED, THEN
                                ;WAITS FOR THE OPERATOR TO RESPOND THAT THIS HAS BEEN ACCOMPLISHED.
                                ;UPON THE RESPONSE, THE PROGRAM VERIFIES THAT THE TRANSPORT SHOWS
                                ;A WRITE-PROTECTED STATUS, THEN ATTEMPTS TO WRITE DATA ON THE
                                ;TAPE AND EXPECTS THE APPROPRIATE ERROR TERMINATION INDICATING THAT
                                ;THE WRITE FUNCTION COULD NOT BE PERFORMED BECAUSE THE REEL IS
                                ;WRITE-PROTECTED.  IF THE APPROPRIATE TERMINATION IS NOT RECEIVED,
                                ;AN ERROR IS REPORTED.
                                ;
                                ;
                                ;
                                ;INSTRUCTS THE OPERATOR TO PLACE THE TAPE TRANSPORT(S)
                                ;ON-LINE (IF ANY ARE OFF-LINE) THEN ATTEMPTS TO PERFORM AN EXTENDED
                                ;STATUS READOUT.  FOR EACH TRANSPORT EQUIPPED WITH THIS FEATURE,
                                ;THE PROGRAM FORMATS AND PRINTS OUT THE RESULTING STATUS. IF THE
                                ;TRANSPORT IS NOT EQUIPPED WITH THIS FEATURE, A MESSAGE INDICATING
                                ;SUCH IS ISSUED.
                                ;
                                ;
                                ;
                                ;
066646                          BGNTST
066646                                                          T10::
066646                          RFLAGS  R0              ;GET OPERATOR FLAGS
066646  104421                                                  TRAP    C$RFLA
066650  001403                  BEQ     21$             ;BR, IF OK TO RUN
066652  012700  072230          MOV     #T38NE,R0       ;"TEST NOT EXECUTED"
066656  000402                  BR      3$              ;JUMP IF NOT FIRST TEST
066660                  21$:
066660  012700  073345          MOV     #T38ID,R0       ;TEST ID MESSAGE
066664  004737  016510          3$:     JSR     PC,TSTSETUP     ;DO THE COMMON SETUP
066670  004737  020500          JSR     PC,CHKMAN       ;IS MANUAL INTERVENTION ALLOWED?
066674  103402                  BCS     22$             ;BR, IF MANUAL INTER ALLOWED
066676  000137  071430          JMP     64$             ;JUMP IF NOT ALLOWED
066702                  22$:
066702  005037  002222          2$:     CLR     FATFLG          ;CLEAR THE FATAL ERROR FLAG
066706  012737  176750  071442  MOV     #65000.,T38DLY  ;SET UP DELAY COUNTER
066714  004737  015774          5$:     JSR     PC,SOFINIT      ;DO A SOFT INIT
066720  103427                  BCS     23$             ;BRANCH IF OK
066722  010001                  MOV     R0,R1           ;CONTENTS OF TSSR REGISTER
066724  032701  000200          BIT     #SSR,R1         ;CHECK FOR TSSR SET
066730  001023                  BNE     23$             ;KEEP GOING IF NOT SET
066732                          DELAY   250             ;CALL DELAY ROUTINE
066732  012727  000250                                          MOV     #250,(PC)+
066736  000000                                                  .WORD   0
066740  013727  002116                                          MOV     L$DLY,(PC)+
066744  000000                                                  .WORD   0
066746  005367  177772                                          DEC     -6(PC)
```

```
                066752  001375                                                          BNE     .-4
                066754  005367  177756                                                  DEC     -22(PC)
                066760  001367                                                          BNE     .-20
     5964       066762  005337  071442          DEC     T38DLY          ;BUMP COUNTER DOWN
     5965       066766  001352                  BNE     5$              ;BR, IF MORE TIME LEFT
     5966       066770                          ERRDF   ERRNO,SFIERR,SFIMSG     ;REPORT FATAL ERROR
                066770  104455                                                          TRAP    C$ERDF
                066772  001751                                                          .WORD   1001
                066774  003652                                                          .WORD   SFIERR
                066776  012034                                                          .WORD   SFIMSG
     5967       067000  012700  073372  23$:    MOV     #MIMENU,R0      ;MENU OF MANUAL INTERVENTIONS
     5968       067004  012701  000006          MOV     #6,R1           ;MAXIMUM ALLOWED SELECTION
     5969       067010  004737  020256          JSR     PC,GETSEL       ;GO GET THE OPERATORS SELECTION
     5970       067014  010004                  MOV     R0,R4           ;GET NUMBER FROM ROUTINE
     5971       067016  006304                  ASL     R4              ;CONVERT TO WORD OFFSET
     5972       067020  000174  067024          JMP     @6$(R4)         ;JUMP TO PROPER LOOP
     5973       067024  066702          6$:     .WORD   2$              ;RETYPE THE MENU
     5974       067026  067042                  .WORD   10$             ; 1 TURN ON LED'S
     5975       067030  067324                  .WORD   15$             ; 2 TURN OFF LED'S
     5976       067032  067556                  .WORD   20$             ; 3 ONLINE ATTENTION
     5977       067034  070212                  .WORD   25$             ; 4 WRITE PROTECT
     5978       067036  071146                  .WORD   35$             ; 5 EXTENDED TRANSPORT STATUS
     5979       067040  071424                  .WORD   63$             ; 6 LEAVE THE TEST
     5980       067042                  10$:    PRINTF  #T38MS2         ;TELL OPERATOR TO CNTRL-C FOR EXIT
                067042  012746  073241                                                  MOV     #T38MS2,-(SP)
                067046  012746  000001                                                  MOV     #1,-(SP)
                067052  010600                                                          MOV     SP,R0
                067054  104417                                                          TRAP    C$PNTF
                067056  062706  000004                                                  ADD     #4,SP
     5981       067062  004737  073776          JSR     PC,T38REST      ;SET PACKET TO INITIAL VALUES
     5982       067066  004737  015774          JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
     5983       067072  103405                  BCS     100$            ;BR IF SOFT INIT = OK
     5987       067074  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
     5988       067076                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
                067076  104455                                                          TRAP    C$ERDF
                067100  001752                                                          .WORD   1002
                067102  003652                                                          .WORD   SFIERR
                067104  012034                                                          .WORD   SFIMSG
     5989       067106  013737  002202  072170  100$:   MOV     UNITN,T38DSW    ;SET UNIT NUMBER
     5990
     5991       067114  012704  072150          MOV     #T38PK2,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
     5992       067120  004737  010662          JSR     PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
     5993       067124  103405                  BCS     110$            ;BR, IF COMMAND ISSUED OK
     5997       067126  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
     5998       067130                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
                067130  104456                                                          TRAP    C$ERHRD
                067132  001753                                                          .WORD   1003
                067134  005056                                                          .WORD   WRTMSG
                067136  012034                                                          .WORD   SFIMSG
     5999       067140                  110$:
     6000       067140  112737  000000  071461          MOVB    #0,T38BS1       ;CLEAR BIT #4
     6001       067146  112737  000011  071460          MOVB    #11,T38BS0      ;WRITE MISC COMMAND
     6002       067154  012704  071450          MOV     #T38PACKET,R4   ;SET UP NEW WRT. SUBSYS MEM. COMMAND
     6003                                  ;
     6004                                  ;NOTE: THIS COMMAND TURNS ON THE PROCESSOR FAIL LED
     6005                                  ;
     6006       067160  010465  000000          MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
```

```
6007 067164  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6008 067170  103405                  BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
6009 067172  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6013 067174                          ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     067174  104455                                                          TRAP    C$ERDF
     067176  001754                                                          .WORD   1004
     067200  072646                                                          .WORD   T38SSR
     067202  012046                                                          .WORD   PKTSSR
6014 067204                  150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     067204  104406                                                          TRAP    C$CLP1
6015 067206                          SETPRI  #PRI07                  ;RAISE THE PRIORITY
     067206  012700  000340                                                  MOV     #PRI07,R0
     067212  104441                                                          TRAP    C$SPRI
6016 067214  005037  071434          CLR     TTION2                  ;ASSUME INTERRUPTS ARE ENABLED
6017 067220  032737  000100  177560  BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
6018 067226  001005                  BNE     701$                    ;BRANCH IF YES
6019 067230  005237  071434          INC     TTION2                  ;FLAG SET IF INTERRUPTS OFF
6020 067234  052737  000100  177560  BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
6021 067242  012701  000060  701$:   MOV     #TTIVEC,R1              ;START OF TTI VECTORS
6022 067246  011137  071436          MOV     (R1),IVSAV2             ;SAVE THE CURRENT TTI VECTOR
6023 067252  012721  070730          MOV     #590$,(R1)+             ;SET NEW INTERRUPT ROUTINE
6024 067256  011137  071440          MOV     (R1),IPSAV2             ;SAVE THE VECTOR PRIORITY
6025 067262  012711  000340          MOV     #PRI07,(R1)             ;USE PRIORITY SEVEN
6026 067266                          SETPRI  #PRI00                  ;LOWER INTERRUPT BR LEVEL
     067266  012700  000000                                                  MOV     #PRI00,R0
     067272  104441                                                          TRAP    C$SPRI
6027 067274  012701  177777          MOV     #-1,R1                  ;DATA TO WRITE TO TSDB
6028 067300  000240          12$:    NOP                             ;ALLOW OPERATOR TO TYPE ^C
6029 067302  012702  001750          MOV     #1000.,R2               ;SET-UP INNER LOOP
6030 067306  110165  000000  14$:    MOVB    R1,TSDB(R5)             ;WRITE DATA TO TSDB
6031 067312  016500  000002          MOV     TSSR(R5),R0             ;READ TSSR
6032 067316  005302                  DEC     R2                      ;REDUCE INNER COUNT
6033 067320  001372                  BNE     14$                     ;LOOP TILL EXPIRES
6034 067322  000766                  BR      12$                     ;LOOP UNTIL HALTED
6035
6036 067324                  15$:    PRINTF  #T38MS2                 ;TYPE CNTL C TO EXIT
     067324  012746  073241                                                  MOV     #T38MS2,-(SP)
     067330  012746  000001                                                  MOV     #1,-(SP)
     067334  010600                                                          MOV     SP,R0
     067336  104417                                                          TRAP    C$PNTF
     067340  062706  000004                                                  ADD     #4,SP
6037 067344  004737  015774          JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
6038 067350  103405                  BCS     200$                    ;BR IF SOFT INIT = OK
6042 067352  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6043 067354                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     067354  104455                                                          TRAP    C$ERDF
     067356  001755                                                          .WORD   1005
     067360  003652                                                          .WORD   SFIERR
     067362  012034                                                          .WORD   SFIMSG
6044 067364                  200$:
6045 067364  013737  002202  072170  MOV     UNITN,T38DSW            ;SET UNIT NUMBER
6046 067372  012704  072150          MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
6047 067376  004737  010662          JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
6048 067402  103405                  BCS     210$                    ;BR, IF COMMAND ISSUED OK
6052 067404  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6053 067406                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     067406  104456                                                          TRAP    C$ERHRD
```

```
                067410  001756                                                          .WORD   1006
                067412  005056                                                          .WORD   WRTMSG
                067414  012034                                                          .WORD   SFIMSG
 6054                                   ;*****************************************************************
 6055                                   ;
 6056                                   ;       THIS WRITE SUB-SYSTEM MEMORY COMMAND JUST HOLDS THE LEDS OFF
 6057                                   ;
 6058                                   ;*****************************************************************
 6059 067416                            210$:
 6060 067416  112737  000000  071461            MOVB    #0,T38BS1               ;CLEAR BIT #4
 6061 067424  112737  000025  071460            MOVB    #25,T38BSO              ;STOP DRIVE TEST 22
 6062 067432  012704  071450                    MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
 6063 067436  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 6064 067442  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 6065 067446  103405                            BCS     250$                    ;BR IF CARRY SET (GOOD RETURN)
 6066 067450  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6070 067452                                    ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                067452  104455                                                          TRAP    C$ERDF
                067454  001757                                                          .WORD   1007
                067456  072646                                                          .WORD   T38SSR
                067460  012046                                                          .WORD   PKTSSR
 6071 067462                            250$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                067462  104406                                                          TRAP    C$CLP1
 6072 067464                                    SETPRI  #PRI07                  ;RAISE THE PRIORITY
                067464  012700  000340                                                  MOV     #PRI07,R0
                067470  104441                                                          TRAP    C$SPRI
 6073 067472  005037  071434                    CLR     TTION2                  ;ASSUME INTERRUPTS ARE ENABLED
 6074 067476  032737  000100  177560            BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
 6075 067504  001005                            BNE     710$                    ;BRANCH IF YES
 6076 067506  005237  071434                    INC     TTION2                  ;FLAG SET IF INTERRUPTS OFF
 6077 067512  052737  000100  177560            BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
 6078 067520  012701  000060            710$:   MOV     #TTIVEC,R1              ;START OF TTI VECTORS
 6079 067524  011137  071436                    MOV     (R1),TVSAV2             ;SAVE THE CURRENT TTI VECTOR
 6080 067530  012721  070730                    MOV     #590$,(R1)+             ;SET NEW INTERRUPT ROUTINE
 6081 067534  011137  071440                    MOV     (R1),TPSAV2             ;SAVE THE VECTOR PRIORITY
 6082 067540  012711  000340                    MOV     #PRI07,(R1)             ;USE PRIORITY SEVEN
 6083 067544                                    SETPRI  #PRI00                  ;LOWER INTERRUPT BR LEVEL
                067544  012700  000000                                                  MOV     #PRI00,R0
                067550  104441                                                          TRAP    C$SPRI
 6084 067552  000240            260$:   NOP                                     ;ALLOW CNTL C
 6085 067554  000776                    BR      260$                            ;LOOP UNTIL STOPPED
 6086
 6087
 6088 067556                            20$:    PRINTF  #T38MS2                 ;TELL'EM WHAT TO TYPE
                067556  012746  073241                                                  MOV     #T38MS2,-(SP)
                067562  012746  000001                                                  MOV     #1,-(SP)
                067566  010600                                                          MOV     SP,R0
                067570  104417                                                          TRAP    C$PNTF
                067572  062706  000004                                                  ADD     #4,SP
 6089 067576                                    SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
                067576  012700  000000                                                  MOV     #PRI00,R0
                067602  104441                                                          TRAP    C$SPRI
 6090 067604  005037  002224                    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 6091 067610  004737  015774                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 6092 067614  103405                            BCS     300$                    ;BR IF SOFT INIT = OK
 6096 067616  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6097 067620                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
```

```
                067620  104455                                                          TRAP    C$ERDF
                067622  001760                                                          .WORD   1008
                067624  003652                                                          .WORD   SFIERR
                067626  012034                                                          .WORD   SFIMSG
 6098           067630                  300$:
 6099           067630  013737  002202  072170          MOV     UNITN,T38DSW            ;SET UNIT NUMBER IN PACKET
 6100           067636  012737  000040  072166          MOV     #BIT5,T38EAI            ;ENABLE ATTENTION INTERRUPTS
 6101           067644  012704  072150                  MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
 6102           067650  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 6103           067654  103405                          BCS     310$                    ;BR, IF COMMAND ISSUED OK
 6107           067656  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6108           067660                          ERRHRD  ERRNO,WRTMSG,SFIMSG             ;WRITE CHARACTERISTISC FAILED
                067660  104456                                                          TRAP    C$ERHRD
                067662  001761                                                          .WORD   1009
                067664  005056                                                          .WORD   WRTMSG
                067666  012034                                                          .WORD   SFIMSG
 6109           067670                  310$:
 6110           067670  012704  072200                  MOV     #T38PK3,R4              ;SET UP NEW PACKET FOR MESS BUF REL
 6111           067674  010465  000000                  MOV     R4,TSDB(R5)             ;MESSAGE BUFFER RELEASE,ACK,CVC=1 CMD
 6112           067700  004737  016250                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 6113           067704  005002                          CLR     R2                      ;MAKE SURE ALL IS CLEAR
 6114           067706  016501  000002                  MOV     TSSR(R5),R1             ;GET TSSR STATUS
 6115           067712  032701  000100                  BIT     #OFL,R1                 ;IS OFL SET
 6116           067716  001402                          BEQ     320$                    ;BR, IF OFL IS NOT SET
 6117           067720  052702  000100                  BIS     #OFL,R2                 ;SET OFL IN EXPECTED
 6118           067724  052702  000200          320$:   BIS     #SSR,R2                 ;SET UP EXPECTED
 6119           067730  020201                          CMP     R2,R1                   ;IS EVERYTHING OK
 6120           067732  001404                          BEQ     350$                    ;BR, IF ALL IS WELL
 6124           067734                          ERRHRD  ERRNO,T38SST,PKTSSR             ;DEVICE FATAL SSR FAILED TO SET
                067734  104456                                                          TRAP    C$ERHRD
                067736  001762                                                          .WORD   1010
                067740  073056                                                          .WORD   T38SST
                067742  012046                                                          .WORD   PKTSSR
 6125           067744                  350$:   CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
                067744  104406                                                          TRAP    C$CLP1
 6126           067746                          PRINTF  #T38MS1                         ;TELL OPERATOR TO TOGGLE SWITCH
                067746  012746  073146                                                  MOV     #T38MS1,-(SP)
                067752  012746  000001                                                  MOV     #1,-(SP)
                067756  010600                                                          MOV     SP,R0
                067760  104417                                                          TRAP    C$PNTF
                067762  062706  000004                                                  ADD     #4,SP
 6127           067766                          PRINTF  #T38MS2                         ;TELL OPERATOR TO DO ^C TO EXIT
                067766  012746  073241                                                  MOV     #T38MS2,-(SP)
                067772  012746  000001                                                  MOV     #1,-(SP)
                067776  010600                                                          MOV     SP,R0
                070000  104417                                                          TRAP    C$PNTF
                070002  062706  000004                                                  ADD     #4,SP
 6128           070006                          SETPRI  #PRI07                          ;RAISE THE PRIORITY
                070006  012700  000340                                                  MOV     #PRI07,R0
                070012  104441                                                          TRAP    C$SPRI
 6129           070014  005037  071434                  CLR     TTION2                  ;ASSUME INTERRUPTS ARE ENABLED
 6130           070020  032737  000100  177560          BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
 6131           070026  001005                          BNE     720$                    ;BRANCH IF YES
 6132           070030  005237  071434                  INC     TTION2                  ;FLAG SET IF INTERRUPTS OFF
 6133           070034  052737  000100  177560          BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
 6134           070042  012701  000060          720$:   MOV     #TTIVEC,R1              ;START OF TTI VECTORS
 6135           070046  011137  071436                  MOV     (R1),TVSAV2             ;SAVE THE CURRENT TTI VECTOR
```

```
6136 070052 012721 070730          MOV     #590$,(R1)+     ;SET NEW INTERRUPT ROUTINE
6137 070056 011137 071440          MOV     (R1),TPSAV2     ;SAVE THE VECTOR PRIORITY
6138 070062 012711 000340          MOV     #PRI07,(R1)     ;USE PRIORITY SEVEN
6139 070066                        SETPRI  #PRI00          ;LOWER INTERRUPT BR LEVEL
     070066 012700 000000                                          MOV     #PRI00,R0
     070072 104441                                                 TRAP    C$SPRI
6140 070074 000240           360$: NOP                     ;ALLOW CONTROL C
6141 070076 005737 002224          TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
6142 070102 001001                 BNE     370$            ;BRANCH IF YES
6143 070104 000773                 BR      360$            ;WAIT SOME MORE FOR INTERRUPT
6144 070106                  370$: PRINTF  #T38INT         ;"INTERRUPT RECEIVED"
     070106 012746 072736                                          MOV     #T38INT,-(SP)
     070112 012746 000001                                          MOV     #1,-(SP)
     070116 010600                                                 MOV     SP,R0
     070120 104417                                                 TRAP    C$PNTF
     070122 062706 000004                                          ADD     #4,SP
6145 070126 016501 000002          MOV     TSSR(R5),R1     ;READ TSSR STATUS
6146 070132 032701 000100          BIT     #OFL,R1         ;CHECK THE OFF-LINE BIT
6147 070136 001011                 BNE     380$            ;BR, IF DRIVE IS OFF-LINE
6148 070140                        PRINTF  #T380NL         ;"DRIVE IS NOW ON-LINE"
     070140 012746 072766                                          MOV     #T380NL,-(SP)
     070144 012746 000001                                          MOV     #1,-(SP)
     070150 010600                                                 MOV     SP,R0
     070152 104417                                                 TRAP    C$PNTF
     070154 062706 000004                                          ADD     #4,SP
6149 070160 000410                 BR      390$            ;ALMOST DONE
6150 070162                  380$: PRINTF  #T380FL         ;"DRIVE IS NOW OFF-LINE"
     070162 012746 073022                                          MOV     #T380FL,-(SP)
     070166 012746 000001                                          MOV     #1,-(SP)
     070172 010600                                                 MOV     SP,R0
     070174 104417                                                 TRAP    C$PNTF
     070176 062706 000004                                          ADD     #4,SP
6151 070202 005037 002224    390$: CLR     INTRECV         ;CLEAR INTERRUPT FLAG
6152 070206 000137 067630          JMP     300$            ;TRY AGAIN
6153 070212                   25$: GMANIL  T38MSG,T38DAT,-1,NO  ;WAIT FOR OPERATOR TO MOUNT TAPE
     070212 104443                                                 TRAP    C$GMAN
     070214 000404                                                 BR      10000$
     070216 073774                                                 .WORD   T38DAT
     070220 000120                                                 .WORD   T$CODE
     070222 073305                                                 .WORD   T38MSG
     070224 177777                                                 .WORD   -1
     070226                                                  10000$:
6154 070226                        BNCOMPLETE 25$          ;RETRY IF ERROR
     070226 103371                                                 BCC     25$
6155 070230 005737 073774          TST     T38DAT          ;DID OPERATOR SAY 'YES' ?
6156 070234 001002                 BNE     27$             ;BRANCH IF YES
6157 070236 000137 066702          JMP     2$              ;RETURN TO MAIN MENU
6158 070242                  27$:
6159 070242 004737 015774          JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
6160 070246 103405                 BCS     400$            ;BR IF SOFT INIT = OK
6164 070250 010001                 MOV     R0,R1           ;SAVE CONTENTS OF TSSR
6165 070252                        ERRDF   ERRNO,SFIERR,SFIMSG  ;DEVICE FATAL ERROR DURING INIT
     070252 104455                                                 TRAP    C$ERDF
     070254 001763                                                 .WORD   1011
     070256 003652                                                 .WORD   SFIERR
     070260 012034                                                 .WORD   SFIMSG
6166 070262                  400$: CKLOOP                  ;LOOP IF SELECTED
```

```
                070262  104406                                                          TRAP    C$CLP1
6167            070264  013737  002202  072170          MOV     UNITN,T38DSW            ;SET UNIT NUMBER
6168            070272  012704  072150                  MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
6169            070276  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
6170            070302  103405                          BCS     410$                    ;BR, IF COMMAND ISSUED OK
6174            070304  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6175            070306                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
                070306  104456                                                          TRAP    C$ERHRD
                070310  001764                                                          .WORD   1012
                070312  005056                                                          .WORD   WRTMSG
                070314  012034                                                          .WORD   SFIMSG
6176            070316                  410$:   CKLOOP                                  ;LOOP IF SELECTED
                070316  104406                                                          TRAP    C$CLP1
6177            070320  013701  071474                  MOV     T38BFR+6,R1             ;PICK UP XSTO CONTENTS
6178            070324  010102                          MOV     R1,R2                   ;SET UP EXPECTED
6179            070326  052702  000004                  BIS     #BIT2,R2                ;SET UP THE WRITE LOCKED BIT
6180            070332  020102                          CMP     R1,R2                   ;ARE THEY CORRECT
6181            070334  001406                          BEQ     430$                    ;BR, IF ALL IS WELL (OK)
6185            070336                                  ERRHRD  ERRNO,T38WRL,EXPREC     ;"WRITE LOCKED BIT IS NOT SET ETC."
                070336  104456                                                          TRAP    C$ERHRD
                070340  001765                                                          .WORD   1013
                070342  072464                                                          .WORD   T38WRL
                070344  015474                                                          .WORD   EXPREC
6186            070346  005237  002222                  INC     FATFLG                  ;SET FATAL FLAG
6187            070352                  430$:   CKLOOP                                  ;LOOP IF SELECTED
                070352  104406                                                          TRAP    C$CLP1
6188            070354  005737  002222                  TST     FATFLG                  ;WAS THE DRIVE NOT WRITE LOCKED
6189            070360  001402                          BEQ     435$                    ;BR, IF FLAG NOT SET
6190            070362  000137  066702                  JMP     2$                      ;RE-WRITE MENU
6191            070366  017737  112532  072222  435$:   MOV     @FREE,T38WR             ;SET UP WRITE BUFFER ADDRESS
6192            070374  012704  072220                  MOV     #T38PK4,R4              ;GET PACKET ADDRESS
6193            070400  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
6194            070404  004737  016250                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
6195            070410  016501  000002                  MOV     TSSR(R5),R1             ;GET TSSR
6196            070414  012702  100206                  MOV     #SC!SSR!BIT1!BIT2,R2    ;SET UP EXPECTED
6197            070420  020102                          CMP     R1,R2                   ;ARE THEY EQUAL (CORRECT)
6198            070422  001404                          BEQ     440$                    ;BR, IF CORRECT STATUS
6202            070424                                  ERRHRD  ERRNO,T38WRT,PKTSSR     ;"TSSR INCORRECT AFTER WRITE COMMAND
                070424  104456                                                          TRAP    C$ERHRD
                070426  001766                                                          .WORD   1014
                070430  072400                                                          .WORD   T38WRT
                070432  012046                                                          .WORD   PKTSSR
6203            070434                  440$:   CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
                070434  104406                                                          TRAP    C$CLP1
6204            070436  013701  071474                  MOV     T38BFR+6,R1             ;READ XSTO CONTENTS
6205            070442  010102                          MOV     R1,R2                   ;SET UPR EXPECTED
6206            070444  052702  004000                  BIS     #BIT11,R2               ;SET THE WRITE LOCK ERROR BIT (XSTO)
6207            070450  020102                          CMP     R1,R2                   ;WAS THE BIT SET
6208            070452  001404                          BEQ     450$                    ;BR, IF IT WAS (GOOD)
6212            070454                                  ERRHRD  ERRNO,T38WLE,EXPREC     ;"WRITE LOCK ERROR BIT NOT SET"
                070454  104456                                                          TRAP    C$ERHRD
                070456  001767                                                          .WORD   1015
                070460  072525                                                          .WORD   T38WLE
                070462  015474                                                          .WORD   EXPREC
6213            070464                  450$:   CKLOOP                                  ;LOOP IF SELECTED
                070464  104406                                                          TRAP    C$CLP1
6214            070466  000137  066702                  JMP     2$                      ;GO BACK TO MENU
```

```
                                ;*****************************************************************************
                                ;       SERVO EXERCISER NO LONGER USED
                                ;*****************************************************************************
070472                          30$:
070472                                  PRINTB  #T38MS3                 ;''EXE ANY OTHER MENU SELECTION TO STOP
070472  012746  072305                                                          MOV     #T38MS3,-(SP)
070476  012746  000001                                                          MOV     #1,-(SP)
070502  010600                                                                  MOV     SP,R0
070504  104414                                                                  TRAP    C$PNTB
070506  062706  000004                                                          ADD     #4,SP
070512  004737  073776                  JSR     PC,T38REST              ;SET PACKET TO INITIAL VALUES
070516  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
070522  103405                          BCS     500$                    ;BR IF SOFT INIT = OK
070524  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
070526                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
070526  104455                                                                  TRAP    C$ERDF
070530  001770                                                                  .WORD   1016
070532  003652                                                                  .WORD   SFIERR
070534  012034                                                                  .WORD   SFIMSG
070536  013737  002202  072170  500$:   MOV     UNITN,T38DSW            ;SET UNIT NUMBER
070544  012704  072150                  MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
070550  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
070554  103405                          BCS     510$                    ;BR, IF COMMAND ISSUED OK
070556  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
070560                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
070560  104456                                                                  TRAP    C$ERHRD
070562  001771                                                                  .WORD   1017
070564  005056                                                                  .WORD   WRTMSG
070566  012034                                                                  .WORD   SFIMSG
070570                          510$:
070570  112737  000000  071461          MOVB    #0,T38BS1               ;CLEAR BIT #4
070576  112737  000020  071460          MOVB    #20,T38BS0              ;EXECUTE DRIVE TEST 22
070604  012704  071450                  MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
070610  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
070614  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
070620  103405                          BCS     550$                    ;BR IF CARRY SET (GOOD RETURN)
070622  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
070624                                  ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
070624  104455                                                                  TRAP    C$ERDF
070626  001772                                                                  .WORD   1018
070630  072646                                                                  .WORD   T38SSR
070632  012046                                                                  .WORD   PKTSSR
070634                          550$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
070634  104406                                                                  TRAP    C$CLP1
070636                                  SETPRI  #PRI07                  ;RAISE THE PRIORITY
070636  012700  000340                                                          MOV     #PRI07,R0
070642  104441                                                                  TRAP    C$SPRI
070644  005037  071434                  CLR     TTION2                  ;ASSUME INTERRUPTS ARE ENABLED
070650  032737  000100  177560          BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
070656  001005                          BNE     555$                    ;BRANCH IF YES
070660  005237  071434                  INC     TTION2                  ;FLAG SET IF INTERRUPTS OFF
070664  052737  000100  177560          BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
070672  012701  000060          555$:   MOV     #TTIVEC,R1              ;START OF TTI VECTORS
070676  011137  071436                  MOV     (R1),TVSAV2             ;SAVE THE CURRENT TTI VECTOR
070702  012721  070730                  MOV     #590$,(R1)+             ;SET NEW INTERRUPT ROUTINE
070706  011137  071440                  MOV     (R1),TPSAV2             ;SAVE THE VECTOR PRIORITY
```

```
6261 070712  012711  000340                MOV     #PRI07,(R1)             ;USE PRIORITY SEVEN
6262 070716                                SETPRI  #PRI00                  ;LOWER INTERRUPT BR LEVEL
     070716  012700  000000                                                     MOV     #PRI00,R0
     070722  104441                                                             TRAP    C$SPRI
6263 070724  000240                560$:   NOP                             ;LOOP AWHILE
6264 070726  000776                        BR      560$                    ;STAY IN ''TIGHT'' LOOP
6265                               ;+
6266                               ;PROCESS CONSOLE INTERRUPTS
6267                               ;-
6268
6269 070730  010046                590$:   MOV     R0,-(SP)                ;SAVE WORK REGISTER
6270 070732  113700  177562                MOVB    @#TTIBFR,R0             ;GET THE OPERATOR INPUT
6271 070736  042700  000200                BIC     #200,R0                 ;STRIP OFF PARITY BIT
6272 070742  122700  000015                CMPB    #15,R0                  ;IS IT A CARRIAGE RETURN ?
6273 070746  001075                        BNE     591$                    ;JUST EXIT IF NOT
6274 070750  012766  066702  000002        MOV     #2$,2(SP)               ;RETURN TO MASTER MENU
6275 070756  005066  000004                CLR     4(SP)                   ;FORCE PRIORTY 0
6276 070762  013737  071436  000060        MOV     TVSAV2,@#TTIVEC ;RESTORE VECTOR
6277 070770  013737  071440  000062        MOV     TPSAV2,@#TTIVEC+2       ;RESTORE SUPER PRIORTY
6278 070776  112737  000025  071460        MOVB    #25,T38BS0              ;STOP DRIVE TEST 22
6279 071004  112737  000000  071461        MOVB    #0,T38BS1               ;CLEAR BS1
6280 071012  012704  071450                MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
6281 071016  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
6282 071022  012737  176750  071442        MOV     #65000.,T38DLY          ;SET UP DELAY COUNTER
6283 071030  004737  016250        592$:   JSR     PC,WAITF                ;DO A WAIT FOR SSR
6284 071034  016501  000002                MOV     TSSR(R5),R1             ;CONTENTS OF TSSR REGISTER
6285 071040  032701  000200                BIT     #SSR,R1                 ;CHECK FOR TSSR SET
6286 071044  001017                        BNE     595$                    ;KEEP GOING IF NOT SET
6287 071046                                DELAY   250                     ;CALL DELAY ROUTINE
     071046  012727  000250                                                     MOV     #250.,(PC)+
     071052  000000                                                             .WORD   0
     071054  013727  002116                                                     MOV     L$DLY,(PC)+
     071060  000000                                                             .WORD   0
     071062  005367  177772                                                     DEC     -6(PC)
     071066  001375                                                             BNE     .-4
     071070  005367  177756                                                     DEC     -22(PC)
     071074  001367                                                             BNE     .-20
6288 071076  005337  071442                DEC     T38DLY                  ;BUMP COUNTER DOWN
6289 071102  001352                        BNE     592$                    ;BR, IF MORE TIME LEFT
6290 071104  004737  016336        595$:   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6291 071110  103405                        BCS     580$                    ;BR IF CARRY SET (GOOD RETURN)
6292 071112  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6296 071114                                ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     071114  104455                                                             TRAP    C$ERDF
     071116  001773                                                             .WORD   1019
     071120  072646                                                             .WORD   T38SSR
     071122  012046                                                             .WORD   PKTSSR
6297 071124                        580$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     071124  104406                                                             TRAP    C$CLP1
6298 071126  005737  071434                TST     TTION2                  ;ARE SUPER INTERRUPTS ENABLED
6299 071132  001403                        BEQ     591$                    ;BR, IF YES
6300 071134  042737  000100  177560        BIC     #100,@#TTICSR           ;RESTORE REGISTER
6301 071142  012600                591$:   MOV     (SP)+,R0                ;RESTORE REGISTER
6302 071144  000002                        RTI                             ;RETURN
6303 071146                        35$:
6304 071146  004737  073776                JSR     PC,T38REST              ;SET PACKET TO INITIAL VALUES
6305 071152  004737  015774                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
```

```
6306 071156  103405                    BCS     600$                    ;BR IF SOFT INIT = OK
6310 071160  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6311 071162                            ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     071162  104455                                                            TRAP    C$ERDF
     071164  001774                                                            .WORD   1020
     071166  003652                                                            .WORD   SFIERR
     071170  012034                                                            .WORD   SFIMSG
6312 071172                    600$:   CKLOOP                          ;LOOP IF SELECTED
     071172  104406                                                            TRAP    C$CLP1
6313 071174  012701  071466            MOV     #T38BFR,R1              ;ADDRESS OF MESSAGE BUFFER
6314 071200  012702  125252            MOV     #125252,R2              ;ALTERNATING 1'S AND 0'S
6315
6316 071204  010221            601$:   MOV     R2,(R1)+                ;CLEAR OUT THE MESSAGE BUFFER
6317 071206  022701  072142            CMP     #T38EB,R1               ;END OF BUFFER YET
6318 071212  001401                    BEQ     605$                    ;BR, IF AT END OF BUFFER
6319 071214  000773                    BR      601$                    ;NOT AT END KEEP GOING
6320 071216  013737  002202  072170 605$:   MOV     UNITN,T38DSW            ;SET UNIT NUMBER
6321 071224  012704  072150            MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
6322 071230  004737  010662            JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
6323 071234  103405                    BCS     610$                    ;BR, IF COMMAND ISSUED OK
6327 071236  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6328 071240                            ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     071240  104456                                                            TRAP    C$ERHRD
     071242  001775                                                            .WORD   1021
     071244  005056                                                            .WORD   WRTMSG
     071246  012034                                                            .WORD   SFIMSG
6329 071250                    610$:   CKLOOP                          ;LOOP IF SELECTED
     071250  104406                                                            TRAP    C$CLP1
6330 071252  112737  000000  071461        MOVB    #0,T38BS1               ;CLEAR BIT #4
6331 071260  112737  000024  071460        MOVB    #24,T38BS0              ;READ EXTENDED DRIVE STATUS
6332 071266  012704  071450            MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
6333 071272  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
6334 071276  012737  000144  071442        MOV     #100.,T38DLY            ;SET UP DELAY ROUTINE
6335 071304  004737  016250        620$:   JSR     PC,WAITF                ;WAIT AWHILE FOR SSR TO SET
6336 071310  016501  000002            MOV     TSSR(R5),R1             ;SEE IF IT REALLY DID
6337 071314  032701  000200            BIT     #SSR,R1                 ;JUST CHECK THAT BIT
6338 071320  001017                    BNE     630$                    ;BR, IF SSR IS SET
6339 071322                            DELAY   250                     ;DELAY ABOUT .25 SEC
     071322  012727  000250                                                    MOV     #250,(PC)+
     071326  000000                                                            .WORD   0
     071330  013727  002116                                                    MOV     L$DLY,(PC)+
     071334  000000                                                            .WORD   0
     071336  005367  177772                                                    DEC     -6(PC)
     071342  001375                                                            BNE     .-4
     071344  005367  177756                                                    DEC     -22(PC)
     071350  001367                                                            BNE     .-20
6340 071352  005337  071442            DEC     T38DLY                  ;START DELAY COUNT DOWN
6341 071356  001352                    BNE     620$                    ;BR, IF COUNTER IS NOT AT DONE
6342 071360  004737  016336        630$:   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6343 071364  103405                    BCS     650$                    ;BR IF CARRY SET (GOOD RETURN)
6344 071366  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6348 071370                            ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     071370  104455                                                            TRAP    C$ERDF
     071372  001776                                                            .WORD   1022
     071374  072646                                                            .WORD   T38SSR
     071376  012046                                                            .WORD   PKTSSR
6349 071400                    650$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

```
        071400  104406                                                          TRAP    C$CLP1
 6350   071402  012700  071506          MOV     #T38BFR+20,R0           ;MESSAGE BUFFER ADDRESS
 6351   071406  005001                  CLR     R1                      ;NO HIGH ORDER ADDRESS BITS
 6352   071410  005037  003134          CLR     KTENABLE                ;NO KT11 STUFF EITHER
 6353   071414  004737  074034          JSR     PC,T38MBP               ;GO PRINT MESSAGE BUFFER CONTENTS
 6354   071420  000137  066702          JMP     2$                      ;GO BACK TO MENU
 6355
 6356
 6357   071424  000137  000200  63$:    JMP     200                     ;REALLY RETURN TO THE SUPERVISOR
 6358   071430                  64$:    EXIT    TST                     ;LEAVE TEST
        071430  104432                                                          TRAP    C$EXIT
        071432  003054                                                          .WORD   L10075-.
 6359
 6360                                   ;+
 6361                                   ;LOCAL TEXT MESSAGES FOR TEST
 6362                                   ;-
 6363
 6364                                   ;LOCAL STORAGE FOR THIS TEST
 6365                                   ;-
 6366                                   ;+
 6367                                   ;LOCAL STORAGE FOR THIS TEST
 6368                                   ;-
 6369
 6370   071434  000000          TTION2:         .WORD   0               ;WORD SET IF SUPERVISOR TTI INTER OFF
 6371   071436  000000          TVSAV2: .WORD   0               ;SAVE TTI VECTOR
 6372   071440  000000          TPSAV2: .WORD   0               ;SAVE TTI PRIORITY
 6373
 6374   071442  000000          T38DLY: .WORD   0                       ;DELAY COUNTER FOR TEST
 6376           071450                  .=<.+10>&177770
 6378   071450                  T38PACKET:                              ;COMMAND PACKET FOR TEST
 6379   071450  140006                  .WORD   140006                  ;WRITE SUBSYSTEM MEM. CMD.  ACK,CVC=1
 6380   071452  071460                  .WORD   T38TAD                  ;ADDRESS OF CHARACTERISTICS BLOCK
 6381   071454  000000                  .WORD   0
 6382   071456  000012                  .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE
 6383   071460                  T38TAD:                         ;CHARACTERISTICS DATA BLOCK
 6384   071460     000          T38BS0: .BYTE   0                       ;BSEL0 BYTE
 6385   071461     000          T38BS1: .BYTE   0                       ;BSEL1 BYTE
 6386   071462  000000          T38BS2: .WORD   0                       ;BSEL1 WORD
 6387   071464  000000                  .WORD   0                       ;DATA
 6388   071466                  T38BFR: .BLKW   150.                    ;MESSAGE BUFFER
 6389   072142  000000          T38EB:  .WORD                           ;END OF BUFFER ADDRESS
 6390
 6391
 6393           072150                  .=<.+10>&177770
 6395   072150                  T38PK2:                                 ;COMMAND PACKET FOR TEST
 6396   072150  140004                  .WORD   140004                  ;WRITE CHARA. MEM. CMND.,  ACK,CVC=1
 6397   072152  072160                  .WORD   T38DTA                  ;ADDRESS OF SELECT DATA BLOCK
 6398   072154  000000                  .WORD   0
 6399   072156  000012                  .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE
 6400
 6401
 6402   072160                  T38DTA:                                 ;SELECT DATA BLOCK
 6403   072160  071466                  .WORD   T38BFR                  ;ADDRESS OF MESSAGE BUFFER
 6404   072162  000000                  .WORD   0
 6405   072164  000400                  .WORD   256.                    ;LENGTH OF MESSAGE BUFFER
 6406   072166  000000          T38EAI: .WORD   0                       ;EAI BIT WORD
 6407   072170  000000          T38DSW: .WORD   0                       ;DRIVE SELECT WORD ETC
```

```
 6409            072200                          .=<.+10>&177770
 6411 072200     140212          T38PK3: .WORD   140212          ;MESSAGE BUFFER RELEASE COMMAND
 6412 072202     000000                  .WORD   0               ;NOT USED
 6413 072204     000000                  .WORD   0               ;NOT USED
 6414 072206     000000                  .WORD   0               ;NOT USED
 6415 072210     000000                  .WORD   0               ;NOT USED
 6416                                    ;
 6417                                    ;WRITE TAPE PACKET
 6418                                    ;
 6420            072220                          .=<.+10>&177770
 6422 072220     140005          T38PK4: .WORD   140005          ;WRITE, ACK, CVC=1 COMMAND
 6423 072222     000000          T38WR:  .WORD   0               ;ADDRESS OF WRITE BUFFER
 6424 072224     000000                  .WORD   0               ;MORE ADDRESS OF WRITE BUFFER
 6425 072226     000400          T38SIZ: .WORD   256.            ;SIZE OF RECORD
 6426
 6427
 6428
 6429
 6430                                    ;+
 6431                                    ;LOCAL TEXT MESSAGES FOR TEST
 6432                                    ;-
 6433
 6434
 6435
 6436
 6437
 6438 072230     123     164     141    T38NE:  .ASCIZ  'Stand-alone Manual Intervention Not Executed'
 6439 072305     045     116     045    T38MS3: .ASCIZ  '%N%A Type <RETURN> To Stop Servo Exerciser, Return To Menu'
 6440 072400     124     123     123    T38WRT: .ASCIZ  'TSSR Not Correct After WRITE, With WRITE PROTECT On'
 6441 072464     127     122     111    T38WRL: .ASCIZ  'WRITE LOCKED Bit Not Set In XSTO'
 6442 072525     127     122     111    T38WLE: .ASCIZ  'WRITE LOCK ERROR Bit Not Set In XSTO'
 6443 072572     127     122     111    T38NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
 6444 072646     103     157     156    T38SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
 6445 072736     045     116     045    T38INT: .ASCIZ  '%N%A Interrupt Received'
 6446 072766     045     116     045    T38ONL: .ASCIZ  '%N%A Drive Is Now   ON-LINE'
 6447 073022     045     116     045    T38OFL: .ASCIZ  '%N%A Drive Is Now  OFF-LINE'
 6448 073056     103     157     156    T38SST: .ASCIZ  'Contents Of TSSR Incorrect After MESSAGE BUFFER RELEASE'
 6449 073146     045     116     045    T38MS1: .ASCIZ  '%N%AToggle ON-LINE Switch to Generate ATTENTION Interrupts'
 6450 073241     045     116     045    T38MS2: .ASCIZ  '%N%AType RETURN To Return To Menu%N'
 6451 073305     111     163     040    T38MSG: .ASCIZ  'Is Write-Protected Tape Mounted'
 6452 073345     115     141     156    T38ID:  .ASCIZ  'Manual Intervention'
 6453                                    .EVEN
 6454 073372 073416 073470 073516       MIMENU: .WORD   1$,2$,3$,4$,5$,6$
 6455 073406 073665 073730 073773               .WORD   8$,9$,10$,0
 6456
 6457 073416     012     123     105    1$:     .ASCIZ  <12>'SELECT OPERATION FROM FOLLOWING OPTIONS:'
 6458 073470     012     011     060    2$:     .ASCIZ  <12>'    0       Display This Menu'
 6459 073516     011     061     011    3$:     .ASCIZ  '        1       Turn On All M7196 LED''s'
 6460 073550     011     062     011    4$:     .ASCIZ  '        2       Turn Off All M7196 LED''s'
 6461 073603     011     063     011    5$:     .ASCIZ  '        3       Offline/Online Attention'
 6462 073637     011     064     011    6$:     .ASCIZ  '        4       Write Protect Test'
 6463 073665     011     065     011    8$:     .ASCIZ  '        5       Print Extended Transport Status'
 6464 073730     011     066     011    9$:     .ASCIZ  '        6       Return to Diagnostic Supervisor'
 6465 073773     000                    10$:    .ASCIZ  ''
 6466                                    .EVEN
 6467
 6468                                    ;+
```

```
                                        ;LOCAL STORAGE FOR THIS TEST
                                        ;-

073774  000000                          T38DAT: .WORD   0               ;LOGICAL RESPONSE TO QUESTION
073776                                  T38REST:
073776                                          SAVREG                  ;SAVE THE REGISTERS
074002  012701  071450                          MOV     #T38PACKET,R1   ;START OF THE PACKET
074006  012721  140206                          MOV     #140206,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
074012  012721  071460                          MOV     #T38TAD,(R1)+   ;ADDRESS OF DATA BLOCK
074016  005021                                  CLR     (R1)+           ;EXTENDED ADDRESS
074020  012721  000006                          MOV     #6.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
074024  005021                                  CLR     (R1)+           ;CLEAR BSEL0 AND BSEL1
074026  005021                                  CLR     (R1)+           ;CLEAR SEL2
074030  005011                                  CLR     (R1)            ;CLEAR DATA AREA
074032  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
                                        ;       R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
                                        ;       NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
                                        ;
                                        ;
                                        ;-

074034                                  T38MBP:
074034                                          SAVREG                  ;SAVE THE REGISTERS
074040  010005                                  MOV     R0,R5           ;SAVE LOW ORDER ADDRESS
074042  005737  003134                          TST     KTENABLE        ;ADDRESS ABOVE 28K?
074046  001001                                  BNE     910$            ;BR IF YES
074050  005001                                  CLR     R1              ;SET HIGH ORDER ADDRESS TO 0
074052  010103                          910$:   MOV     R1,R3           ;SAVE HIGH ORDER ADDRESS
074054  006100                                  ROL     R0              ;SHIFT BIT15 TO C BIT
074056  006101                                  ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
074060                                          PRINTX  #T38AS0,R1,R5   ;PRINT MESSAGE BUFFER ADDRESS
074060  010546                                                                  MOV     R5,-(SP)
074062  010146                                                                  MOV     R1,-(SP)
074064  012746  074336                                                          MOV     #T38AS0,-(SP)
074070  012746  000003                                                          MOV     #3,-(SP)
074074  010600                                                                  MOV     SP,R0
074076  104415                                                                  TRAP    C$PNTX
074100  062706  000010                                                          ADD     #10,SP
074104                                          PRINTX  #T38AS1         ;PRINT HEADER FOR CONTENTS
074104  012746  074403                                                          MOV     #T38AS1,-(SP)
074110  012746  000001                                                          MOV     #1,-(SP)
074114  010600                                                                  MOV     SP,R0
074116  104415                                                                  TRAP    C$PNTX
074120  062706  000004                                                          ADD     #4,SP
074124  010501                                  MOV     R5,R1           ;COPY LOW ORDER ADDRESS
074126  010300                                  MOV     R3,R0           ;COPY HIGH ORDER ADDRESS
```

```
6514 074130  001403                    BEQ     913$            ;BR IF NOT ABOVE 28K
6515 074132  004737  017316             JSR     PC,SETMAP       ;SETUP PAR ADDRESS IN R0
6516 074136  010005                     MOV     R0,R5           ;GET PAR FORMAT ADDRESS ABOVE 28K
6517 074140  010537  074504   913$:     MOV     R5,T38CNT       ;HOLD ADDRESS
6518 074144  011504             911$:   MOV     (R5),R4         ;GET BUFFER ENTRY
6519 074146  022704  125252             CMP     #125252,R4      ;CHECK FOR NO LOAD CONDITION
6520 074152  001417                     BEQ     912$            ;BR, IF BUFFER WASN'T LOADED
6521 074154  010403                     MOV     R4,R3           ;MAKE COPY
6522 074156  042704  170377             BIC     #170377,R4      ;ONLY BITS 11,10,9 AND 8 ARE SAVED
6523 074162  000241                     CLC                     ;CLEAR CARRY
6524 074164  006004                     ROR     R4              ;11 TO 10 BIT POSITION
6525 074166  006004                     ROR     R4              ;10 TO 9 BIT POSITION
6526 074170  006004                     ROR     R4              ;9 TO 8 BIT POSITON
6527 074172  006004                     ROR     R4              ;8 TO 7 BIT POSITION
6528 074174  042703  177760             BIC     #177760,R3      ;ONLY BITS 3,2,1 AND 0 ARE SAVED
6529 074200  060403                     ADD     R4,R3           ;'OR'EM TOGETHER
6530 074202  010325                     MOV     R3,(R5)+        ;PUT BACK IN BUFFER
6531 074204  020527  072142             CMP     R5,#T38EB       ;END OF BUFFER YET
6532 074210  001355                     BNE     911$            ;BR, IF NOT AT END YET
6533 074212  013705  074504   912$:     MOV     T38CNT,R5       ;PUT ADDRESS BACK
6534 074216  012704  000001             MOV     #1,R4           ;START BYTE NUMBER AT ONE
6535 074222                     915$:   PRINTX  #T38ASN,R4,(R5)+        ;PRT MEM BUFFER W/NEWLINE
     074222  012546                                                        MOV     (R5)+,-(SP)
     074224  010446                                                        MOV     R4,-(SP)
     074226  012746  074460                                                MOV     #T38ASN,-(SP)
     074232  012746  000003                                                MOV     #3,-(SP)
     074236  010600                                                        MOV     SP,R0
     074240  104415                                                        TRAP    C$PNTX
     074242  062706  000010                                                ADD     #10,SP
6536 074246  005037  074504                     CLR     T38CNT          ;CLEAR COUNTER
6537 074252  000412                             BR      921$            ;SKIP OTHER PRINT
6538 074254                     920$:   PRINTX  #T38ASC,R4,(R5)+        ;PRINT THE CONTENTS OF MEMORY BUFFER
     074254  012546                                                        MOV     (R5)+,-(SP)
     074256  010446                                                        MOV     R4,-(SP)
     074260  012746  074441                                                MOV     #T38ASC,-(SP)
     074264  012746  000003                                                MOV     #3,-(SP)
     074270  010600                                                        MOV     SP,R0
     074272  104415                                                        TRAP    C$PNTX
     074274  062706  000010                                                ADD     #10,SP
6539 074300  005237  074504             921$:   INC     T38CNT          ;BUMP COUNTER
6540 074304  005204                             INC     R4              ;NUMBER OF THE NEXT
6541 074306  020427  000200                     CMP     R4,#128.        ;DONE ALL YET ?
6542 074312  003010                             BGT     50$             ;BRANCH IF ALL DONE
6543 074314  023727  074504  000004             CMP     T38CNT,#4       ;DONE FOUR YET
6544 074322  001401                             BEQ     925$            ;BR, IF THREE DONE
6545 074324  000753                             BR      920$            ;KEEP GOING
6546 074326  005037  074504             925$:   CLR     T38CNT          ;CLEAR COUNTER
6547 074332  000733                             BR      915$            ;PRINT WITH NEW LINE
6548 074334  000207                     50$:    RTS     PC              ;RETURN
6549
6550 074336     045     116     045  T38AS0: .ASCIZ  '%N%A Message Buffer Address = %01%05'
6551 074403     045     116     045  T38AS1: .ASCIZ  '%N%A Message Buffer Contents:'
6552 074441     045     101     040  T38ASC: .ASCIZ  '%A  %D4%A: %03'
6553 074460     045     116     045  T38ASN: .ASCIZ  '%N%A Byte%D4%A: %03'
6554                                         .EVEN
6555 074504  000000                 T38CNT: .WORD                   ;COUNTER FOR PRINT
6556 074506                                 ENDTST
```

```
        074506
        074506  104401                          L10075:
                                                        TRAP    C$ETST
```

```
                        .SBTTL  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.
                ;
                ;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.
                ;
074510                  BGNTST
074510                                                          T11::
074510                  RFLAGS  R0                      ;GET OPERATOR FLAGS
074510  104421                                                  TRAP    C$RFLA
074512  001403          BEQ     10$                     ;BR, IF OK TO RUN
074514  012700  076503  MOV     #T39NE,R0               ;"TEST NOT EXECUTED"
074520  000402          BR      11$                     ;JUMP OUT OF TEST IF NOT
074522  012700  077632  10$:    MOV     #TST39ID,R0     ;TEST ID MESSAGE
074526  004737  016510  11$:    JSR     PC,TSTSETUP     ;DO THE COMMON SETUP
074532  004737  020500          JSR     PC,CHKMAN       ;IS MANUAL INTERVENTION ALLOWED?
074536  103402          BCS     20$                     ;BR, IF MANUAL INTERVENTION ALLOWED
074540  000137  075720  JMP     64$                     ;JUMP TO OUT IF NOT
074544                  20$:
074544  004737  015774          JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
074550  103405          BCS     25$                     ;BR IF SOFT INIT = OK
074552  010001          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
074554                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
074554  104455                                                  TRAP    C$ERDF
074556  002115                                                  .WORD   1101
```

```
                074560  003652                                                          .WORD   SFIERR
                074562  012034                                                          .WORD   SFIMSG
        6618    074564                          25$:    CKLOOP                                  ;LOOP IF SELECTED
                074564  104406                                                          TRAP    C$CLP1
        6619    074566  013737  002202  076450          MOV     UNITN,T39DSW            ;SET UNIT NUMBER
        6620    074574  012704  076430                  MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
        6621    074600  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
        6622    074604  103405                          BCS     50$                     ;BR, IF COMMAND ISSUED OK
        6626    074606  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
        6627    074610                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
                074610  104456                                                          TRAP    C$ERHRD
                074612  002116                                                          .WORD   1102
                074614  005056                                                          .WORD   WRTMSG
                074616  012034                                                          .WORD   SFIMSG
        6628    074620                          50$:    CKLOOP                          ;LOOP IF SELECTED
                074620  104406                                                          TRAP    C$CLP1
        6629    074622  013701  075760                  MOV     T39BFR+12,R1            ;GET XST2 STATUS FROM MESSAGE BUFFER
        6630    074626                                  PRINTX  #T39SFS                 ;"STATE OF EXTENDED FEATURES SW ="
                074626  012746  077351                                                  MOV     #T39SFS,-(SP)
                074632  012746  000001                                                  MOV     #1,-(SP)
                074636  010600                                                          MOV     SP,R0
                074640  104415                                                          TRAP    C$PNTX
                074642  062706  000004                                                  ADD     #4,SP
        6631    074646  032701  000200                  BIT     #BIT7,R1                ;CHECK STATE OF E.F.S.
        6632    074652  001011                          BNE     100$                    ;BR, IF EXT. FEA. SW. IS ON
        6633    074654                                  PRINTX  #T39OFF                 ;" OFF"
                074654  012746  077475                                                  MOV     #T39OFF,-(SP)
                074660  012746  000001                                                  MOV     #1,-(SP)
                074664  010600                                                          MOV     SP,R0
                074666  104415                                                          TRAP    C$PNTX
                074670  062706  000004                                                  ADD     #4,SP
        6634    074674  000410                          BR      110$                    ;SKIP OTHER PRINT STATEMENT
        6635    074676                          100$:   PRINTX  #T39ON                  ;" ON "
                074676  012746  077504                                                  MOV     #T39ON,-(SP)
                074702  012746  000001                                                  MOV     #1,-(SP)
                074706  010600                                                          MOV     SP,R0
                074710  104415                                                          TRAP    C$PNTX
                074712  062706  000004                                                  ADD     #4,SP
        6636    074716                          110$:   PRINTX  #T39SBS                 ;"STATE OF BUFFERING SWITCH ="
                074716  012746  077423                                                  MOV     #T39SBS,-(SP)
                074722  012746  000001                                                  MOV     #1,-(SP)
                074726  010600                                                          MOV     SP,R0
                074730  104415                                                          TRAP    C$PNTX
                074732  062706  000004                                                  ADD     #4,SP
        6637    074736  032701  000100                  BIT     #BIT6,R1                ;CHECK STATE OF BUFFERING SW
        6638    074742  001011                          BNE     120$                    ;BR, IF BUFFERING IS ON
        6639    074744                                  PRINTX  #T39OFF                 ;" OFF"
                074744  012746  077475                                                  MOV     #T39OFF,-(SP)
                074750  012746  000001                                                  MOV     #1,-(SP)
                074754  010600                                                          MOV     SP,R0
                074756  104415                                                          TRAP    C$PNTX
                074760  062706  000004                                                  ADD     #4,SP
        6640    074764  000410                          BR      130$                    ;SKIP OTHER PRINT STATEMENT
        6641    074766                          120$:   PRINTX  #T39ON                  ;" ON "
                074766  012746  077504                                                  MOV     #T39ON,-(SP)
                074772  012746  000001                                                  MOV     #1,-(SP)
                074776  010600                                                          MOV     SP,R0
```

```
                075000  104415
                075002  062706  000004
 6642           075006  042701  177700          130$:   BIC     #177700,R1              ;ONLY LEAVE MICROCODE REV LEVEL
 6643           075012  010137  077570                  MOV     R1,T39RL                ;LOAD UP REV LEVEL
 6644           075016                                  PRINTX  #T39MCL,T39RL           ;'MICROCODE REVISION LEVEL =000XXX''
                075016  013746  077570                                                          MOV     T39RL,-(SP)
                075022  012746  077513                                                          MOV     #T39MCL,-(SP)
                075026  012746  000002                                                          MOV     #2,-(SP)
                075032  010600                                                                  MOV     SP,R0
                075034  104415                                                                  TRAP    C$PNTX
                075036  062706  000006                                                          ADD     #6,SP
 6645           075042  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 6646           075046  103405                          BCS     140$                    ;BR IF SOFT INIT = OK
 6650           075050  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6651           075052                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
                075052  104455                                                                  TRAP    C$ERDF
                075054  002117                                                                  .WORD   1103
                075056  003652                                                                  .WORD   SFIERR
                075060  012034                                                                  .WORD   SFIMSG
 6652           075062                          140$:   CKLOOP                                  ;LOOP IF SELECTED
                075062  104406                                                                  TRAP    C$CLP1
 6653           075064  013737  002202  076450          MOV     UNITN,T39DSW            ;SET UNIT NUMBER
 6654           075072  012704  076430                  MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
 6655           075076  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 6656           075102  103405                          BCS     150$                    ;BR, IF COMMAND ISSUED OK
 6660           075104  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6661           075106                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
                075106  104456                                                                  TRAP    C$ERHRD
                075110  002120                                                                  .WORD   1104
                075112  005056                                                                  .WORD   WRTMSG
                075114  012034                                                                  .WORD   SFIMSG
 6662           075116                          150$:   CKLOOP                                  ;LOOP IF SELECTED
                075116  104406                                                                  TRAP    C$CLP1
 6663           075120  005737  002226                  TST     EXTFEA                  ;CHECK FOR EXTENDED FEATURES SW SWITCH
 6664           075124  001036                          BNE     174$                    ;BR IF SWITCH IS ON
 6665           075126  112737  000200  075741          MOVB    #200,T39BS1             ;WRITE MISCELLANEOUS CONT/READ STATUS
 6666           075134  112737  000010  075740          MOVB    #10,T39BS0              ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
 6667           075142  012704  075730                  MOV     #T39PACKET,R4           ;WRITE SUBSYS MEM PACKET
 6668           075146  010465  000000                  MOV     R4,TSDB(R5)             ;ISSUE COMMAND
 6669           075152  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
 6670           075156  103405                          BCS     160$                    ;BR, IF NO ERROR
 6671           075160  010001                          MOV     R0,R1                   ;ERROR, SAVE TSSR
 6675           075162                                  ERRHRD  ERRNO,T39NBA,PKTSSR     ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
                075162  104456                                                                  TRAP    C$ERHRD
                075164  002121                                                                  .WORD   1105
                075166  077205                                                                  .WORD   T39NBA
                075170  012046                                                                  .WORD   PKTSSR
 6676           075172                          160$:   CKLOOP                                  ;LOOP IF SELECTED
                075172  104406                                                                  TRAP    C$CLP1
 6677           075174  012704  076430                  MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
 6678                                           ;*************************************************************************
 6679                                           ;
 6680                                           ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
 6681                                           ;
 6682                                           ;*************************************************************************
 6683
 6684           075200  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
```

```
6685 075204  103405                    BCS     170$                    ;BR, IF COMMAND ISSUED OK
6689 075206  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6690 075210                            ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     075210  104456                                                            TRAP    C$ERHRD
     075212  002122                                                            .WORD   1106
     075214  005056                                                            .WORD   WRTMSG
     075216  012034                                                            .WORD   SFIMSG
6691 075220                    170$:   CKLOOP                          ;SCOPE LOOP
     075220  104406                                                            TRAP    C$CLP1
6692 075222  005037  002202    174$:   CLR     UNITN                   ;SET TO DRIVE 0
6693 075226  013737  002202  076450  175$:   MOV     UNITN,T39DSW            ;SET UNIT NUMBER
6694 075234  012704  076430            MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
6695 075240  004737  010662            JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
6696 075244  103405                    BCS     180$                    ;BR, IF COMMAND ISSUED OK
6700 075246  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6701 075250                            ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     075250  104456                                                            TRAP    C$ERHRD
     075252  002123                                                            .WORD   1107
     075254  005056                                                            .WORD   WRTMSG
     075256  012034                                                            .WORD   SFIMSG
6702 075260                    180$:   CKLOOP                          ;LOOP IF SELECTED
     075260  104406                                                            TRAP    C$CLP1
6703
6704 075262  016501  000002    190$:   MOV     TSSR(R5),R1             ;GET TSSR STATUS
6705 075266  032701  000100            BIT     #OFL,R1                 ;CHECK FOR OFF-LINE
6706 075272  001414                    BEQ     200$                    ;BR, IF DRIVE IS ON-LINE
6707 075274                            PRINTX  #T390F2,UNITN           ;''DRIVE NUMBER XX IS OFF-LINE''
     075274  013746  002202                                                    MOV     UNITN,-(SP)
     075300  012746  076744                                                    MOV     #T390F2,-(SP)
     075304  012746  000002                                                    MOV     #2,-(SP)
     075310  010600                                                            MOV     SP,R0
     075312  104415                                                            TRAP    C$PNTX
     075314  062706  000006                                                    ADD     #6,SP
6708 075320  000137  075654                    JMP     250$                    ;DO NOT TRY TO GET ANYMORE INFO.
6709 075324                    200$:   PRINTX  #T390N2,UNITN           ;''DRIVE NUMBER XX IS ON-LINE''
     075324  013746  002202                                                    MOV     UNITN,-(SP)
     075330  012746  077010                                                    MOV     #T390N2,-(SP)
     075334  012746  000002                                                    MOV     #2,-(SP)
     075340  010600                                                            MOV     SP,R0
     075342  104415                                                            TRAP    C$PNTX
     075344  062706  000006                                                    ADD     #6,SP
6710 075350  013701  075754            MOV     T39BFR+6,R1             ;READ EXTENDED STATUS (XST0)
6711 075354  032701  000004            BIT     #BIT2,R1                ;IS DRIVE WTITE PROTECTED
6712 075360  001013                    BNE     210$                    ;BR, IF WRITE PROTECTED
6713 075362                            PRINTX  #T39WPN,UNITN           ;''DRIVE NUMBER IS NOT WRT PRO''
     075362  013746  002202                                                    MOV     UNITN,-(SP)
     075366  012746  077126                                                    MOV     #T39WPN,-(SP)
     075372  012746  000002                                                    MOV     #2,-(SP)
     075376  010600                                                            MOV     SP,R0
     075400  104415                                                            TRAP    C$PNTX
     075402  062706  000006                                                    ADD     #6,SP
6714 075406  000412                    BR      220$                    ;SKIP OVER
6715 075410                    210$:   PRINTX  #T39WRT,UNITN           ;''DRIVE NUMBER XX IS WRT PRO''
     075410  013746  002202                                                    MOV     UNITN,-(SP)
     075414  012746  077053                                                    MOV     #T39WRT,-(SP)
     075420  012746  000002                                                    MOV     #2,-(SP)
     075424  010600                                                            MOV     SP,R0
```

```
                075426  104415                                                          TRAP    C$PNTX
                075430  062706  000006                                                  ADD     #6,SP
6716            075434  012737  125252  076046  220$:   MOV     #125252,T39BFR+100      ;SET 1 LOC TO KNOWN VALUE
6717            075442  112737  000000  075741          MOVB    #0,T39BS1               ;EXTENDED TAPE STATUS
6718            075450  112737  000024  075740          MOVB    #24,T39BS0              ;EXTENDED TAPE STATUS
6719            075456  012704  075730                  MOV     #T39PACKET,R4           ;WRITE SUBSYS MEM PACKET
6720            075462  010465  000000                  MOV     R4,TSDB(R5)             ;ISSUE COMMAND
6721            075466  012737  000144  075724          MOV     #100.,T39DLY            ;SET UP DELAY ROUTINE
6722            075474  004737  016250          222$:   JSR     PC,WAITF                ;WAIT AWHILE FOR SSR TO SET
6723            075500  016501  000002                  MOV     TSSR(R5),R1             ;SEE IF IT REALLY DID
6724            075504  032701  000200                  BIT     #SSR,R1                 ;JUST CHECK THAT BIT
6725            075510  001017                          BNE     225$                    ;BR, IF SSR IS SET
6726            075512                                  DELAY   250                     ;DELAY ABOUT .25 SEC
                075512  012727  000250                                                  MOV     #250,(PC)+
                075516  000000                                                          .WORD   0
                075520  013727  002116                                                  MOV     L$DLY,(PC)+
                075524  000000                                                          .WORD   0
                075526  005367  177772                                                  DEC     -6(PC)
                075532  001375                                                          BNE     .-4
                075534  005367  177756                                                  DEC     -22(PC)
                075540  001367                                                          BNE     .-20
6727            075542  005337  075724                  DEC     T39DLY                  ;START DELAY COUNT DOWN
6728            075546  001352                          BNE     222$                    ;BR, IF COUNTER IS NOT AT DONE
6729            075550  004737  016336          225$:   JSR     PC,CHKTSSR              ;WAIT FOR SSR
6730            075554  103405                          BCS     230$                    ;BR, IF NO ERROR
6731            075556  010001                          MOV     R0,R1                   ;ERROR, SAVE TSSR
6735            075560                                  ERRHRD  ERRNO,T39NBA,PKTSSR     ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
                075560  104456                                                          TRAP    C$ERHRD
                075562  002124                                                          .WORD   1108
                075564  077205                                                          .WORD   T39NBA
                075566  012046                                                          .WORD   PKTSSR
6736            075570                  230$:   CKLOOP                                  ;LOOP IF SELECTED
                075570  104406                                                          TRAP    C$CLP1
6737            075572  023727  076046  125252          CMP     T39BFR+100,#125252      ;DID LOC GET OVER WRITTEN
6738            075600  001013                          BNE     240$                    ;BR, IF IT DIDN'T GET ETC.
6739            075602                                  PRINTX  #T39ETN,UNITN           ;'DRIVE DOESN'T HAVE EXT TAPE STATUS
                075602  013746  002202                                                  MOV     UNITN,-(SP)
                075606  012746  076651                                                  MOV     #T39ETN,-(SP)
                075612  012746  000002                                                  MOV     #2,-(SP)
                075616  010600                                                          MOV     SP,R0
                075620  104415                                                          TRAP    C$PNTX
                075622  062706  000006                                                  ADD     #6,SP
6740            075626  000412                          BR      250$                    ;SKIP OVER
6741            075630                  240$:   PRINTX  #T39ETS,UNITN           ;'DRIVE HAS EXT TAPE STATUS''
                075630  013746  002202                                                  MOV     UNITN,-(SP)
                075634  012746  076562                                                  MOV     #T39ETS,-(SP)
                075640  012746  000002                                                  MOV     #2,-(SP)
                075644  010600                                                          MOV     SP,R0
                075646  104415                                                          TRAP    C$PNTX
                075650  062706  000006                                                  ADD     #6,SP
6742            075654  005237  002202          250$:   INC     UNITN                   ;BUMP DRIVE NUMBER
6743            075660  023727  002202  000003          CMP     UNITN,#3                ;AT END OF DRIVES YET
6744            075666  001402                          BEQ     63$                     ;BR, IF NO MORE DRIVES
6745            075670  000137  075226                  JMP     175$                    ;DO NEXT DRIVE
6746            075674                  63$:    PRINTX  #T39NFL                 ;NEW LINE
                075674  012746  076500                                                  MOV     #T39NFL,-(SP)
                075700  012746  000001                                                  MOV     #1,-(SP)
```

```
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTX
                                                                                ADD     #4,SP
 075704  010600
 075706  104415
 075710  062706  000004
6747 075714  000137  000200            JMP     200             ;RETURN TO SUPERVISOR
6748 075720                    64$:    EXIT    TST             ;EXIT THIS SECTION
 075720  104432                                                 TRAP    C$EXIT
 075722  001736                                                 .WORD   L10076-.
6749                           ;+
6750                           ;LOCAL TEXT MESSAGES FOR TEST
6751                           ;-
6752
6753                           ;LOCAL STORAGE FOR THIS TEST
6754                           ;-
6755
6756 075724  000000            T39DLY: .WORD   0                       ;DELAY COUNTER FOR TEST
6758         075730                    .=<.+10>&177770
6760 075730                    T39PACKET:                      ;COMMAND PACKET FOR TEST
6761 075730  140006                    .WORD   140006          ;WRITE SUBSYSTEM MEM. CMD,  ACK,CVC=1
6762 075732  075740                    .WORD   T39TAD          ;ADDRESS OF CHARACTERISTICS BLOCK
6763 075734  000000                    .WORD   0
6764 075736  000012                    .WORD   10.             ;STARTING VALUE OF BLOCK SIZE
6765 075740                    T39TAD:                 ;CHARACTERISTICS DATA BLOCK
6766 075740     000            T39BS0: .BYTE   0               ;BSEL0 BYTE
6767 075741     000            T39BS1: .BYTE   0               ;BSEL1 BYTE
6768 075742  000000            T39BS2: .WORD   0               ;BSEL1 WORD
6769 075744  000000                    .WORD   0               ;DATA
6770 075746                    T39BFR: .BLKW   150.            ;MESSAGE BUFFER
6771
6772
6774         076430                    .=<.+10>&177770
6776 076430                    T39PK2:                         ;COMMAND PACKET FOR TEST
6777 076430  140004                    .WORD   140004          ;WRITE CHARA. MEM. CMND.,  ACK,CVC=1
6778 076432  076440                    .WORD   T39DTA          ;ADDRESS OF SELECT DATA BLOCK
6779 076434  000000                    .WORD   0
6780 076436  000012                    .WORD   10.             ;STARTING VALUE OF BLOCK SIZE
6781
6782
6783 076440                    T39DTA:                         ;SELECT DATA BLOCK
6784 076440  075746                    .WORD   T39BFR          ;ADDRESS OF MESSAGE BUFFER
6785 076442  000000                    .WORD   0
6786 076444  000400                    .WORD   256.            ;LENGTH OF MESSAGE BUFFER
6787 076446  000000            T39EAI: .WORD   0               ;EAI BIT WORD
6788 076450  000000            T39DSW: .WORD   0               ;DRIVE SELECT WORD ETC
6790         076460                    .=<.+10>&177770
6792 076460  140012            T39PK3: .WORD   140012          ;MESSAGE BUFFER RELEASE COMMAND
6793 076462  000000                    .WORD   0               ;NOT USED
6794                           ;
6795                           ;WRITE TAPE PACKET
6796                           ;
6798         076470                    .=<.+10>&177770
6800 076470  140005            T39PK4: .WORD   140005          ;WRITE, ACK, CVC=1 COMMAND
6801 076472  000000            T39WR:  .WORD   0               ;ADDRESS OF WRITE BUFFER
6802 076474  000000                    .WORD   0               ;MORE ADDRESS OF WRITE BUFFER
6803 076476  000400            T39SIZ: .WORD   256.            ;SIZE OF RECORD
6804
6805
6806
```

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


076500  045  116  000  T39NFL: .ASCIZ  '%N'
076503  123  164  141  T39NE:  .ASCIZ  'Stand-alone Configuration Typeout Not Executed'
076562  045  116  045  T39ETS: .ASCIZ  '%N%A Extended Tape Status Availiable, Drive Number %D2'
076651  045  116  045  T39ETN: .ASCIZ  '%N%A Extended Tape Status NOT Availiable, Drive Number %D2'
076744  045  116  045  T39OF2: .ASCIZ  '%N%A Drive Number %D2%A Is Off-Line'
077010  045  116  045  T39ON2: .ASCIZ  '%N%A Drive Number %D2%A Is On-Line'
077053  045  116  045  T39WRT: .ASCIZ  '%N%A Drive Number %D2%A Is Write Protected'
077126  045  116  045  T39WPN: .ASCIZ  '%N%A Drive Number %D2%A Is NOT Write Protected'
077205  127  122  111  T39NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
077261  103  157  156  T39SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'

077351  045  116  045  T39SFS: .ASCIZ  '%N%A State Of Extended Features Switch  ='
077423  045  116  045  T39SBS: .ASCIZ  '%N%A State Of Buffering Switch         ='
077475  045  101  040  T39OFF: .ASCIZ  '%A OFF'
077504  045  101  040  T39ON:  .ASCIZ  '%A ON '
077513  045  116  045  T39MCL: .ASCIZ  '%N%A M7196 Microcode Revision Level     =%02'
                             .EVEN
077570  000000           T39RL:  .WORD   0
                             .EVEN
                             .EVEN

                        ;+
                        ;LOCAL STORAGE FOR THIS TEST
                        ;-

077572  000000           T39DAT: .WORD   0              ;LOGICAL RESPONSE TO QUESTION
077574                   T39REST:
077574                           SAVREG                 ;SAVE THE REGISTERS
077600  012701  075730           MOV     #T39PACKET,R1  ;START OF THE PACKET
077604  012721  140006           MOV     #140006,(R1)+  ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
077610  012721  075740           MOV     #T39TAD,(R1)+  ;ADDRESS OF DATA BLOCK
077614  005021                   CLR     (R1)+          ;EXTENDED ADDRESS
077616  012721  000006           MOV     #6.,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
077622  005021                   CLR     (R1)+          ;CLEAR BSEL0 AND BSEL1
077624  005021                   CLR     (R1)+          ;CLEAR SEL2
077626  005011                   CLR     (R1)           ;CLEAR DATA AREA
077630  000207                   RTS     PC             ;RETURN

                        ;+
                        ;LOCAL TEXT MESSAGES FOR TEST
                        ;-

077632  103  157  156  TST39ID:        .ASCIZ  'Configuration Typeout'
                             .EVEN
077660                       ENDTST
077660                                                  L10076:
077660  104401                                                   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.
                        ;-


 077662                         BGNTST
 077662                                                                  T12::
 077662  104421                 RFLAGS  R0              ;GET OPERATOR FLAGS
                                                                         TRAP    C$RFLA
 077664  001403                 BEQ     1$              ;BR, IF OK TO RUN
 077666  012700  101255         MOV     #T40NE,R0       ;"TEST NOT EXECUTED"
 077672  000402                 BR      100$            ;JUST EXIT IF NOT
 077674  012700  101322 1$:     MOV     #TST40ID,R0     ;TEST ID MESSAGE
 077700  004737  016510 100$:   JSR     PC,TSTSETUP     ;DO THE COMMON SETUP
 077704  004737  020500         JSR     PC,CHKMAN       ;SEE IF MANUAL INTERVENTION ALLOWED
 077710  103402                 BCS     2$              ;CARRY SET IF INTERVENTION ALLOWED
 077712  000137  100376         JMP     64$             ;EXIT IF NO MANUAL INTERVENTION
 077716  004737  015774 2$:     JSR     PC,SOFINIT      ;DO A SOFT INIT
 077722  103405                 BCS     5$              ;BRANCH IF OK
 077724  010001                 MOV     R0,R1           ;CONTENTS OF TSSR REGISTER
 077726                         ERRDF   ERRNO,SFIERR,SFIMSG     ;REPORT FATAL ERROR
 077726  104455                                                          TRAP    C$ERDF
 077730  002261                                                          .WORD   1201
```

```
                077732  003652                                                  .WORD   SFIERR
                077734  012034                                                  .WORD   SFIMSG
 6922   077736  012700  100414          5$:     MOV     #SCMENU,R0      ;MENU OF SCOPE LOOP SELECTIONS
 6923   077742  012701  000010                  MOV     #8.,R1          ;MAXIMUM ALLOWED SELECTION
 6924   077746  004737  020256                  JSR     PC,GETSEL       ;GO GET THE OPERATORS SELECTION
 6925   077752  005700                          TST     R0              ;WAS ZERO SPECIFIED ?
 6926   077754  001760                          BEQ     2$              ;REPEAT MENU IF YES.
 6927   077756  020027  000007                  CMP     R0,#7           ;EXTENDED TSSR ?
 6928   077762  001015                          BNE     3$              ;BRANCH IF NOT
 6929   077764  005737  002226                  TST     EXTFEA                  ;CHECK FOR EXTENDED FEATURES SET
 6930   077770  001012                          BNE     3$              ;BR, IF IT IS ON
 6931   077772                                  PRINTF  #EXFMSG         ;WARN OPERATOR EXTENDED FEATURES CLEAR
                077772  012746  101177                                                  MOV     #EXFMSG,-(SP)
                077776  012746  000001                                                  MOV     #1,-(SP)
                100002  010600                                                  MOV     SP,R0
                100004  104417                                                  TRAP    C$PNTF
                100006  062706  000004                                                  ADD     #4,SP
 6932   100012  000137  077716                  JMP     2$              ;GO BACK TO BASIC MENU
 6933   100016  010004          3$:     MOV     R0,R4           ;SAVE THE MENU SELECTION
 6934   100020                          SETPRI  #PRI07          ;RAISE THE PRIORITY
                100020  012700  000340                                                  MOV     #PRI07,R0
                100024  104441                                                  TRAP    C$SPRI
 6935   100026  005037  100406                  CLR     TTION           ;ASSUME INTERRUPTS ARE ENABLED
 6936   100032  032737  000100  177560          BIT     #100,@#TTICSR   ;ARE TTI INTERRUPTS ON ?
 6937   100040  001005                          BNE     4$              ;BRANCH IF YES
 6938   100042  005237  100406                  INC     TTION           ;FLAG SET IF INTERRUPTS OFF
 6939   100046  052737  000100  177560          BIS     #100,@#TTICSR   ;ENABLE INTERRUPTS
 6940   100054  012701  000060          4$:     MOV     #TTIVEC,R1      ;START OF TTI VECTORS
 6941   100060  011137  100410                  MOV     (R1),TVECSAV    ;SAVE THE CURRENT TTI VECTOR
 6942   100064  012721  100310                  MOV     #60$,(R1)+      ;SET NEW INTERRUPT ROUTINE
 6943   100070  011137  100412                  MOV     (R1),TPRISAV    ;SAVE THE VECTOR PRIORITY
 6944   100074  012711  000340                  MOV     #PRI07,(R1)     ;USE PRIORITY SEVEN
 6945   100100                          SETPRI  #PRI00          ;LOWER INTERRUPT BR LEVEL
                100100  012700  000000                                                  MOV     #PRI00,R0
                100104  104441                                                  TRAP    C$SPRI
 6946   100106  006304                          ASL     R4              ;CONVERT TO WORD OFFSET
 6947   100110  000174  100114                  JMP     @6$(R4)         ;JUMP TO PROPER LOOP
 6948   100114  077716          6$:     .WORD   2$              ;RETYPE THE MENU
 6949   100116  100136                  .WORD   10$             ;TSBA READ ACCESS
 6950   100120  100146                  .WORD   15$             ;TSSR READ ACCESS
 6951   100122  100160                  .WORD   20$             ;TSSR WRITE ACCESS
 6952   100124  100200                  .WORD   25$             ;TSDB HIGH BYTE WRITE ACCESS
 6953   100126  100224                  .WORD   30$             ;TSDB LOW BYTE WRITE ACCESS
 6954   100130  100250                  .WORD   35$             ;TSDB MAINTENANCE MODE
 6955   100132  100270                  .WORD   40$             ;TSDBX WRITE ACCESS
 6956   100134  100402                  .WORD   65$             ;LEAVE THE TEST
 6957
 6958
 6959   100136  105065  000000  10$:     CLRB    TSDB(R5)        ;ENTER MAINTENANCE MODE
 6960   100142  011500  12$:     MOV     (R5),R0         ;READ TSBA REGISTER
 6961   100144  000776                  BR      12$             ;LOOP UNTIL HALTED
 6962
 6963
 6964   100146  012703  000002  15$:     MOV     #TSSR,R3        ;ADDRESS OF TSSR REGISTER
 6965   100152  060503                  ADD     R5,R3           ;POINT TO TSV05'S REGISTERS
 6966   100154  011300  18$:     MOV     (R3),R0         ;READ TSSR REGISTER
 6967   100156  000776                  BR      18$             ;LOOP UNTIL STOPPED
```

```
 100160  004737  020174          20$:    JSR     PC,GETPAT               ;READ THE DATA PATTERN
 100164  010001                          MOV     R0,R1                   ;DATA PATTERN FOR LOOP
 100166  012703  000002                  MOV     #TSSR,R3                ;ADDRESS OF TSSR
 100172  060503                          ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
 100174  010113                  22$:    MOV     R1,(R3)                 ;WRITE DATA TO TSSR
 100176  000776                          BR      22$                     ;LOOP

 100200  105065  000000          25$:    CLRB    TSDB(R5)                ;ENTER MAINTENANCE MODE
 100204  004737  020174                  JSR     PC,GETPAT               ;READ THE DATA PATTERN
 100210  010001                          MOV     R0,R1                   ;DATA PATTERN FOR LOOP
 100212  012703  000001                  MOV     #TSDBH,R3               ;ADDRESS OF HIGH BYTE OF TSDB
 100216  060503                          ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
 100220  110113                  27$:    MOVB    R1,(R3)                 ;WRITE THE DATA TO TSDB, HIGH BYTE
 100222  000776                          BR      27$                     ;LOOP UNTIL STOPPED

 100224  105065  000000          30$:    CLRB    TSDB(R5)                ;ENTER MAINTENANCE MODE
 100230  004737  020174                  JSR     PC,GETPAT               ;READ THE DATA PATTERN
 100234  010001                          MOV     R0,R1                   ;DATA PATTERN FOR LOOP
 100236  012703  000000                  MOV     #TSDB,R3                ;ADDRESS OF TSSR
 100242  060503                          ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
 100244  110113                  32$:    MOVB    R1,(R3)                 ;WRITE DATA TO TSSR, LOW BYTE
 100246  000776                          BR      32$                     ;LOOP UNTIL HALTED BY OPERATOR

 100250  004737  020174          35$:    JSR     PC,GETPAT               ;READ THE DATA PATTERN
 100254  010001                          MOV     R0,R1                   ;DATA PATTERN FOR LOOP
 100256  012703  000000                  MOV     #TSDB,R3                ;SELECT TSDB
 100262  060503                          ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
 100264  010113                  37$:    MOV     R1,(R3)                 ;WRITE THE DATA PATTERN

 100266  000776                          BR      37$                     ;LOOP UNTIL HALTED

 100270  004737  020174          40$:    JSR     PC,GETPAT               ;READ THE DATA PATTERN
 100274  010001                          MOV     R0,R1                   ;SAVE THE DATA PATTERN
 100276  012703  000003                  MOV     #TSSRH,R3               ;BYTE ADDRESS OF TSSR, HIGH BYTE
 100302  060503                          ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
 100304  110113                  42$:    MOVB    R1,(R3)                 ;WRITE THE DATA TO REGISTER
 100306  000776                          BR      42$                     ;LOOP UNTIL HALTED


                                 ;+
                                 ;PROCESS CONSOLE INTERRUPTS
                                 ;-

 100310  010046                  60$:    MOV     R0,-(SP)                ;SAVE WORK REGISTER
 100312  113700  177562                  MOVB    @#TTIBFR,R0             ;GET THE OPERATOR INPUT
 100316  042700  000200                  BIC     #200,R0                 ;STRIP OFF PARITY BIT
 100322  122700  000015                  CMPB    #15,R0                  ;IS IT A CARRIAGE RETURN ?
 100326  001021                          BNE     61$                     ;JUST EXIT IF NOT
 100330  012766  077716  000002          MOV     #2$,2(SP)               ;RETURN TO MASTER MENU
 100336  005066  000004                  CLR     4(SP)                   ;FORCE PRIORITY ZERO
 100342  013737  100410  000060          MOV     TVECSAV,@#TTIVEC        ;RESTORE SUPERVISOR VECTOR
 100350  013737  100412  000062          MOV     TPRISAV,@#TTIVEC+2      ;RESTORE SUPERVISOR PRIORITY
 100356  005737  100406                  TST     TTION                   ;ARE SUPERVISOR INTERRUPTS ENABLED ?
 100362  001403                          BEQ     61$                     ;BRANCH IF YES
```

```
 7025 100364  042737  000100  177560          BIC     #100,@#TTICSR           ;TURN OFF TTI INTERRUPTS
 7026 100372  012600                  61$:     MOV     (SP)+,R0                ;RESTORE REGISTER
 7027 100374  000002                          RTI                             ;RETURN FROM INTERUPT
 7028
 7029 100376                          64$:
 7030 100376                          63$:     EXIT    TST                     ;EXIT THE TEST
      100376  104432                                                                  TRAP    C$EXIT
      100400  000736                                                                 .WORD   L10077-.
 7031 100402  000137  000200          65$:     JMP     200                     ;RETURN TO SUPERVISOR
 7032
 7033                                 ;+
 7034                                 ;LOCAL STORAGE FOR THIS TEST
 7035                                 ;-
 7036
 7037 100406  000000                  TTION:          .WORD   0               ;WORD SET IF SUPERVISOR TTI INTER OFF
 7038 100410  000000                  TVECSAV:        .WORD   0               ;SAVE TTI VECTOR
 7039 100412  000000                  TPRISAV:        .WORD   0               ;SAVE TTI PRIORITY
 7040
 7041
 7042                                 ;+
 7043                                 ;MENU FOR OPERATOR INPUT FOR SCOPE LOOPS
 7044                                 ;-
 7045
 7046                                         .EVEN
 7047 100414  100446  100521  100547  SCMENU: .WORD   1$,2$,3$,4$,5$,6$
 7048 100430  100720  100756  101024          .WORD   7$,8$,9$,10$,11$,12$,0
 7049
 7050
 7051 100446    012     123     105   1$:     .ASCIZ  <12>'SELECT SCOPE LOOP FROM FOLLOWING OPTIONS:'
 7052 100521    012     011     060   2$:     .ASCIZ  <12>'    0       Display This Menu'
 7053 100547    011     061     011   3$:     .ASCIZ  '        1       TSBA Read Access'
 7054 100573    011     062     011   4$:     .ASCIZ  '        2       TSSR Read Access'
 7055 100617    011     063     011   5$:     .ASCIZ  '        3       Initialize (TSSR Write Access,'
 7056 100661    011     064     011   6$:     .ASCIZ  '        4       TSDB High Byte Write Access'
 7057 100720    011     065     011   7$:     .ASCIZ  '        5       TSDB Low Byte Write Access'
 7058 100756    011     066     011   8$:     .ASCIZ  '        6       TSDB Maintenance Mode Write Access'
 7059 101024    011     067     011   9$:     .ASCIZ  '        7       TSDBX (TSSR High Byte) Write Access'
 7060 101073    011     070     011   10$:    .ASCIZ  '        8       Return to Diagnostic Supervisor'
 7061 101136    000                   11$:    .ASCIZ  ''
 7062 101137    124     171     160   12$:    .ASCIZ  'Type RETURN To Stop Scope Loops'
 7063 101177    045     116     045   EXFMSG: .ASCIZ  '%N%A *** Extended Features Switch Not On *** '
 7064 101255    123     164     141   T4ONE:  .ASCIZ  'Stand-alone Scope Loops Not Executed'
 7065 101322    123     143     157   TST4OID:        .ASCIZ  'Scope Loops'
 7066
 7067 101336                                  .EVEN
      101336                                  ENDTST
                                                                          L10077:
      101336  104401                                                              TRAP    C$ETST
 7068 101340                                  ENDMOD
```

```
                                        .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
101340  000010                        .WORD L10100-L$HARD/2
                                L$HARD::
                                        GPRMA   HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
101342  000031                        .WORD   T$CODE
101344  101362                        .WORD   HPM1
101346  160010                        .WORD   T$LOLIM
101350  177776                        .WORD   T$HILIM
                                        GPRMA   HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
101352  001031                        .WORD   T$CODE
101354  101416                        .WORD   HPM2
101356  000000                        .WORD   T$LOLIM
101360  000776                        .WORD   T$HILIM
                                        ;GPRMD  HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
                                        ENDHRD
                                        .EVEN
                                L10100:
101362  104     105     126     HPM1:   .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
101416  111     116     124     HPM2:   .ASCIZ  'INTERRUPT VECTOR    '
101442  111     116     124     HPM3:   .ASCIZ  'INTERRUPT PRIORITY     '
                                        .EVEN
```

```
                                        .SBTTL  SOFTWARE PARAMETER CODING SECTION

                                ;++
                                ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--
101472                                  BGNSFT
101472  000003                          .WORD L10101-L$SOFT/2
101474                          L$SOFT::
                                ;       GPRML   SPM1,0,-1,YES                   ; GET TRANSPORT TEST FLAG.
101474                                  GPRML   SPM4,2,-1,YES                   ; GET ITERATION CONTROL.
101474  001130                          .WORD   T$CODE
101476  101532                          .WORD   SPM4
101500  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
101502                                  ENDSFT
                                        .EVEN
101502                          L10101:

101502  105  116  101  SPM1:   .ASCIZ  'ENABLE TRANSPORT TESTS '
101532  111  116  110  SPM4:   .ASCIZ  'INHIBIT ITERATIONS     '
101562  120  105  122  SPM6:   .ASCIZ  'PER TEST ERROR LIMIT   '
101612  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).
                                ;

101642                          PATCH::

101642                                  .BLKW   32.

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

```
ADDSSR  012126 G        C$AU  = 000052        DEVDRO  023376          FREEHI  003130          INTCPC  016150
ADR   = 000020 G        C$AUTO= 000061        DEVNRD  023315          FRESIZ  003126 G        INTFLA  016145
AMBTSS  006635          C$BRK = 000022        DEVNXR  023233          FUSI    004117          INTMAS  016144
ASSEMB= 000010          C$BSEG= 000004        DEVONL  023163          F$AU  = 000015          INTR    016216 G
A1716 = 000003          C$BSUB= 000002        DEVSUM  023126          F$AUTO= 000020          INTREC  002224 G
BADDAT  003156 G        C$CEFG= 000045        DFPTBL  002156 G        F$BGN = 000040          INTVEC  016146
BADSSR  015700 G        C$CLCK= 000062        DIAGMC= 000000          F$CLEA= 000007          INTX    004300
BDVPCR= 177520 G        C$CLEA= 000012        DICEB = 000001          F$DU  = 000016          INVERT  021206 G
BENBSW  002230 G        C$CLOS= 000035        DSBINT  016204          F$END = 000041          IOKCKI= 000200
BIE   = 040000          C$CLP1= 000006        DUAD12  004643          F$HARD= 000004          IOKSTP= 000001
BIT0  = 000001 G        C$CVEC= 000036        DUFLG   003112 G        F$HW  = 000013          IPRI    002212 G
BIT00 = 000001 G        C$DCLN= 000044        DUMMY   003062          F$INIT= 000006          ISR   = 000100 G
BIT01 = 000002 G        C$DODU= 000051        EF.CON= 000036 G        F$JMP = 000050          IVEC    002210 G
BIT02 = 000004 G        C$DRPT= 000024        EF.NEW= 000035 G        F$MOD = 000000          IXE   = 004000 G
BIT03 = 000010 G        C$DU  = 000053        EF.PWR= 000034 G        F$MSG = 000011          I$AU  = 000041
BIT04 = 000020 G        C$EDIT= 000003        EF.RES= 000037 G        F$PROT= 000021          I$AUTO= 000041
BIT05 = 000040 G        C$ERDF= 000055        EF.STA= 000040 G        F$PWR = 000017          I$CLN = 000041
BIT06 = 000100 G        C$ERHR= 000056        EMAXDU  016777          F$RPT = 000012          I$DU  = 000041
BIT07 = 000200 G        C$ERRO= 000060        EN    = 000000          F$SEG = 000003          I$HRD = 000041
BIT08 = 000400 G        C$ERSF= 000054        ENAINT  016152          F$SOFT= 000005          I$INIT= 000041
BIT09 = 001000 G        C$ERSO= 000057        ENVIRN  020630          F$SRV = 000010          I$MOD = 000041
BIT1  = 000002 G        C$ESCA= 000010        EPRTSW  002200 G        F$SUB = 000002          I$MSG = 000041
BIT10 = 002000 G        C$ESEG= 000005        EPRT1   006360          F$SW  = 000014          I$PROT= 000040
BIT11 = 004000 G        C$ESUB= 000003        EPRT2   006360          F$TEST= 000001          I$PTAB= 000041
BIT12 = 010000 G        C$ETST= 000001        ERCM    011733          GDDAT   003160 G        I$PWR = 000041
BIT13 = 020000 G        C$EXIT= 000032        ERRHI   002236 G        GERRMA  002174 G        I$RPT = 000041
BIT14 = 040000 G        C$GETB= 000026        ERRK    016756          GETPAT  020174 G        I$SEG = 000041
BIT15 = 100000 G        C$GETW= 000027        ERRLO   002240 G        GETSEL  020256 G        I$SETU= 000041
BIT2  = 000004 G        C$GMAN= 000043        ERRNO = 002261          G$CNTO= 000200          I$SFT = 000041
BIT3  = 000010 G        C$GPHR= 000042        ERRVEC= 000004 G        G$DELM= 000372          I$SRV = 000041
BIT4  = 000020 G        C$GPLO= 000030        ERTABE  003376          G$DISP= 000003          I$SUB = 000041
BIT5  = 000040 G        C$GPRI= 000040        ERTABL  003176          G$EXCP= 000400          I$TST = 000041
BIT6  = 000100 G        C$INIT= 000011        ESUM    016760          G$HILI= 000002          J$JMP = 000167
BIT7  = 000200 G        C$INLP= 000020        EVL   = 000004 G        G$LOLI= 000001          KIPAR0= 172340
BIT8  = 000400 G        C$MANI= 000050        EXBCNT= 000010          G$NO  = 000000          KIPAR1= 172342
BIT9  = 001000 G        C$MEM = 000031        EXFMSG  101177          G$OFFS= 000400          KIPAR2= 172344
BOE   = 000400 G        C$MSG = 000023        EXPBRE  015502 G        G$OFSI= 000376          KIPAR3= 172346
BRINIT  004457          C$OPEN= 000034        EXPD    002232 G        G$PRMA= 000001          KIPAR4= 172350
BSEL0 = 000000          C$PNTB= 000014        EXPGOT  004533          G$PRMD= 000002          KIPAR5= 172352
BSEL1 = 000001          C$PNTF= 000017        EXPGT2  004567          G$PRML= 000000          KIPAR6= 172354
CHKAMB  016044          C$PNTS= 000016        EXPMSG  002322 G        G$RADA= 000140          KIPAR7= 172356
CHKMAN  020500 G        C$PNTX= 000015        EXPREC  015474 G        G$RADB= 000000          KIPDR0= 172300
CHKTSS  016336          C$QIO = 000377        EXTA    005772          G$RADD= 000040          KIPDR1= 172302
CKDROP  017202          C$RDBU= 000007        EXTEND  005770          G$RADL= 000120          KIPDR2= 172304
CKEMAX  017102          C$REFG= 000047        EXTFEA  002226 G        G$RADO= 000020          KIPDR3= 172306
CKMSG   011360 G        C$RESE= 000033        E$END = 002100          G$XFER= 000004          KIPDR4= 172310
CKMSG2  011500 G        C$REVI= 000003        E$LOAD= 000035          G$YES = 000010          KIPDR5= 172312
CKRAM   011114 G        C$RFLA= 000021        FATERR= 000060          HIADDR= 001400          KIPDR6= 172314
CKRAM2  011224 G        C$RPT = 000025        FATFLG  002222 G        HOE   = 100000 G        KIPDR7= 172316
CMDPKT  021260 G        C$SEFG= 000046        FERCM   011722          HPM1    101362          KTENAB  003134 G
CMPMEM  017660          C$SPRI= 000041        FIFEXP  012170 G        HPM2    101416          KTFLG   003132 G
CONFIG  017250          C$SVEC= 000037        FIF1MS  012242          HPM3    101442          KTINIT  021054
COUNT   002310 G        C$TPRI= 000013        FIF2MS  012311          IBE   = 010000 G        KTOFF   017274
CSRADD  002206 G        DATA    002312 G      FILLME  017422          IDU   = 000040 G        KTON    017256
CTAB    003164 G        DATASC  020232        FNOINT  004215          IER   = 020000 G        LERRMA  002172 G
CTABE   003176 G        DEBUGM  011632        FORCER  002176 G        IFAULT  004256          LERRNO= 000000
CTABM   003164 G        DEVCNT  002220 G      FREE    003124 G        INCERK  017044          LISTAL= 000001
```

```
LOE   = 040000 G      L$UNIT  002012 G      L10071  056510        OFL   = 000100        PRMNO   002320 G
LOOPCN  002216 G      L10000  002164        L10072  057754        ONEFIL= 000000        PRMSGE  014552 G
LOOPCO  013126        L10001  002176        L10073  066644        O$APTS= 000000        PRMSG0  014732
LOOPFL  003162 G      L10002  005766        L10074  065004        O$AU  = 000001        PRMSG1  014777
LOT   = 000010 G      L10003  012044        L10075  074506        O$BGNR= 000001        PRMSG2  015035
L$ACP   002110 G      L10004  012062        L10076  077660        O$BGNS= 000001        PROASC  014420
L$APT   002036 G      L10005  012100        L10077  101336        O$DU  = 000001        PR1ASC  014465
L$AU    022352 G      L10006  012106        L10100  101362        O$ERRT= 000000        PST32W  003152 G
L$AUT   002070 G      L10007  012124        L10101  101502        O$GNSW= 000001        PUNIT   022304
L$AUTO  022556 G      L10010  012142        MEMADD  013754 G      O$POIN= 000001        PW.D11= 000021
L$CCP   002106 G      L10011  012166        MEMCK   021276 G      O$SETU= 000000        PW.D13= 000022
L$CLEA  022636 G      L10012  012240        MENASC  020447        PASRPT  022054        PW.D22= 000020
L$CO    002032 G      L10013  012410        MENERR  020374        PATCH   101642 G      PW.NOP= 000000
L$DEPO  002011 G      L10014  013124        MENRES  020476        PATDAT  020230        PW.NO1= 000023
L$DESC  003410 G      L10015  013752        MIMENU  073372        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$DU    022450 G      L10024  015576        MSGLOO  013064 G      PKLOW = 000002        PW.WMI= 000010
L$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  022302        MS.ATT= 000006        PKTGET  012064 G      P.ACK = 100000
L$ENVI  002044 G      L10031  022446        MS.EXT= 000200        PKTMES  012110 G      P.CMD = 000037
L$ETP   002102 G      L10032  022554        MS.RSD= 000001        PKTRAM  004745 G      P.CONT= 000012
L$EXP1  002046 G      L10033  022634        MS.RSF= 000020        PKTSSR  012046 G      P.CVC = 040000
L$EXP4  002064 G      L10034  022662        MS.RST= 000010        PNT   = 001000 G      P.FMT = 000140
L$EXP5  002066 G      L10035  023124        M8186   005554        PRAMPK  013776        P.FORM= 000011
L$HARD  101342 G      L10036  024424        M8189   005645        PRASC   014523        P.GETS= 000017
L$HIME  002120 G      L10037  026416        NBA   = 002000        PRBEXP  015470        P.IE  = 000200
L$HPCP  002016 G      L10040  024700        NEWPAS  022010        PRBMSG  015336        P.INIi= 000013
L$HPTP  002022 G      L10041  025144        NODEV   003114 G      PRBREC  015472        P.MODE= 007400
L$HW    002156 G      L10042  031772        NOEXTF  030164        PRBTOT  015423        P.OPP = 020000
L$ICP   002104 G      L10043  026772        NOINIT  004335        PRBYTE  015122 G      P.POSI= 000010
L$INIT  021556 G      L10044  027262        NOINTR  004221        PRI   = 002000 G      P.READ= 000001
L$LADP  002026 G      L10045  027560        NOITS   002170 G      PRIADD  010160        P.SWB = 010000
L$LAST  102004 G      L10046  030170        NOMAN   020534        PRIA0   010230        P.WRIT= 000005
L$LOAD  002100 G      L10047  034562        NOMEM   005460        PRIBX0  007612 G      P.WRTC= 000004
L$LUN   002074 G      L10050  032326        NP.IR = 000200        PRIEQU  010060        P.WRTS= 000006
L$MREV  002050 G      L10051  032720        NP.LOO= 000040        PRIPKT  007370 G      QVP     002204 G
L$NAME  002000 G      L10052  033326        NP.OUT= 000100        PRIRAM  010066        RAMASC  014156
L$PRIO  002042 G      L10053  040354        NP.WRP= 000020        PRITAD  010274        RAMDAT  002242 G
L$PROT  021546 G      L10054  035646        NSI     004152        PRITSS  006024        RAMERR  015510 G
L$PRT   002112 G      L10055  036600        NSINIT  004407        PRIT0   010356        RAMEXP  015530 G
L$REPP  002062 G      L10056  050466        NUL     004527        PRIT1   010421        RAMFOR  010116
L$REV   002010 G      L10057  040660        NULCR   004530        PRIXOR  007742 G      RAMSIZ  002302 G
L$RPT   022664 G      L10060  042070        NXM   = 004000        PRI00 = 000000 G      RAMTAD  015516 G
L$SOFT  101474 G      L10061  043430        NXMFLG  003136 G      PRI01 = 000040 G      RCVHIA  002304 G
L$SPC   002056 G      L10062  044006        NXMHI   003142 G      PRI02 = 000100 G      RCVLOA  002306 G
L$SPCP  002020 G      L10063  045260        NXMLO   003140 G      PRI03 = 000140 G      RDERR   005206
L$SPTP  002024 G      L10064  046324        NXMTST  021452        PRI04 = 000200 G      RECMSG  002466 G
L$STA   002030 G      L10065  051746        NXR     003740        PRI05 = 000240 G      RECV    002234 G
L$SW    002166 G      L10066  062574        NXRERR  005736 G      PRI06 = 000300 G      REGSAV  020140
L$TEST  002114 G      L10067  053752        NXRX    003777        PRI07 = 000340 G      RETERR  005372
L$TIML  002014 G      L10070  055240        NXTU    022022        PRMESS  014242        REWIND  011014 G
```

```
RMCHBE= 000167          S1.IID= 004000          TST40I  101322          T10     066646 G        T15BFR  033372
RMCHEN= 000200          S1.I1R= 020000          TSV2    002000 G        T11     074510 G        T15BF2  034060
RMMSGB= 000215          S1.I2R= 040000          TSV3    002176 G        T12     077662 G        T15BS0  034060
RMMSGE= 000234          S1.PAR= 100000          TSV4    021546 G        T12BFR  030262          T15BS1  034061
RMPKTB= 000201          S2.ATI= 000010          TSV5    023446 G        T12BKG  030737          T15DAT  033360
RMPKTE= 000210          S2.BTI= 000004          TSV6    101340 G        T12BLK  030314          T15LOO  032024
RMR   = 010000          S2.DIM= 000200          TTIBFR= 177562 G        T12CHA  031730          T15PAC  033350
RWPACK  011110          S2.ILW= 000100          TTICSR= 177560 G        T12CKR  031510 G        T15PK2  034050
SC    = 100000          S2.INR= 000020          TTION   100406          T12CON  031316          T15RES  034452
SCE   = 020000          S2.OUT= 000040          TTION2  071434          T12DAT  030250          T15RT2  034524
SCHERR  005300          S2.UND= 000003          TTIVEC= 000060 G        T12DPR  031116          T15SSR  034066
SCME    005013          TBLEND= 003062 G        TVECSA  100410          T12GET  030476          T15S2   034062
SCMENU  100414          TCOASC  006476          TVSAV2  071436          T12HIA  030302          T15S3   034064
SDELAY  010660          TCOCOD  006676          T$ARGC= 000001          T12KT   030310          T16BEN  040240
SELASC  020442          TEMP1   003116 G        T$CODE= 001130          T12LOA  030304          T16BFR  040212
SELDAT= 000004          TEMP2   003120 G        T$ERRN= 002261          T12LOO  026446          T16BFS  040232
SEL2  = 000002          TERCLS= 000016          T$EXCP= 000000          T12MSG  030641          T16CLR  040056
SETMAP  017316          TESTNO= 000014          T$FLAG= 000040          T12NIN  031025          T16DAT  040200
SETU    022106          TEXASC  006435          T$GMAN= 000000          T12NXM  031207          T16DT2  040250
SFFMSG  012102 G        TFCASC  006537          T$HILI= 000776          T12PAC  030240          T16D01  036632
SFHERR  003705          TIMEXP  015552 G        T$LAST= 000001          T12PAR  030306          T16D28  036634
SFIERR  003652          TIMSGO  015600          T$LOLI= 000000          T12SET  031666          T16D53  036636
SFIMSG  012034 G        TINERR  012021          T$LSYM= 010000          T12SWR  031620          T16D78  036640
SFPTBL  002166 G        TMPBFR  002632 G        T$LTNO= 000014          T12TBE  030446          T16LEN  037162
SIFLAG  003154 G        TNAM    016704          T$NEST= 177777          T12WRT  030552          T16LOO  034602
SIMSG   011766          TPRISA  100412          T$NS0 = 000000          T121LO  026560          T16PAC  040170
SKIPT   003400          TPSAV2  071440          T$NS1 = 000005          T122LO  027114          T16PK2  040240
SOFINI  015774 G        TRANST  002166 G        T$NS2 = 000002          T123LO  027334          T16REJ  037274
SPACE   010466 G        TSBA  = 000000 G        T$NS3 = 000003          T124LO  027734          T16RES  040010
SPM1    101502          TSBAH = 000001 G        T$PTNU= 000000          T124TS  030312          T16SEX  040150
SPM4    101532          TSDB  = 000000 G        T$SAVL= 177777          T13BFR  024062          T16SRD  040102
SPM6    101562          TSDBH = 000001 G        T$SEGL= 177777          T13DAT  024050          T16SSR  036712
SPM7    101612          TSFCOD  007236          T$SEK0= 010000          T13LCO  023464          T16TSB  037060
SR0   = 177572          TSREJ = 000006          T$SUBN= 000000          T13MEM  024155          T16T01  037411
SR1   = 177574          TSSDEF  006606          T$TAGL= 177777          T13NBA  024214          T16T28  037510
SR2   = 177576          TSSR  = 000002 G        T$TAGN= 010102          T13PAC  024040          T16T53  037610
SR3   = 172516          TSSRBI  003502 G        T$TEMP= 000000          T13RES  024356          T16T78  037710
SSR   = 000200          TSSRFO  006415          T$TEST= 000014          T13SSR  024267          T16WMI  040122
STATCO  012412          TSSRH = 000003 G        T$TSTM= 177777          T14BFR  025176          T162SS  036747
SVCGBL= 000000          TSSX    004020          T$TSTS= 000001          T14BS0  025170          T163SS  037013
SVCINS= 000000          TSTBLK  002752 G        T$$AU = 010031          T14BS1  025171          T17BEN  050342
SVCSUB= 000001          TSTCNT  002214 G        T$$AUT= 010033          T14BS2  025172          T17BFR  050222
SVCTAG= 000000          TSTEND  016720          T$$CLE= 010034          T14DAT  025170          T17BFS  050242
SVCTST= 000001          TSTFLA  002314 G        T$$DU = 010032          T14DTA  025610          T17CLE  050126
S$LSYM= 010000          TSTLOO  016456 G        T$$HAR= 010100          T14LOO  024444          T17CLR  047740
S0.IDB= 000010          TSTPTR  002316 G        T$$HW = 010000          T14NBA  025622          T17DAT  050210
S0.IFB= 000002          TSTSET  016510 G        T$$INI= 010030          T14NIN  026063          T17DT2  050360
S0.IFP= 000001          TST12I  030450          T$$MSG= 010025          T14PAC  025160          T17EXE  046504
S0.ILD= 000020          TST13I  024102          T$$PRO= 010027          T14PK2  025600          T17EXP  046362
S0.ION= 000040          TST14I  026241          T$$RPT= 010035          T14RES  026306          T17EXS  046402
S0.IRD= 000100          TST15I  034433          T$$SEG= 010000          T14RST  026344          T17LOO  040404
S0.IRW= 000004          TST16I  036642          T$$SOF= 010101          T14SSR  025773          T17MSK  046356
S0.ISP= 000200          TST17I  046606          T$$SRV= 010026          T14TSB  026155          T17PAC  050200
S1.ICE= 002000          TST18I  051176          T$$SUB= 010074          T142RE  025676          T17PK2  050350
S1.IEO= 010000          TST19I  060162          T$$SW = 010001          T15AM2  034142          T17RFI  050106
S1.IFM= 001000          TST20I  065162          T$$TES= 010077          T15AM3  034243          T17RSF  050004
S1.IHE= 000400          TST39I  077632          T1      023446 G        T15AM4  034345          T17SET  050146
```

```
T17SNP  050026          T19SSR  060223          T3      026420 G        T39NBA  077205          WF.IED= 000010
T17SRD  047764          T19WCT  062122          T3BFLG  003150 G        T39NE   076503          WF.IER= 000004
T17SSR  046625          T19WFI  062066          T3.1    026446          T39NFL  076500          WF.IHI= 000200
T17WFD  046504          T19WFM  062142          T3.2    027006          T39OFF  077475          WF.IRE= 000040
T17WFI  050052          T19WST  061523          T3.3    027276          T39OF2  076744          WF.IWF= 000020
T171CM  047145          T191CM  060660          T3.4    027574          T39ON   077504          WF.IWR= 000100
T172CM  047227          T192CM  060742          T38ASC  074441          T39ON2  077010          WF.I3R= 000002
T172SS  046662          T192SS  060260          T38ASN  074460          T39PAC  075730          WF.I4R= 000001
T173CM  047323          T193CM  061036          T38ASO  074336          T39PK2  076430          WRTCHR  010662 G
T173SS  046726          T193SS  060324          T38AS1  074403          T39PK3  076460          WRTERR  005113
T174CM  047407          T194SS  060371          T38BFR  071466          T39PK4  076470          WRTMSG  005056
T174SS  046773          T195CM  061124          T38BSO  071460          T39RES  077574          WSMBK   021270 G
T175CM  047472          T195SS  060434          T38BS1  071461          T39RL   077570          XFERAS  015740
T175SS  047036          T196CM  061200          T38BS2  071462          T39SBS  077423          XNXM    016376
T176CM  047546          T196SS  060500          T38CNT  074504          T39SFS  077351          XORBFO  007674
T176SS  047102          T197CM  061263          T38DAT  073774          T39SIZ  076476          XORFOR  010012
T177CM  047654          T197SS  060543          T38DLY  071442          T39SSR  077261          XST0    = 000006 G
T18BFR  051702          T198CM  061351          T38DSW  072170          T39TAD  075740          XST1    = 000010 G
T18CMP  051403          T198SS  060612          T38DTA  072160          T39WPN  077126          XST2    = 000012 G
T18DAT  051670          T199CM  061440          T38EAI  072166          T39WR   076472          XST3    = 000014 G
T18DT2  051740          T2      024426 G        T38EB   072142          T39WRT  077053          XST4    = 000016 G
T18EXP  051146          T2.1    024444          T38ID   073345          T4      031774 G        XSOBOT= 000002
T18LOO  050506          T2.2    024714          T38INT  072736          T4.1    032024          XSOEOT= 000001
T18MSK  051142          T20BEN  066522          T38MBP  074034          T4.2    032330          XSOIE = 000040
T18PAC  051660          T20BFR  066402          T38MSG  073305          T4.3    032722          XSOILA= 000400
T18PK2  051730          T20BFS  066422          T38MS1  073146          T40NE   101255          XSOILC= 001000
T18SET  051550          T20CLE  066314          T38MS2  073241          T5      034564 G        XSOLET= 020000
T18SMI  051526          T20CLR  066102          T38MS3  072305          T5.1    034602          XSOMOT= 000200
T18SRD  051506          T20CWP  066007          T38NBA  072572          T5.2    035662          XSONEF= 002000
T18SSR  051235          T20DAT  066370          T38NE   072230          T6      040356 G        XSOONL= 000100
T18XS   051172          T20DT2  066540          T38OFL  073022          T6.1    040404          XSOPED= 000010
T182SS  051272          T20EXE  065162          T38ONL  072766          T6.2    040674          XSORLL= 010000
T183SS  051336          T20EXP  065042          T38PAC  071450          T6.3    042104          XSORLS= 040000
T19BEN  062452          T20EXS  065062          T38PK2  072150          T6.4    043444          XSOTMK= 100000
T19BFC  057776          T20LOO  062614          T38PK3  072200          T6.5    044022          XSOVCK= 000020
T19BFR  062332          T20MSK  065036          T38PK4  072220          T6.6    045274          XSOWLE= 004000
T19BFS  062352          T20PAC  066360          T38RES  073776          T7      050470 G        XSOWLK= 000004
T19CLE  062200          T20PK2  066530          T38SIZ  072226          T8      051750 G        XXCOMM  003122 G
T19CLR  061734          T20RFI  066166          T38SSR  072646          T8.1    051766          X$ALWA= 000000
T19CNV  062244          T20RSF  065706          T38SST  073056          T8.2    053766          X$FALS= 000040
T19DAT  062320          T20SET  066334          T38TAD  071460          T8.3    055254          X$OFFS= 000400
T19DT2  062470          T20SEX  066276          T38WLE  072525          T8.4    056524          X$TRUE= 000020
T19EXE  060162          T20SRD  066126          T38WR   072222          T9      062576 G        X1.COR= 020000
T19EXP  060042          T20SSR  065211          T38WRL  072464          T9.1    062614          X1.DLT= 100000
T19EXS  060062          T20SWP  065577          T38WRT  072400          UAM     = 000200 G      X1.MBZ= 017375
T19LOO  051766          T20WFI  066206          T39BFR  075746          UNITN   002202 G        X1.RBP= 000400
T19MSK  060036          T20WFM  066242          T39BSO  075740          UNREC   = 000006        X1.SPA= 040000
T19PAC  062310          T20WMI  066262          T39BS1  075741          USI     004123          X1.UNC= 000002
T19PK2  062460          T20WNP  066146          T39BS2  075742          WAITF   016250 G        X2.BUF= 000100
T19PRE  057774          T202SS  065246          T39DAT  077572          WC.IFA= 000200          X2.EXT= 000200
T19RFI  062046          T203SS  065312          T39DLY  075724          WC.IFE= 000002          X2.OPM= 100000
T19RSF  062000          T204SS  065357          T39DSW  076450          WC.IGO= 000001          X2.RCE= 040000
T19RST  061630          T205SS  065422          T39DTA  076440          WC.IRE= 000010          X2.REV= 000077
T19SET  062220          T206SS  065466          T39EAI  076446          WC.IRW= 000004          X2.SPA= 035400
T19SEX  062162          T208SS  065531          T39ETN  076651          WC.IOT= 000100          X2.UNI= 000007
T19SNP  062022          T23A    003144 G        T39ETS  076562          WC.I1T= 000040          X2.WCF= 002000
T19SRD  061760          T23B    003146 G        T39MCI  077513          WC.I5R= 000020          X3.DCK= 000010
```

```
TSV6 - PARAMETER CODING MACRO M1113  25-MAY-82 09:19  PAGE 127-5
SYMBOL TABLE                                                                    SEQ 0297

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

. ABS.  102004      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  31032 WORDS  ( 122 PAGES)
DYNAMIC MEMORY:  20346 WORDS  ( 78 PAGES)
ELAPSED TIME:  00:38:31
CVTSBA0,CVTSBA0=SVC/ML,TSV1B,TSV22B,TSV3B,TSV4,TSV55B,TSV6
```