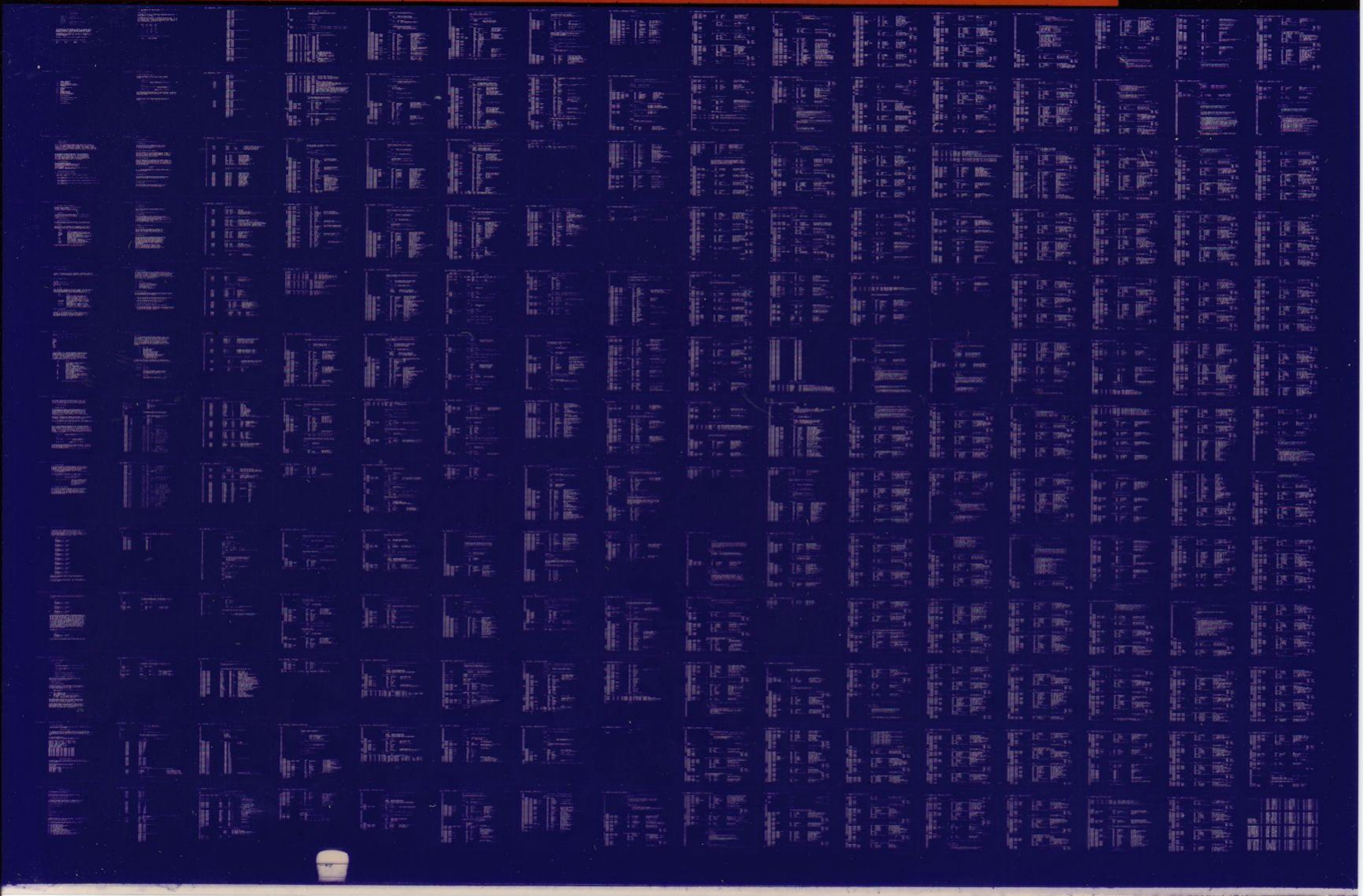
TSV05 TSV05 CTRL LTZ
CVTSBCO

AH-T096C-MC
FICHE 01 OF 02

APR 1985
digital
Made In USA



TSV05 TSV05 CTRL LT2
CVTSBCO

AH-T096C-MC
FICHE 02 OF 02

APR 1985
digital
Made In USA

.REM

IDENTIFICATION

PRODUCT ID: AC-T095C-MC

PRODUCT TITLE: CVTSBCO TSV05 CTRL LT2

DECO/DEPO: 1.0

DEPARTMENT: COMPUTER SPECIAL SYSTEMS/PPG

DATE: JUNE 4, 1984

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 (MSAAA.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 (0) ?  8<CR>

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2.7 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

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

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

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

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

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

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

BASIC EPROR 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 REWIND 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:PNI:HOE:UAM

 UNITS (D) ?  1

 UNIT 0

 DEVICE ADDRESS  (O) 172520 ? <CR>

 VECTOR (O) 224 ? <CR>
 CHANGE SW (L) ? N<CR>
```

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

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

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


0 ERRORS
```

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

PROGRAM RUN TIMES

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

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

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

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

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

5.0 DEVICE INFORMATION TABLES

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

```
        CHANGE HW (L) ?

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

        UNIT 0

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

        VECTOR (0) 224 ?          <ENTER ADDRESS OF INTERRUPT
                                  VECTOR>
```

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

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

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

6.0 TEST SUMMARIES

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

TEST DESCRIPTION:

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

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

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

TEST 3: DMA MEMORY ADDRESSING

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

****

CAUTION

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

****

TEST 4: RAM EXERCISER TEST

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

TEST 5: EXTENDED FEATURES SWITCH AND TIMERS A,B

TEST DESCRIPTION:

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

TEST 6: FIFO EXERCISER

TEST DESCRIPTION:

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

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

TEST DESCRIPTION:

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

TEST 8: TRANSPORT BUS INTERFACE LOOPBACK TEST

TEST DESCRIPTION:

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

TEST 9: READ/WRITE DATA PARITY TEST

TEST DESCRIPTION:

THIS TEST VERIFIES THAT THE WRITE DATA PARITY GENERATOR
AND THE READ DATA PARITY CHECKER OPERATE PROPERLY. THE
TRANSPORT BUS SIGNAL LOOPBACK MODE IS ENABLED AND A
SET WRONG PARITY FUNCTION IS EXECUTED. THEN VARIOUS
WRITE SUBSYSTEM MEMORY FUNCTIONS ARE PERFORMED TO
WRITE DATA TO AND FROM THE FIFO IN LOOPBACK MODE.
THE PROGRAM THEN CHECKS TO INSURE A READ DATA PARITY
ERROR OCCURRED.
A RESET FIFO IS DONE AND THE READ DATA PARITY
ERROR BIT IS AGAIN TESTED TO INSURE IT CLEARED.
FINALLY A CLEAR WRONG PARITY FUNCTION IS DONE
AND IT IS VERIFIED THE DATA WORD CAN PASS IN LOOPBACK
MODE WITHOUT SETTING READ DATA PARITY ERROR.

TEST 10: MANUAL INTERVENTION

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

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

TEST 11: CONFIGURATION TYPEOUT

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

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

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

3.0 MICROCODE REVSION LEVEL OF THE M7196.

4.0 NUMBER OF TAPE TRANSPORTS CONNECTED TO THE CONTROLLER.

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

TEST 12: SCOPE LOOPS

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

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

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

7.0 MAINTENANCE HISTORY

REVISION A - MARCH 1982

REVISION B - APRIL 1983

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

REVISION C - JUNE 1984

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

H2


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

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

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

002000                                  POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
002000                                  HEADER  CVTSB,C,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  103                             .ASCII  /C/
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  101322                          .WORD   L$HARD
002020                          L$SPCP::                        ;POINTER TO S.W. QUES.
002020  101454                          .WORD   L$SOFT
002022                          L$HPTP::                        ;PTR. TO DEF. H.W. PTABLE
002022  002156                          .WORD   L$HW
002024                          L$SPTP::                        ;PTR. TO S.W. PTABLE
002024  002166                          .WORD   L$SW
002026                          L$LADP::                        ;DIAG. END ADDRESS
002026  102004                          .WORD   L$LAST
002030                          L$STA::                         ;RESERVED FOR APT STATS
002030  000000                          .WORD   0
002032                          L$CO::
002032  000000                          .WORD   0
002034                          L$DTYP::                        ;DIAGNOSTIC TYPE
002034  000000                          .WORD   0
002036                          L$APT::                         ;APT EXPANSION
002036  000000                          .WORD   0
002040                          L$DTP::                         ;PTR. TO DISPATCH TABLE
002040  002124                          .WORD   L$DISPATCH
002042                          L$PRIO::                        ;DIAGNOSTIC RUN PRIORITY
```

PROGRAM HEADER

```
002042  000000                  .WORD   0
002044                  L$ENVI::                ;FLAGS DESCRIBE HOW IT WAS SETUP
002044  000000                  .WORD   0
002046                  L$EXP1::                ;EXPANSION WORD
002046  000000                  .WORD   0
002050                  L$MREV::                ;SVC REV AND EDIT #
002050     003                  .BYTE   C$REVISION
002051     003                  .BYTE   C$EDIT
002052                  L$EF::                  ;DIAG. EVENT FLAGS
002052  000000                  .WORD   0
002054  000000                  .WORD   0
002056                  L$SPC::
002056  000000                  .WORD   0
002060                  L$DEVP::                ; POINTER TO DEVICE TYPE LIST
002060  003402                  .WORD   L$DVTYP
002062                  L$REPP::                ;PTR. TO REPORT CODE
002062  022620                  .WORD   L$RPT
002064                  L$EXP4::
002064  000000                  .WORD   0
002066                  L$EXP5::
002066  000000                  .WORD   0
002070                  L$AUT::                 ;PTR. TO ADD UNIT CODE
002070  022306                  .WORD   L$AU
002072                  L$DUT::                 ;PTR. TO DROP UNIT CODE
002072  022404                  .WORD   L$DU
002074                  L$LUN::                 ;LUN FOR EXERCISERS TO FILL
002074  000000                  .WORD   0
002076                  L$DESP::                ;POINTER TO DIAG. DESCRIPTION
002076  003410                  .WORD   L$DESC
002100                  L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
002100  104035                  EMT     E$LOAD
002102                  L$ETP::                 ;POINTER TO ERRTBL
002102  000000                  .WORD   0
002104                  L$ICP::                 ;PTR. TO INIT CODE
002104  021512                  .WORD   L$INIT
002106                  L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  022572                  .WORD   L$CLEAN
002110                  L$ACP::                 ;PTR. TO AUTO CODE
002110  022512                  .WORD   L$AUTO
002112                  L$PRT::                 ;PTR. TO PROTECT TABLE
002112  021502                  .WORD   L$PROT
002114                  L$TEST::                ;TEST NUMBER
002114  000000                  .WORD   0
002116                  L$DLY::                 ;DELAY COUNT
002116  000000                  .WORD   0
002120                  L$HIME::                ;PTR. TO HIGH MEM
002120  000000                  .WORD   0
```

```
                                        .SBTTL  DISPATCH TABLE

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

                                        DISPATCH 12
002122  000014                          .WORD   12
002124                          L$DISPATCH::
002124  023402                          .WORD   T1
002126  024366                          .WORD   T2
002130  026360                          .WORD   T3
002132  031754                          .WORD   T4
002134  034544                          .WORD   T5
002136  040336                          .WORD   T6
002140  050450                          .WORD   T7
002142  051730                          .WORD   T8
002144  062556                          .WORD   T9
002146  066626                          .WORD   T10
002150  074470                          .WORD   T11
002152  077642                          .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.
                            ;--
                                    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
```

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


                                BGNMOD  TSV3
002176                  TSV3::
002176
                                .SBTTL  GLOBAL EQUATES SECTION

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

                                EQUALS          ; GET STANDARD EQUATES.
002176
                        ;
                        ; 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
```

N2


```
                        ;
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                  MOE== 100000
    33
    34 002176                   KT11    ..                  ;DEFINE MEMORY MANAGEMENT REGISTERS
                         SBTTL  MEMORY MANAGEMENT DEFINITIONS
                        ;*KT11 VECTOR ADDRESS
000250                  MMVEC=  250
                        ;*KT11 STATUS REGISTER ADDRESSES
177572                  SR0=    177572
177574                  SR1=    177574
177576                  SR2=    177576
172516                  SR3=    172516
                        .IF NB
                        ;*USER "I" PAGE DESCRIPTOR REGISTERS
                        UIPDR0= 177600
                        UIPDR1= 177602
                        UIPDR2= 177604
                        UIPDR3= 177606
                        UIPDR4= 177610
                        UIPDR5= 177612
                        UIPDR6= 177614
                        UIPDR7= 177616
                        .IF NB
                        ;*USER "D" PAGE DESCRIPTOR REGISTORS
                        UDPDR0= 177620
                        UDPDR1= 177622
                        UDPDR2= 177624
                        UDPDR3= 177626
                        UDPDR4= 177630
                        UDPDR5= 177632
                        UDPDR6= 177634
                        UDPDR7= 177636
                        .ENDC
```

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

                                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          ;I3RESV3 - 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
                        ;
                                MSA.NOP = 0*2           ;NO-OP (NOTHING TRIGGERED)
000000                          MSA.VOL = 1*2           ;SIMULATE ON-LINE/OFF-LINE TRANSISTION
000002                          MSA.NRAM= 2*2           ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
000004                          MSA.FRAM= 3*2           ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
000006
                        ;*
                        ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
                        ;-
                        NP.IR           = BIT7          ;INTERRUPT REQUEST (0-1 TRANSITION)
000200                  NP.OUT          = BIT6          ;TAPE DATA DIRECTION OUT (0= IN)
000100                  NP.LOOP         = BIT5          ;ENABLE TRANSPORT LOOPBACK
000040                  NP.WRP          = BIT4          ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
000020                  ;*
                        ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
                        ;-

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

```
                        .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   PRI04   ;INTERRUPT PRIORITY.
002214  000000                  TSTCNT::        .WORD   0       ;NUMBER OF TESTS RUN IN THIS PASS
002216  000000                  LOOPCNT::       .WORD   0       ;REMAINING ITERATION COUNT FOR TEST
002220  000000                  DEVCNT::        .WORD   0       ;NUMBER OF DEVICE UNDER TEST
002222  000000                  FATFLG::        .WORD   0       ;SET IF FATAL ERROR IS DETECTED IN TEST
002224  000000                  INTRECV::       .WORD   0       ;SET IF TAPE INTERRUPT WAS RECEIVED
002226  000000                  EXTFEA::        .WORD   0       ;EXTENDED FEATURES SOFTWARE SW 0=OFF;1=ON
002230  000000                  BENBSW::        .WORD   0       ;BUFFER ENABLE SWITCH SW 0=OFF;1=ON
002232  000000                  EXPD::          .WORD   0       ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
002234  000000                  RECV::          .WORD   0       ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
002236  000000                  ERRHI::         .WORD   0       ;HIGH ADDRESS MEMORY ERROR
002240  000000                  ERRLO::         .WORD   0       ;LOW ADDRESS MEMORY ERROR
002242                          RAMDATA::       .BLKW   16.     ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
002302  000000                  RAMSIZ::        .WORD   0       ;RAM DATA SIZE FOR PRAMPKT ROUTINE
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                  T38FLG::        .WORD   0               ;TEST 38 FLAG +0
003152  002000                  PST32W::        .WORD   2000            ;32W BLOCK ADDRESS FOR 32K START
003154  000000                  SIFLAG::        .WORD   0               ;
003156  000000                  BADDAT::        .WORD   0               ;ACTUAL DATA
003160  000000                  GDDAT::         .WORD   0               ;EXPECTED DATA
003162  000000                  LOOPFL::        .WORD   0
003164                          CTAB::                                  ;CONFIGURATION TABLES.
003164  000000                  CTABM::         .WORD   0               ;CONFIG WORK
003166  000000                                  .WORD   0
003170  000000                                  .WORD   0
003172  000000                                  .WORD   0
003174  177777                                  .WORD   -1              ;END OF MEM TABLE.
003176                          CTABE::
                                ;ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
                                ;
                                ;       0       =       UNIT NOT TESTED
                                ;       100000  =       UNIT ONLINE, NO ERRORS
                                ;       10XXXX  =       UNIT ONLINE, ENCOUNTERED XXXX ERRORS
                                ;       160000  =       UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
                                ;       160001  =       UNIT DROPPED, NOT IDLE AT START
                                ;       14XXXX  =       UNIT DROPPED, ENCOUNTERED XXXX ERRORS
                                ;
003176                          ERTABL:         .BLKW   64.
003376  000000                  ERTABE:         .WORD   0

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

```
                                        .SBTTL  GLOBAL TEXT MESSAGES
                                ;**
                                ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
                                ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
                                ; MORE THAN ONE TEST.
                                ; -
                                ;*
                                ;NAMES OF DEVICES SUPPORTED
                                ;-
003402                                  DEVTYP  <TSV05>
003402                          L$DVTYP::
003402  124     123     126             .ASCIZ  /TSV05/
                                        .EVEN

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

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

C4


```
004300    045     101     040  INTX:    .ASCIZ  /%A  CPU PC: %O6%A  TSBA: %O6/
004335    040     040     042  NOINIT:  .ASCIZ  /  "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
004407    040     040     042  NSINIT:  .ASCIZ  /  "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
004457    040     040     042  BRINIT:  .ASCIZ  /  "BUS-RESET" DIDN'T INITIALIZE THE DPU/
004527    000                  NUL:     .ASCIZ  //
004530    045     116     000  NULCR:   .ASCIZ  /%N/
004533    045     101     040  EXPGOT:  .ASCIZ  /%A EXP'D: %O6%A, REC'D: %O6/
004567    045     116     045  EXPGT2:  .ASCIZ  /%N%A EXP'D: %O6%A, %O6%N%A REC'D: %O%A, %O6/
004643    045     101     040  DUAD12:  .ASCIZ  /%A REG(W) WRITTEN TO: %O6%A REG(R) READ: EXP'D: %O6%A, REC'D: %O6/
004745    122     101     115  PKTRAM::         .ASCIZ  'RAM Contents Do Not Match Packet Sent'
005013    040     040     103  SCME:    .ASCIZ  /  CONFIG DOESN'T MATCH MFG. MASTER/
005056    127     122     111  WRTMSG:  .ASCIZ  'WRITE CHARACTERISTICS Failed'
005113    124     123     123  WRTERR:  .ASCIZ  'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
005206    124     123     123  RDERR:   .ASCIZ  'TSSR Incorrect After READ Command, More Bits Set Than SSR'
005300    106     101     124  SCHERR:  .ASCIZ  'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
005372    105     122     122  RETERR:  .ASCIZ  'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
005460    045     116     045  NOMEM:   .ASCIZ  '%N%A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****%N'
005554    045     116     045  M8186:   .ASCIZ  '%N%A ***************** 11/23A SYSTEM ******************%N'
005645    045     116     045  M8189:   .ASCIZ  '%N%A ***************** 11/23B SYSTEM ******************%N'
                                        .EVEN
                                        .SBTTL  GLOBAL ERROR REPORT SECTION

                               ;++
                               ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
                               ; CALLS THAT ARE USED IN MORE THAN ONE TEST.
                               ; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
                               ;--
005736                                  BGNMSG  NXRERR          ;NON-EXISTANT DEVICE REGISTER.
005736                         NXRERR::
005736                                  PRINTX  #NXRX,NODEV     ;NODEV = NEXM ADDRESS.
005736  013746  003114                  MOV     NODEV,-(SP)
005742  012746  003777                  MOV     #NXRX,-(SP)
005746  012746  000002                  MOV     #2,-(SP)
005752  010600                          MOV     SP,R0
005754  104415                          TRAP    C$PNTX
005756  062706  000006                  ADD     #6,SP
005762  004737  005770                  JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
005766                                  ENDMSG
005766                         L10002:
005766  104423                          TRAP    C$MSG
                               ;
                               ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
                               ; TO ANY OF THE ABOVE ERROR SIGNATURES.
                               ;
005770  005727                 EXTEND: TST      (PC)+
005772  000000                 EXTA:    0                       ; 0 = NO EXTENSION.
005774  001402                          BEQ     1$
005776  004777  177770                  JSR     PC,@EXTA        ; APPEND EXTENSION TEXT.
006002                         1$:      PRINTX  #NULCR          ; PRINT A BLANK LINE
006002  012746  004530                  MOV     #NULCR,-(SP)
006006  012746  000001                  MOV     #1,-(SP)
006012  010600                          MOV     SP,R0
006014  104415                          TRAP    C$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$PNTX
        006200  062706  000006                  ADD     #6,SP
    718
    719 006204  010403              20$:        MOV     R4,R3           ;GET THE TSSR CONTENTS
    720 006206  042703  177761                  BIC     #^CTERCLS,R3    ;CLEAR ALL BUT TERMINATION
    721 006212  016303  006676                  MOV     TCOCOD(R3),R3   ;GET THE TERMINATION CODE MEANING
    722 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$PNTX
        006234  062706  000006                  ADD     #6,SP
    723 006240  010403                          MOV     R4,R3           ;TSSR CONTENTS AGAIN
    724 006242  042703  177717                  BIC     #^CFATERR,R3    ;CLEAR ALL BUT FATAL TERMINATION
    725 006246  001416                          BEQ     25$             ;DON'T PRINT IF ZERO
    726 006250  006203                          ASR     R3
    727 006252  006203                          ASR     R3
    728 006254  006203                          ASR     R3              ;ALINE TERMINATION CODE FOR INDEX
    729 006256  016303  007236                  MOV     TSFCOD(R3),R3   ;GET THE FATAL TERMINATION CODE
    730 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$PNTX
        006300  062706  000006                  ADD     #6,SP
    731 006304  042704  176377          25$:    BIC     #^CHIADDR,R4    ;CLEAR ALL BUT EXTENDED ADDRESS
    732 006310  001411                          BEQ     30$             ;DON'T PRINT IF ZERO
    733 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$PNTX
        006330  062706  000006                  ADD     #6,SP
    734 006334  013703  002200          30$:    MOV     EPRTSW,R3               ;PRINT MEAASGE BUFFER ADDRESS
    735 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
    736 006356  000207                          RTS     PC              ;RETURN TO CALLER
```

```
006360                                 EPRT2:
006360     045     116     045         EPRT1:  .ASCIZ  '%N%A *****REPLACE M7196*****

006415     045     116     045         TSSRFOR:        .ASCIZ  '%N%A TSSR = %06'
006435     045     116     045         TEXASC:         .ASCIZ  '%N%A Extended Address Bits = %06'
006476     045     116     045         TCOASC:         .ASCIZ  '%N%A Termination Class Code = %T'
006537     045     116     045         TFCASC:         .ASCIZ  '%N%A Fatal Termination Class Code = %T
006606     045     116     045         TSSDEF:         .ASCIZ  '%N%A TSSR Bits Set: %T'
006635     045     116     045         AMBTSSR:        .ASCIZ  '%N%A TSSR Contents Are Ambiguous'
                                               .EVEN
006676  006716  006741  006767         TCOCOD: .WORD   1$,2$,3$,4$,5$,6$,7$,8$
006716     116     157     162         1$:     .ASCIZ  'Normal Termination'
006741     124     145     162         2$:     .ASCIZ  'Termination Condition'
006767     124     141     160         3$:     .ASCIZ  'Tape Status Alert'
007011     106     165     156         4$:     .ASCIZ  'Function Reject'
007031     122     145     143         5$:     .ASCIZ  'Recoverable Error - Tape Position One Record Down
007113     122     145     143         6$:     .ASCIZ  'Recoverable Error - Tape Was Not Moved'
007162     125     156     162         7$:     .ASCIZ  'Unrecoverable Error'
007206     106     141     164         8$:     .ASCIZ  'Fatal Controller Error'
                                               .EVEN

007236  007246  007302  007313         TSFCOD: .WORD   1$,2$,3$,4$
007246     111     156     164         1$:     .ASCIZ  'Internal Diagnostic Failure'
007302     122     145     163         2$:     .ASCIZ  'Reserved'
007313     102     165     163         3$:     .ASCIZ  'Bus Interface or Sanity Check Error'
007357     122     145     163         4$:     .ASCIZ  'Reserved'
                                               .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
```

```
TSV3 - GLOBAL AREAS     MACRO M1113  14 JUN 84 15:55                                SEQ 0047
PRIXOR    PRINT EXPD, RECV AND XOR

        007760  010346                          MOV     R3,-(SP)
        007762  010146                          MOV     R1,-(SP)
        007764  010246                          MOV     R2,-(SP)
        007766  012746  010012                  MOV     #XORFOR,-(SP)
        007772  012746  000004                  MOV     #4,-(SP)
        007776  010600                          MOV     SP,R0
        010000  104414                          TRAP    C$PNTB
        010002  062706  000012                  ADD     #12,SP
    872 010006  010300                          MOV     R3,R0           ;R0 HAS XOR ON RETURN
    873 010010  000207                          RTS     PC              ;RETURN TO CALLER
    874
    875 010012      045     116     045  XORFOR: .ASCIZ  '%N%A EXPD: %O6%A RECV: %O6%A XOR: %O6'
    876                                          .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 = %O6'
                                        .EVEN
```

```
                                        .SBTTL  PRIADD    PRINT MEMORY ERROR ADDRESS
                                ;+
                                ;
                                ;PRINT MEMORY ADDRESS
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                                ;
                                ; IMPLICIT INPUTS
                                ;
                                ;       ERRHI   - HIGH ORDER ADDRESS
                                ;       ERRLO   - LOW ORDER ADDRESS
                                ;
                                ;
010160                          PRIADD:
                                        SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
010160  013700  002236                  MOV     ERRHI,R0        ;GET HIGH ADDRESSS
010164  013701  002240                  MOV     ERRLO,R1        ;GET LOW ADDRESS
010170  010102                          MOV     R1,R2           ;COPY LOW ADDRESS
010174  006101                          ROL     R1              ;SHIFT BIT 15 TO C BIT
010176  006100                          ROL     R0              ;SHIFT INTO HIGH ORDER
010200                                  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:
                                        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
    960 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
    961 010354  000207                          RTS     PC              ;RETURN
    962
    963 010356     045     116     045  PRIT0:  .ASCIZ  '<NB>A MEMORY TEST ADDRESS LOW = <06>
    964 010421     045     116     045  PRIT1:  .ASCIZ  '<NB>A MEMORY TEST ADDRESS HIGH = <06>
    965                                         .EVEN
```

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

                        ;+
                        ;
                        ;ROUTINE TO ISSUE A SPACE RECORDS
                        ;COMMAND (FORWARD OR REVERSE)
                        ;
                        ;INPUT:
                        ;
                        ;       R3      NUMBER OF RECORDS TO BE SPACED OVER
                        ;               BIT15 CONTROLS DIRECTION
                        ;               BIT15 = 0 IS FORWARD
                        ;               BIT15 = 1 IS REVERSE
                        ;       R5      FIRST DEVICE UNIBUS ADDRESS
                        ;
                        ;       REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
                        ;
                        ;OUTPUT:
                        ;
                        ;       CARRY   SET - SPACE RECORDS COMMAND OK
                        ;               CLR - SPACE RECORDS FAILED
                        ;
                        ;
                        ;       R0      THE CONTENTS OF R4 IS MOVED TO R0
                        ;
                        ;
                        ;IMPLICIT OUTPUT:
                        ;
                        ;       TAPE HAS BEEN MOVED
                        ;
                        ;SIDE EFFECTS:
                        ;
                        ;
                        ;-

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

```
        010576  005367  177756                  DEC     -22(PC)
        010602  001367                          BNE     .-20
  1018  010604  005337  010660                  DEC     SDELAY                  ;BUMP DELAY COUNTER DOWN
  1019  010610  001356                          BNE     15$                     ;BR. IF MORE DELAY
  1020  010612  000411                          BR      60$                     ;BR IF TROUBLE CARRY = CLEAR
  1021  010614  016501  000002        20$:     MOV     TSSR(R5),R1             ;READ TSSR
  1022  010620  012702  000200                  MOV     #SSR,R2                 ;SET UP EXPECTED
  1023  010624  020201                25$:     CMP     R2,R1                   ;ARE THEY OK
  1024  010626  001401                          BEQ     40$                     ;BR. IF EQUAL = OK
  1025  010630  000402                          BR      60$                     ;TROUBLE EXIT
  1026  010632  000261                40$:     SEC                             ;SET CARRY NO TROUBLE
  1027  010634  000401                          BR      70$                     ;EXIT
  1028  010636  000241                60$:     CLC                             ;CARRY CLEAR = ERROR
  1029  010640                        70$:
  1030  010640  010400                          MOV     R4,R0                   ;PASS PACKET ADDRESS
  1031  010642  000207                          RTS     PC                      ;RETURN
  1032                                ;
  1033                                ;
  1034                                ;
  1035                                ;PACKET FOR SPACE COMMAND
  1036                                ;
  1038          010650                        .=<.+10>&177770
  1040                                ;
  1041                                ;COMMAND WORD
  1042  010650  000000                80$:     .WORD
  1043                                ;NUMBER OF RECORDS TO BE SPACED OVER WORD
  1044  010652  000000                90$:     .WORD
  1045  010654  000000                         .WORD
  1046  010656  000000                         .WORD
  1047  010660  000000                SDELAY:  .WORD   0                       ;DELAY COUNTER
  1048                                         .EVEN
  1049                                         .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:
                                ;-
1074 010662                     WRTCHR::
1075 010662                             SAVREG                          ;SAVE THE GENERAL REGISTERS
1076 010666  005037  002230             CLR     BENBSW          ;CLEAR BUFFER ENABLE SWITCH
1077 010672  005037  002226             CLR     EXTFEA                  ;CLEAR EXTENDED FEATURES SW SWITCH
1078 010676  010465  000000     10$:    MOV     R4,TSDB(R5)             ;SEND OUT COMMAND
1079 010702  004737  016336             JSR     PC,CHKTSSR              ;WAIT FOR SSR
1080 010706  103401                     BCS     20$                     ;BR. IF SSR IS SET AND OK
1081 010710  000435                     BR      60$                     ;BR IF TROUBLE CARRY = CLEAR
1082 010712  016501  000002     20$:    MOV     TSSR(R5),R1             ;READ TSSR
1083 010716  012702  000200             MOV     #SSR,R2                 ;SET UP EXPECTED
1084 010722  032701  000100             BIT     #OFL,R1                 ;WAS OFF LINE SET IN TSSR
1085 010726  001402                     BEQ     25$                     ;BR. IF NO OFL SET
1086 010730  052702  000100             BIS     #OFL,R2                 ;MAKE THEM LOOK ALIKE
1087 010734  020201     25$:    CMP     R2,R1                   ;ARE THEY OK
1088 010736  001401                     BEQ     40$                     ;BR. IF EQUAL = OK
1089 010740  000421                     BR      60$                     ;TROUBLE EXIT
1090 010742  062704  000010     40$:    ADD     #8.,R4                  ;POINT TO WRT CHARA DATA PACKET
1091 010746  011403                     MOV     (R4),R3                 ;GET ADDRESS OF MESSAGE BUFFER
1092 010750  032763  000200  000012     BIT     #X2.EXTF,XST2(R3)       ;EXTENDED FEATURES BIT SET?
1093 010756  001402                     BEQ     45$                     ;BR IF NO
1094 010760  005237  002226             INC     EXTFEA                  ;SET EXTENDED FEATURES SW SWITCH
1095 010764                     45$:
1096 010764  032763  000100  000012     BIT     #X2.BUFE,XST2(R3)       ;BUFFER ENABLE SWITCH SET
1097 010772  001402                     BEQ     50$                     ;BR. IF SWITCH NOT SET
1098 010774  005237  002230             INC     BENBSW                  ;SET SOFTWARE SWITCH FOR ENABLED
1099 011000                     50$:
1100 011000  000261                     SEC                             ;SET CARRY NO TROUBLE
1101 011002  000401                     BR      70$                     ;EXIT
1102 011004  000241     60$:    CLC                             ;CARRY CLEAR = ERROR
1103 011006  016500  000002     70$:    MOV     TSSR(R5),R0             ;RETURN TSSR CONTENTS
1104 011012  000207                     RTS     PC                      ;RETURN
```

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

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

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

011114                  CKRAM::
011114                          SAVREG                          ;SAVE THE GENERAL REGISTERS
011120  012701  002242          MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
011124  012702  000201          MOV     #RMPKTBEG,R2            ;BYTE ADDRESS OF FIRST RAM DATA
011130  005003                  CLR     R3                      ;CLEAR THE ERROR FLAG
011132  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR
011136  112765  000000  000000  MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
011144  004737  016336  10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011150  010265  000000          MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
011154  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011160  116511  000000          MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
011164  122124                  CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
011166  001401                  BEQ     20$                     ;BRANCH IF OK
011170  005203                  INC     R3                      ;SET ERROR FLAG
011172  005202          20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
011174  020227  000210          CMP     R2,#RMPKTEND            ;REACHED END YET ?
011200  003761                  BLE     10$                     ;BRANCH TILL ALL READ
011202  005703                  TST     R3                      ;WAS AN ERROR FOUND ?
011204  001402                  BEQ     30$                     ;BRANCH IF NOT
011206  000241                  CLC                             ;CLEAR CARRY TO SHOW ERROR
011210  000401                  BR      50$                     ;AND EXIT
011212  000261          30$:    SEC                             ;SHOW GOOD COMPARE
011214  012737  000010  002302  50$:    MOV     #8.,RAMSIZ      ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
011222  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:
                        ;
                        ;       P4      ADDRESS OF THE CHARACTERISTICS DATA
                        ;       R5      FIRST DEVICE UNIBUS ADDRESS
                        ;
                        ;OUTPUT:
                        ;
                        ;       CARRY   SET - RAM MATCHES PACKET
                        ;               CLR - RAM DOES NOT MATCH PACKET
                        ;
                        ;IMPLICIT OUTPUT:
                        ;
                        ;       THE TABLE RAMDATA IS FILLED WITH THE
                        ;       DATA HELD IN RAM.
                        ;       RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
                        ;
                        ;SIDE EFFECTS:
                        ;
                        ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
                        ;-
011224                  CKRAM2::
011224                          SAVREG                          ;SAVE THE GENERAL REGISTERS
011230  012701  002242          MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
011234  012702  000167          MOV     #RMCHBEG,R2             ;BYTE ADDRESS OF FIRST RAM DATA
011240  005003                  CLR     R3                      ;CLEAR THE ERROR FLAG
011242  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR
011246  112765  000000  000000  MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
011254  004737  016336  10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011260  010265  000000          MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
011264  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011270  116511  000000          MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
011274  122124                  CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
011276  001401                  BEQ     20$                     ;BRANCH IF OK
011300  005203                  INC     R3                      ;SET ERROR FLAG
011302  005202          20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
011304  012737  000010  002302  MOV     #8.,RAMSIZ              ;ASSUME EXTFEA NOT SET
011312  005737  002226          TST     EXTFEA                  ;IS THE SOFTWARE EXTENDED FEATURES SET
011316  001407                  BEQ     25$                     ;BR. IF NOT SET
011320  012737  000012  002302  MOV     #10.,RAMSIZ             ;SET RAMSIZ FOR EXTEND FEATURES
011326  020227  000200          CMP     R2,#RMCHEND             ;AT END OF EXTENDED BUFFER
011332  003750                  BLE     10$                     ;BR. IF NOT AT END YET
011334  000403                  BR      27$                     ;AT END BRANCH
011336  020227  000176  25$:    CMP     R2,#RMCHEND-2           ;REACHED END YET ?
011342  003744                  BLE     10$                     ;BRANCH TILL ALL READ
011344  005703          27$:    TST     R3                      ;WAS AN ERROR FOUND ?
011346  001402                  BEQ     30$                     ;BRANCH IF NOT
011350  000241                  CLC                             ;CLEAR CARRY TO SHOW ERROR
011352  000401                  BR      50$                     ;AND EXIT
011354  000261          30$:    SEC                             ;SHOW GOOD COMPARE
011356  000207          50$:    RTS     PC                      ;RETURN
```

```
                                        .SBTTL  CKMSG   - COMPARE WRITE CHAR. MESSAGE BUFFERS
                                ;*
                                ;
                                ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
                                ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
                                ;ERROR PRINT POUTINES.
                                ;
                                ;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::
                                        SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011500  020327  000144                  CMP     R3,#RECMSG-EXPMSG;@@D IS COUNT ABOVE MAX ALLOWED?
011504  003412                          BLE     5$                      ;@@D BR IF NO
011506  012703  000144                  MOV     #RECMSG-EXPMSG,R3;@@D
011512                                  PRINTF  #DEBUGMSG               ;@@D
011512  012746  011632                  MOV     #DEBUGMSG,-(SP)
011516  012746  000001                  MOV     #1,-(SP)
011522  010600                          MOV     SP,R0
011524  104417                          TRAP    C$PNTF
011526  062706  000004                  ADD     #4,SP
011532  010037  002304          5$:     MOV     R0,RCVHIADD             ;SAVE RECV HIGH ADDRESS
011536  010137  002306                  MOV     R1,RCVLOAD              ;SAVE RECV LOW ADDRESS
011542  005737  003134                  TST     KTENABLE                ;TESTING ABOVE 28K?
011546  001403                          BEQ     10$                     ;BR IF NO
011550  004737  017316                  JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
011554  010001                          MOV     R0,R1                   ;GET RETURNED ADDRESS BIASED TO PAR6
011556  005004          10$:    CLR     R4                      ;WORD IN BUFFER
011560  005005                  CLR     R5                      ;CLEAR ERROR SEEN FLAG
011562  111264  002322  15$:    MOVB    (R2),EXPMSG(R4)         ;SAVE EXPD FOR ERROR REPORT
011566  111164  002466          MOVB    (R1),RECMSG(R4)         ;SAVE RECV FOR ERROR REPORT
011572  122221                  CMPB    (R2)+,(R1)+             ;EXPD EQUAL RECV?
011574  001401                  BEQ     25$                     ;BR IF YES
011576  005205                  INC     R5                      ;SET ERROR SEEN FLAG
011600  062704  000001  25$:    ADD     #1,R4                   ;POINT TO NEXT BYTE
011604  020403                  CMP     R4,R3                   ;DONE ALL BYTES?
011606  002001                  BGE     50$                     ;BR IF YES
011610  000764                  BR      15$                     ;DO NEXT BYTE
011612  005705          50$:    TST     R5                      ;ANY ERRORS SEEN?
011614  001402                  BEQ     55$                     ;BR IF NO
011616  000241                  CLC                             ;SET FAILURE
011620  000401                  BR      60$                     ;
011622  000261          55$:    SEC                             ;SET SUCCESS
011624  000207          60$:    RTS     PC                      ;RETURN
```

```
1371 011632      120      122      117  DEBUGMSG: .ASCIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED  ;@@D
1372 011722      045      116      045  FERCM:   .ASCII  /%N%A  ***/
1373 011733      040      040      124  ERCM.    .ASCIZ  /  TSSR ERROR CODE REC'D = /
1374 011766      056      056      056  SIMSG:   .ASCIZ  /.... AFTER DOING SOFT INIT/
1375 012021      124      105      123  TINERR:  .ASCIZ  /TEST: .../
1376                                            .EVEN
1377                                    ;+
1378                                    ;
1379                                    ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
1380                                    ;
1381                                    ;INPUT:
1382                                    ;
1383                                    ;       R1      CONTENTS OF TSSR AT ERROR
1384                                    ;
1385                                    ;SIDE EFFECTS:
1386                                    ;
1387                                    ;       EXECUTES DROP UNIT TO CEASE TESTING
1388                                    ;
1389                                    ;-
1390
1391 012034                                         BGNMSG  SFIMSG
     012034                             SFIMSG::
1392 012034  004737  006024                         JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1393 012040  004737  017202                         JSR     PC,CKDROP       ;DROP UNIT, IF ALLOWED
1394 012044                                         ENDMSG
     012044                             L10003:
     012044  104423                             TRAP    C$MSG
1395
1396                                    ;+
1397                                    ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1398                                    ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
1399                                    ;
1400                                    ;INPUTS:
1401                                    ;
1402                                    ;       R1      TSSR CONTENTS
1403                                    ;       R4      ADDRESS OF COMMAND PACKET
1404                                    ;
1405                                    ;-
1406
1407 012046                                         BGNMSG  PKTSSR
     012046                             PKTSSR::
1408 012046  004737  006024                         JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1409 012052  012700  000004                         MOV     #4,R0           ;NO. OF WORDS IN PACKET
1410 012056  004737  007370                         JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
1411 012062                                         ENDMSG
     012062                             L10004:
     012062  104423                             TRAP    C$MSG
```

```
                                        ;+
                                        ;PRINT ROUTINE TO PRINT THE CONTENTS OF
                                        ;TSSR AND A GET STATUS COMMAND PACKET.
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      TSSR CONTENTS
                                        ;       R4      ADDRESS OF COMMAND PACKET
                                        ;
                                                BGNMSG  PKTGETS
012064                                  PKTGETS::
012064  004737  006024                          JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
012070  012700  000002                          MOV     #2,R0           ;NO. OF WORDS IN GET STATUS PACKET
012074  004737  007370                          JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
012100                                          ENDMSG
012100                                  L10005:
012100  104423                                  TRAP    C$MSG
                                        ;+
                                        ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
                                        ;
                                        ;INPUTS:
                                        ;       R1      TSSR CONTENTS
                                        ;       R4      ADDRESS OF COMMAND PACKET
                                        ;-
                                                BGNMSG  SFFMSG
012102                                  SFFMSG::
012102  004737  006024                          JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
012106                                          ENDMSG
012106                                  L10006:
012106  104423                                  TRAP    C$MSG
                                                .SBTTL  PKTMES  - PRINT TSSR AND MESSAGE BUFFER
                                        ;+
                                        ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
                                        ;BUFFER FOR ERROR REPORTS
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      CONTENTS OF TSSR
                                        ;       R2      LOW ORDER MESSAGE BUFFER
                                        ;       R3      HIGH ORDER MESSAGE BUFFER ADDRESS
                                        ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
                                        ;-
                                                BGNMSG  PKTMES
012110                                  PKTMES::
012110  004737  006024                          JSR     PC,PRITSSR              ;PRINT CONTENTS OF TSSR
012114  010200                                  MOV     R2,R0                   ;LOW ORDER ADDRESS
012116  010301                                  MOV     R3,R1                   ;HIGH ORDER ADDRESS
012120  004737  014242                          JSR     PC,PRMESS               ;PRINT THE MESSAGE BUFFER
012124                                          ENDMSG
012124                                  L10007:
012124  104423                                  TRAP    C$MSG
```

```
1456                                            .SBTTL  ADDSSR    PRINT TEST ADDRESS AND TSSR
1457                                    ;+
1458                                    ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1459                                    ;TSSR AND A MEMORY TEST ADDRESS
1460                                    ;
1461                                    ;INPUTS:
1462                                    ;
1463                                    ;       R5      FIRST DEVICE UNIBUS ADDRESS
1464                                    ;       ERRHI   HIGH ORDER MEMORY TEST ADDRESS
1465                                    ;       ERRLO   LOW ORDER MEMORY TEST ADDRESS
1466                                    ;-
1467
1468 012126                                    BGNMSG  ADDSSR
     012126                             ADDSSR::
1469 012126  004737  010274                    JSR     PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
1470 012132  016501  000002                    MOV     TSSR(R5),R1     ;GET CURRENT TSSR
1471 012136  004737  006024                    JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1472 012142                                    ENDMSG
     012142                             L10010:
     012142  104423                            TRAP    C$MSG
1473
1474                                            .SBTTL  MSGEXP - PRINT WRITE CHAR. EXPD-RECV MESSAGE BUFFERS
1475                                    ;+
1476                                    ;
1477                                    ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
1478                                    ;
1479                                    ;IMPLICIT INPUTS:
1480                                    ;
1481                                    ;       EXPMSG  - EXPECTED MESSAGE BUFFER
1482                                    ;       RECMSG  - RECEIVED MESSAGE BUFFER
1483                                    ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1484                                    ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1485                                    ;-
1486 012144                                    BGNMSG  MSGEXP
     012144                             MSGEXP::
1487 012144  012700  000007                    MOV     #7,R0           ;ASSUME NO EXT FEATURES
1488 012150  005737  002226                    TST     EXTFEA          ;EXT FEATURES SET?
1489 012154  001402                            BEQ     5$              ;BR IF NO
1490 012156  012700  000010                    MOV     #8.,R0          ;EXT FEATURE BUFFER IS 8 WORDS
1491 012162  004737  014552            5$:     JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
1492 012166                                    ENDMSG
     012166                             L10011:
     012166  104423                            TRAP    C$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
                        MOV     R1,-(SP)
                        MOV     #FIF1MSG,-(SP)
                        MOV     #2,-(SP)
                        MOV     SP,R0
                        TRAP    C$PNTX
                        ADD     #6,SP
                        PRINTX  #FIF2MSG        ;PRINT HEADER MSG
                        MOV     #FIF2MSG,-(SP)
                        MOV     #1,-(SP)
                        MOV     SP,R0
                        TRAP    C$PNTX
                        ADD     #4,SP
                        MOV     R1,R0           ;GET BYTE COUNT
                        JSR     PC,PRBYTEXP     ;PRINT FIFO BYTES IN ERROR
                        ENDMSG
                L10012:
                        TRAP    C$MSG
045 116 045 FIF1MSG:    .ASCIZ  '<NL>A NUMBER OF BYTES TRANSFERRED = <D2>'
045 116 045 FIF2MSG:    .ASCIZ  '<NL>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
013070  012100                  10$:    MOV     (R1)+,R0        ;DONE ALL MSG LINES?
013072  001410                          BEQ     20$             ;BR IF YES
013074                                  PRINTX  R0              ;PRINT STATUS BIT NAMES
013074  010046                          MOV     R0,-(SP)
013076  012746  000001                  MOV     #1,-(SP)
013102  010600                          MOV     SP,R0
013104  104415                          TRAP    C$PNTX
013106  062706  000004                  ADD     #4,SP
013112  000766                          BR      10$             ;DO ANOTHER MSG LINE
013114  012700  000012          20$:    MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
013120  004737  014552                  JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
013124                                  ENDMSG
013124                          L10014:
013124  104423                          TRAP    C$MSG

013126  013146  013221  013320  LOOPCOD:        .WORD   1$,2$,3$,4$,5$,6$,7$,0
013146    045     116     045   1$:.ASCIZ  '<CR><LF>A Tape Bus Loopback Signals in Word #0:'
013221    045     116     045   2$:.ASCIZ  '<CR><LF>A      PARERR<15>          IRESV2<14>          IRESV1<13>'
013320    045     116     045   3$:.ASCIZ  '<CR><LF>A IHISP=>IEOT<12>    IWRT=>IIDENT<11>    IREV  =>ICER  <10>'
013417    045     116     045   4$:.ASCIZ  '<CR><LF>A IWFM =>IFMK<09>    IEDIT=>IHER <08>    IFAD  =>ISPEED<07>'
013516    045     116     045   5$:.ASCIZ  '<CR><LF>A ITAD0=>IRDY<06>    ITAD1=>IONL <05>    IERASE=>ILDP  <04>'
013615    045     116     045   6$:.ASCIZ  '<CR><LF>A IREW =>IDBY<03>    IRWU =>IRWD <02>    IFEN  =>IFBY  <01>'
013714    045     116     045   7$:.ASCIZ  '<CR><LF>A IGO  =>IFPT<00>'
                                        .EVEN
```

```
                                        .SBTTL  MSGSUB   PRINT WRITE SUBSYSTEM MESSAGE BUFFER
                                ;*
                                ;
                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
                                ;
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSC  - 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
                ;-
014776          PRAMPKT:
014776                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
014002  012701  002242          MOV     @RAMDATA,R1     ;DATA FROM THE RAM
014006  005002                  CLR     R2              ;INIT BYTE NUMBER
014010  122124          5$:     CMPB    (R1)+,(R4)+     ;COMPARE EXPECTED, RECEIVED
014012  001005                  BNE     7$              ;BR IF NO MATCH
014014                          FORCERROR       7$,NOTSSR
014024  000436                  BR      10$             ;BRD
014026  116105  177777  7$:     MOVB    -1(R1),R5       ;GET RECV RAM DATA
014032  116403  177777          MOVB    -1(R4),R3       ;GET EXPD PACKET DATA
014036                          XOR     R5,R3           ;XOR EXPD/RECV
014046  042703  177400          BIC     #177400,R3      ;LOW BYTE ONLY
014052  116137  177777  002234  MOVB    -1(R1),RECV     ;GET RECEIVED RAM DATA
014060  116437  177777  002232  MOVB    -1(R4),EXPD     ;GET EXPECTED RAM DATA
014066                          PRINTB  @RAMASC,R2,RECV,EXPD,R3
014066  010346                  MOV     R3,-(SP)
014070  013746  002232          MOV     EXPD,-(SP)
014074  013746  002234          MOV     RECV,-(SP)
014100  010246                  MOV     R2,-(SP)
014102  012746  014156          MOV     @RAMASC,-(SP)
014106  012746  000005          MOV     #5,-(SP)
014112  010600                  MOV     SP,R0
014114  104414                  TRAP    C$PNTB
014116  062706  000014          ADD     #14,SP
014122  005202          10$:    INC     R2              ;UPDATE BYTE COUNT
014124  005737  002302          TST     RAMSIZ          ;DEFAULT TO 8.?
014130  001404                  BEQ     15$             ;BR IF YES
014132  020237  002302          CMP     R2,RAMSIZ       ;DONE ALL BYTES?
014136  003724                  BLE     5$              ;BR IF NO
014140  000403                  BR      25$             ;
014142  020227  000010  15$:    CMP     R2,#8.          ;DONE DEFAULT NUMBER OF BYTES?
014146  002720          20$:    BLT     5$              ;BR IF NO
014150  005037  002302  25$:    CLR     RAMSIZ          ;SET DEFAULT RAMSIZ
014154  000207                  RTS     PC              ;RETURN

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

```
                                        .SBTTL  PRMESS  - PRINT CONTENTS OF MESSAGE BUFFER
                                ;+
                                ;THIS ROUTINE PRINTS THE CONTENTS OF
                                ;THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE TSV-05.
                                ;
                                ;INPUT:
                                ;       R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
                                ;       R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
                                ;       NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
                                ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
                                ;-
014242                          PRMESS: 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
014404  002761                          BLT     20$             ;PRINT FIRST 7 WORDS
014406  032763  000200  000012          BIT     #X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
014414  001355                          BNE     20$             ;PRINT EXTENDED STATUS WORD
014416  000207                  50$:    RTS     PC              ;RETURN
014420     045     116     045  PROASC: .ASCIZ  '%N%A Message Buffer Address = %01%05'
014465     045     116     045  PR1ASC: .ASCIZ  '%N%A Message Buffer Contents:'
014523     045     116     045  PRASC:  .ASCIZ  '%N%A    Word%01%A: %0'
```

```
                                        .EVEN
                                        .SBTTL  PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
                                ;+
                                ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
                                ;       R0      - NUMBER OF WORDS IN BUFFER
                                ;IMPLICIT INPUTS:
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;
                                PRMSGEXP::
014552                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
014552                                  MOV     R0,R5           ;SAVE NUMBER OF WORDS
014556  010005                          MOV     RCVLOADD,R0     ;GET RECV LOW ADDRESS
014560  013700  002306                  MOV     R0,R4           ;COPY LOW ADDRESS
014564  010004                          MOV     RCVHIADD,R1     ;GET RECV HIGH ADDRESS
014566  013701  002304                  ROL     R0              ;SHIFT BIT15 TO C BIT
014572  006100                          ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
014574  006101                          PRINTX  #PRMSG0,R1,R4   ;PRINT MESSAGE BUFFER ADDRESS
014576                                  MOV     R4,-(SP)
014576  010446                          MOV     R1,-(SP)
014600  010146                          MOV     #PRMSG0,-(SP)
014602  012746  014732                  MOV     #3,-(SP)
014606  012746  000003                  MOV     SP,R0
014612  010600                          TRAP    C$PNTX
014614  104415                          ADD     #10,SP
014616  062706  000010                  PRINTX  #PRMSG1         ;PRINT HEADER FOR CONTENTS
014622                                  MOV     #PRMSG1,-(SP)
014622  012746  014777                  MOV     #1,-(SP)
014626  012746  000001                  MOV     SP,R0
014632  010600                          TRAP    C$PNTX
014634  104415                          ADD     #4,SP
014636  062706  000004                  CLR     R4              ;NUMBER OF THE CURRENT WORD
014642  005004                          MOV     #EXPMSG,R1      ;GET EXPD BUFFER ADDRESS
014644  012701  002322                  MOV     #RECMSG,R2      ;GET RECV BUFFER ADDRESS
014650  012702  002466                  MOV     (R1),R0         ;GET EXPD
014654  011100                20$:      MOV     (R2),R3         ;GET RECV
014656  011203                          XOR     R0,R3           ;XOR EXPD/RECV
014660                                  PRINTX  #PRMSG2,R4,(R1)+,(R2)+,R3
014670                                  MOV     R3,-(SP)
014670  010346                          MOV     (R2)+,-(SP)
014672  012246                          MOV     (R1)+,-(SP)
014674  012146                          MOV     R4,-(SP)
014676  010446                          MOV     #PRMSG2,-(SP)
014700  012746  015035                  MOV     #5,-(SP)
014704  012746  000005                  MOV     SP,R0
014710  010600                          TRAP    C$PNTX
014712  104415                          ADD     #14,SP
014714  062706  000014                  INC     R4              ;NUMBER OF THE NEXT
014720  005204                          CMP     R4,R5           ;DONE ALL YET?
014722  020405                          BGE     50$             ;BR IF YES
014724  002001                          BR      20$             ;DO ANOTHER
014726  000752                50$:      RTS     PC              ;RETURN
014730  000207          045     116     045     PRMSG0: .ASCIZ  '<CR><LF>Message Buffer Address = <01><05>'
014732     045     116     045     PRMSG1: .ASCIZ  '<CR><LF>Message Buffer Contents:'
014777     045     116     045     PRMSG2: .ASCIZ  '<CR><LF>   WORD #<02><LF>  EXPD: <06><LF>  RECV: <06><LF>  XOR: <0
015035
```

```
                                        .EVEN
                                        .SBTTL  PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
                                ;+
                                ;
                                ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
                                ;       ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
                                ;
                                ;       R0      - NUMBER OF BYTES IN BUFFER
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;-
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$             ;BUG
015266  000404                          BR      35$                     ;BUG
015270                          30$:
015270                                  FORCERROR  27$,NOTSSR           ;BUG
015300                          35$:                                    ;BUG
015300  005204                          INC     R4                      ;NUMBER OF THE NEXT
015302  020405                          CMP     R4,R5                   ;DONE ALL YET?
015304  002001                          BGE     50$                     ;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                          TPAP    C$PNTX
     015330  062706  000006                  ADD     #6,SP
1783 015334  000207                          RTS     PC                      ;RETURN
1784
1785 015336     045     116     045  PRBMSG: .ASCIZ  '<CR><LF>  BYTE #<LF>D2<LF>  EXPD: <LF>03<LF>  RECV: <LF>03<LF>  XOR: <LF>03'
1786 015423     045     116     045  PRBTOT: .ASCIZ  '<CR><LF> NUMBER OF BYTES IN ERROR = <LF>D2'
1787                                  .EVEN
1788 015470  000000                  PRBEXP: .WORD   0                       ;EXPD
1789 015472  000000                  PRBREC: .WORD   0                       ;RECV
1790                                  .SBTTL  EXPREC  - PRINT EXPD/RECV WORD DATA
1791                                ;+
1792                                ;
1793                                ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1794                                ;
1795                                ;INPUTS:
1796                                ;
1797                                ;       R1      RECEIVED DATA
1798                                ;       R2      EXPECTED DATA
1799                                ;
1800                                ;
1801
1802 015474                          BGNMSG  EXPREC
     015474                       EXPREC::
1803 015474  004737  007742                  JSR     PC,PRIXOR               ;PRINT THE DATA
1804 015500                                  ENDMSG
     015500                       L10017:
     015500  104423                          TRAP    C$MSG
1805                                  .SBTTL  EXPBREC - PRINT EXPD/RECV BYTE DATA
1806                                ;+
1807                                ;
1808                                ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
1809                                ;
1810                                ;
1811                                ;INPUTS:
1812                                ;
1813                                ;       R1      RECEIVED DATA BYTE
1814                                ;       R2      EXPECTED DATA BYTE
1815                                ;
1816                                ;-
1817
1818 015502                          BGNMSG  EXPBREC
     015502                       EXPBREC::
1819 015502  004737  007612                  JSR     PC,PRIBXOR              ;PRINT THE DATA
1820 015506                                  ENDMSG
     015506                       L10020:
     015506  104423                          TRAP    C$MSG
1821
1822
1823                                  .SBTTL  RAMERR  - PRINT RAM AND PACKET DATA
1824                                ;+
1825                                ;
1826                                ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1827                                ;
1828                                ;INPUTS:
1829                                ;
1830                                ;       R4      POINTER TO COMMAND PACKET
1831                                ;
```

```
                                ;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::
                                        JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
                                        ENDMSG
                                L10021:
                                        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::
1870 015516  004737  010274            JSR     PC,PRITADD      ;PRINT TEST ADDRESS
1871 015522  004737  013776            JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
1872 015526                            ENDMSG
                                L10022:
     015526  104423                    TRAP    C$MSG

                                        .SBTTL  RAMEXP  - PRINT RAM EXPD/RECV DATA
                                ;*
                                ;
                                ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA
                                ;       R2      EXPECTED DATA
```

```
                                ;       R4      CONTROLLER RAM ADDRESS
                                ;
                                        BGNMSG  RAMEXP
015530                          RAMEXP::
015530  042701  177400                  BIC     #^C<377>,R1             ;SAVE EXPD RAM DATA BYTE
015534  042702  177400                  BIC     #^C<377>,R2             ;SAVE EXPD RAM DATA BYTE
015540  004737  010066                  JSR     PC,PRIRAM               ;PRINT THE RAM ADDRESS
015544  004737  007742                  JSR     PC,PRIXOR               ;PRINT THE DATA
015550                                  ENDMSG
015550                          L10023:
015550  104423                          TRAP    C$MSG

                                        .SBTTL  TIMEXP  - PRINT TIMER A,B AND EXP/REC
                                ;+
                                ;
                                ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                                ;AND TIMER A,B HEADER MESSAGE
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA
                                ;       R2      EXPECTED DATA
                                ;-

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

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

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

                                        BGNMSG  BADSSR
015700                          BADSSR::
015700  010246                          MOV     R2,-(SP)                ;SAVE DATA TRANSFERRED
015702  042702  177400                  BIC     #177400,R2              ;GET JUST ONE BYTE
```

```
                                        PRINTB  #XFERASC,R2
                                        MOV     R2,-(SP)
                                        MOV     #XFERASC,-(SP)
                                        MOV     #2,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTB
                                        ADD     #6,SP
                                        MOV     (SP)+,R2                ;RESTORE R2
                                        JSR     PC,PRITSSR              ;DECODE TSSR CONTENTS
                                        ENDMSG
                                L10025:
                                        TRAP    C$MSG
                                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
016216                          INTR::
016216  012737  000001  002224          MOV     #1,INTRECV      ;SET FLAG TO SHOW INTERRUPT RECEIVED
016224  105037  016145                  CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
016230  132737  000001  016144          BITB    #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
016236  001003                          BNE     1$              ;BR IF YES
016240  152737  000001  016145          BISB    #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.

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

```
                                        .SBTTL  WAITF   - WAIT FOR SUBSYSTEM READY
                                ;
                                ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
                                ;
                                ;INPUTS:
                                ;
                                ;       R5      ADDRESS OF FIRST DEVICE REGISTER
                                ;
                                ;OUTPUTS:
                                ;
                                ;       R0      CONTENTS OF LAST TSSR READ
                                ;       CARRY   SET - READY BIT SET
                                ;               CLR - TIMEOUT WAITING FOR READY
                                ;
016250  000401          WAITF:: BR      1$              ;NOP WHEN SUPER FIXED
016252                          BREAK                   ; DO A SUPVSR BREAK FIRST.
016252  104422                  TRAP    C$BRK
016254  012746  011000    1$:   MOV     #11000,-(SP)    ;25-APRIL-83 REV B  - 1100 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$:     COM     R3              ;GO' ONE, SET FLAG...
016432  012716  016440                  MOV     #3$,(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 RAISED 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
```

```
                                        BIS     #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
                                        ERRDF   1,NXR,NXRERR    ; NO DEVICE HERE    PRINT IT
016556  104455                          TRAP    C$ERDF
016560  000001                          .WORD   1
016562  003740                          .WORD   NXR
016564  005736                          .WORD   NXRERR
                                        BR      2$
016570  052760  160001  003176  3$:     BIS     #160001,ERTABL(R0)      ; FLAG ERROR IN THE ERROR TABLE
016576                                  ERRDF   2,NOINIT        ; DEVICE NOT IDLE
016576  104455                          TRAP    C$ERDF
016600  000002                          .WORD   2
016602  004335                          .WORD   NOINIT
016604  000000                          .WORD   0
016606  012737  177777  003112  2$:     MOV     #-1,DUFLG       ; DROP THE UNIT
016614                                  DODU    UNITN
016614  013700  002202                  MOV     UNITN,R0
016620  104451                          TRAP    C$DODU
016622                                  DOCLN                   ; ABORT THE PASS
016622  104444                          TRAP    C$DCLN
016624  000423                          BR      5$

016626                          4$:     RFLAGS  R0              ; GET THE OPERATOR FLAGS.
016626  104421                          TRAP    C$RFLA
016630  032700  001000                  BIT     #PNT,R0         ; PRINT THE TEST NUMBERS?
016634  001412                          BEQ     1$              ; BR IF NO
016636  011600                          MOV     (SP),R0         ;GET THE ID MESSAGE
016640                                  PRINTF  #TNAM,R0                ;DISPLAY THE TEST ID
016640  010046                          MOV     R0,-(SP)
016642  012746  016704                  MOV     #TNAM,-(SP)
016646  012746  000002                  MOV     #2,-(SP)
016652  010600                          MOV     SP,R0
016654  104417                          TRAP    C$PNTF
016656  062706  000006                  ADD     #6,SP
016662  005237  002214          1$:     INC     TSTCNT          ; BUMP TEST COUNTER.
016666                                  SETPRI  IPRI            ;PRIORITY THAT OF DEVICE
016666  013700  002212                  MOV     IPRI,R0
016672  10444i                          TRAP    C$SPRI
016674  005726                  5$:     TST     (SP)+           ;FIX UP THE STACK
016676  013705  002206                  MOV     CSRADDR,R5      ; ADDRESS OF TSV REGISTERS ON UNIBUS
016702  000207                          RTS     PC
016704     045     123     045  TNAM:   .ASCIZ  '%S%T%A Test'
                                        .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
   2249 016754  000207                  1$:     RTS     PC
   2250
   2251 016756  000000                  ERRK:   0                       ; LOCAL ERROR COUNT.
   2252 016760     045     101     040  ESUM:   .ASCIZ  /%A %D%A ERRORS/
   2253 016777     105     122     122  EMAXDU: .ASCIZ  /ERROR LIMIT REACHED -- DROPPING UNIT/
   2254                                         .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$ERDF
        017160  000004                  .WORD   4
        017162  016777                  .WORD   EMAXDU
        017164  000000                  .WORD   0
017166                                  DODU    UNITN
        017166  013700  002202          MOV     UNITN,R0
        017172  104451                  TRAP    C$DODU
017174                                  DOCLN
        017174  104444                  TRAP    C$DCLN
017176  012600                  2$:     MOV     (SP)+,R0        ; RESTORE R0
017200  000207                          RTS     PC              ; RETURN TO CALLER
```

```
                                        .SBTTL  CKDROP  - CHECK IF UNIT SHOULD BE DROPPED
                                ;+
                                ; CHECK IF UNIT SHOULD BE DROPPED
                                ;-
017202  010046                  CKDROP: MOV     R0,-(SP)
017204                                  FORCERROR       1$,NOTSSR
017214                                  RFLAGS  R0
017214  104421                          TRAP    C$F LA
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.
```

```
2391 017432  010003                        MOV     R0,R3           ;COPY TEST PATTERN
2392 017434  013701  003124                MOV     FREE,R1         ;GET FIRST FREE LOCATION
2393 017440  013702  003126                MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
2394 017444  010321                 10$:    MOV     R3,(R1)+        ;STORE A BACKGROUND WORD
2395 017446  005302                        DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
2396 017450  003375                        BGT     10$             ;BR IF NO
2397 017452  005737  003132                TST     KTFLG           ; GOT KT?
2398 017456  001477                        BEQ     55$             ; NO. GET OUT.
2399 017460  004737  017256                JSR     PC,KTON         ; YES. ENABLE KT.
2400 017464  005000                        CLR     R0              ;HIGH ORDER ADDRESS START
2401 017466  013701  003152                MOV     PST32W,R1       ;GET >28K START ADDRESS (IN 32W BLOCKS)
2402         000006                        .REPT   6
2403                                       CLC                     ;CLEAR C BIT
2404                                       ROL     R1              ;CONVERT BLOCKS TO WORDS
2405                                       ROL     R0              ;MAKE IT DOUBLE PRECISION
2406                                       .ENDR
2407 017536  004737  017316                JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
2408 017542  010320                 30$:    MOV     R3,(R0)+        ;STORE TEST PATTERN IN >28K ADDRESS
2409 017544  020027  160000                CMP     R0,#160000      ;END OF PAR6 MAPPING AREA?
2410 017550  103774                        BLO     30$             ;BR IF NO
2411 017552  162700  020000                SUB     #20000,R0       ;BACKUP INTO PAR6 MAPPING BEGIN
2412 017556  062737  000200  172354        ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
2413 017564  023737  172354  003132        CMP     @#KIPAR6,KTFLG  ;END OF MEMORY?
2414 017572  001427                        BEQ     50$             ;BR IF YES
2415 017574  005737  003144                TST     T23A            ;11/23A?
2416 017600  001407                        BEQ     35$             ;NO KEEP GOING
2417 017602  013704  177572                MOV     SR0,R4          ;GET SR0 CONTENTS
2418 017606  042704  177761                BIC     #177761,R4      ;CLEAR ALL BUT PAGE NUMBER
2419 017612  022704  000016                CMP     #16,R4          ;SEE IF PAGE 7
2420 017616  001415                        BEQ     50$             ;EXIT IF THERE
2421 017620  005737  003146          35$:  TST     T23B            ;11/23B?
2422 017624  001410                        BEQ     45$             ;NO KEEP GOING
2423 017626  023727  172354  007600        CMP     @#KIPAR6,#7600  ;REACHED 18 BITS?
2424 017634  103001                        BHIS    40$             ;YES
2425 017636  000403                        BR      45$             ;NO KEEP GOING
2426 017640  012737  000020  172516  40$:  MOV     #20,SR3         ;SET 22 BIT RELOCATION
2427 017646  000137  017542          45$:  JMP     30$             ;KEEP GOING ON ETC.
2428 017652  004737  017274          50$:  JSR     PC,KTOFF        ; DISABLE KT.
2429 017656  000207                55$:  RTS     PC
```

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

```
2488 020040  010137  002240                    MOV     R1,ERRLO        ;SAVE LOW ORDER IN ERROR
2489 020044  010337  002232                    MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
2490 020050  011437  002234                    MOV     (R4),RECV       ;SAVE RECV FOR ERROR REPORT
2491 020054  000421                            BR      50$             ;
2492 020056  062701  000002            32$:    ADD     #2,R1           ;UPDATE NON PAR6 ADDRESS
2493 020062  005500                            ADC     R0              ;MAKE IT DOUBLE PRECISION ADD
2494 020064  062704  000002                    ADD     #2,R4           ;UPDATE PAR FORMAT ADDRESS
2495 020070  020427  160000                    CMP     R4,#160000      ;END OF PAR6 MAPPING AREA?
2496 020074  103755                            BLO     30$             ;BR IF NO
2497 020076  162704  020000                    SUB     #20000,R4       ;BACKUP INTO PAR6 MAPPING BEGIN
2498 020102  062737  000200  172354            ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
2499 020110  023737  172354  003132            CMP     @#KIPAR6,KTFLG  ;END OF MEMORY?
2500 020116  101744                            BLOS    30$             ;BR IF NO
2501 020120  004737  017274            50$:    JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
2502 020124  000241                            CLC                     ;SET FAILURE
2503 020126  000403                            BR      60$             ;
2504 020130  004737  017274            55$:    JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
2505 020134  000261                            SEC                     ;SET SUCCESS
2506 020136  000207                    60$:    RTS     PC
2507                                           .SBTTL  REGSAV  - SAVE R1-R5 ON STACK
2508                                   ;+
2509                                   ;
2510                                   ;ROUTINE TO
2511                                   ;SAVE R1 THROUGH R5 ON THE STACK
2512                                   ;
2513                                   ;CALLING SEQUENCE:
2514                                   ;
2515                                   ;       JSR     R5,REGSAV
2516                                   ;
2517                                   ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
2518                                   ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
2519                                   ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
2520                                   ;REGISTERS.
2521                                   ;
2522                                   ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
2523                                   ;CALLED VIA A JSR PC INSTRUCTION
2524                                   ;
2525                                   ;-
2526
2527 020140                            REGSAV:
2528 020140  010446                            MOV     R4,-(SP)
2529 020142  010346                            MOV     R3,-(SP)
2530 020144  010246                            MOV     R2,-(SP)
2531 020146  010146                            MOV     R1,-(SP)
2532 020150  010546                            MOV     R5,-(SP)
2533 020152  016605  000012                    MOV     10.(SP),R5
2534 020156  004736                            JSR     PC,@(SP)+
2535 020160  012601                            MOV     (SP)+,R1
2536 020162  012602                            MOV     (SP)+,R2
2537 020164  012603                            MOV     (SP)+,R3
2538 020166  012604                            MOV     (SP)+,R4
2539 020170  012605                            MOV     (SP)+,R5
2540 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
                                ;
020174                          GETPAT::
020174                                  SAVREG                  ;SAVE THE GENERAL REGISTERS
020200                          1$:     GMANID  DATASC,PATDAT,0,377,0,377,NO
020200  104443                          TRAP    C$GMAN
020202  000406                          BR      10000$
020204  020230                          .WORD   PATDAT
020206  000022                          .WORD   T$CODE
020210  020232                          .WORD   DATASC
020212  000377                          .WORD   377
020214  000000                          .WORD   T$LOLIM
020216  000377                          .WORD   T$HILIM
020220                          10000$:
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
                                ;-
020256                          GETSEL::
                                        SAVREG                          ;SAVE GENERAL REGISTERS
020256  010002                         MOV     R0,R2                   ;SAVE THE MENU ADDRESS
020262  010203                 1$:     MOV     R2,R3                   ;START OF MENU STRING
020264  005713                 2$:     TST     (R3)                    ;END OF ASCII ?
020266  001412                         BEQ     3$                      ;BRANCH IF ALL LINES DISPLAYED
020270                                  PRINTF  @SELASC,(R3)+           ;DISPLAY THE MENU
020272  012346                         MOV     (R3)+,-(SP)
020274  012746  020442                 MOV     @SELASC,-(SP)
020300  012746  000002                 MOV     #2,-(SP)
020304  010600                         MOV     SP,R0
020306  104417                         TRAP    C$PNTF
020310  062706  000006                 ADD     #6,SP
020314  000764                         BR      2$
020316                          3$:     GMANID  MENASC,MENRES,D,-1,0,-1,NO
020316  104443                         TRAP    C$GMAN
020320  000406                         BR      10001$
020322  020476                         .WORD   MENRES
020324  000042                         .WORD   T$CODE
020326  020447                         .WORD   MENASC
020330  177777                         .WORD   -1
020332  000000                         .WORD   T$LOLIM
020334  177777                         .WORD   T$HILIM
020336                          10001$:
020336                                  BNCOMPLETE      1$      ;RETRY IF ERROR
020336  103352                         BCC     1$
020340  013700  020476                 MOV     MENRES,R0               ;GET THE OPERATOR'S REPLY
020344  020001                         CMP     R0,R1                   ;COMPARE TO MAXIMUM ALLOWED
020346  101411                         BLOS    5$                      ;BRANCH IF OK
020350                                  PRINTF  @MENERR                 ;DISPLAY ERROR MESSAGE
020350  012746  020374                 MOV     @MENERR,-(SP)
020354  012746  000001                 MOV     #1,-(SP)
020360  010600                         MOV     SP,R0
020362  104417                         TRAP    C$PNTF
020364  062706  000004                 ADD     #4,SP
020370  000735                         BR      1$                      ;RETRY
020372  000207                 5$:     RTS     PC                      ;RETURN TO CALLER
020374     045     116     045 MENERR: .ASCIZ   '<N><A *** Menu Selection Too Large ***'
020442     045     116     045 SELASC: .ASCIZ   '<N><T>'
020447     105     156     164 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::
                                        SAVREG                  ;SAVE THE REGISTERS
020500                                  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  000423                          BR      40$             ;RETURN WITH FLAGS CLEAR
020740  013701  177520          15$:    MOV     BDVPCR,R1       ;SAVE PCR CONTENTS
020744  062701  000001                  ADD     #1,R1           ;ADD ONE TO IT
020750  012702  177520                  MOV     #BDVPCR,R2      ;GET BDV11 PCR ADDRESS
020754  005212                          INC     (R2)            ;TRY TO WRITE TO IT
020756  013703  177520                  MOV     BDVPCR,R3       ;GET RESULTS
020762  020103                          CMP     R1,R3           ;DID IT CHANGE?
020764  001006                          BNE     20$             ;NO, MUST BE 11/23B
020766  005237  003144                  INC     T23A            ;SET THE FLAG
020772  042737  170000  002120          BIC     #170000,L$HIME  ;SUPERVISOR COULD BE WRONG
                                ;       NOP                     ;BR 40$ FOR RELEASE
                                ;       PRINTF  #M8186          ;TELL THE SYSTEM TYPE
021000  000402                          BR      40$             ;RETURN
021002  005237  003146          20$:    INC     T23B            ;SET THE FLAG
                                ;       NOP                     ;BR 40$ FOR RELEASE
                                ;       PRINTF  #M8189          ;TELL THE SYSTEM TYPE
021006                          40$:
021006  000207                          RTS     PC              ;RETURN
```

```
                                        .SBTTL  KTINIT    SETUP KT11 MEMORY MANAGEMENT REGISTERS
                                ;+
                                ;
                                ;ROUTINE TO INIT KT 11
                                ;
                                ;
                                KTINIT:
021010  005037  003132                CLR     KTFLG           ; INIT >28K MEMORY FLAG
021014  005037  003134                CLR     KTENABLE        ; INIT TEST >28K FLAG
021020  023727  002120  001577        CMP     L$MEM,#1577     ; GOT ENOUGH MEMORY (>28K)?
021026  101444                        BLOS    9$              ; NO.
021030  013700  000004                MOV     @#ERRVEC,R0     ; SAVE OLD ERR VEC PTR.
021034  012737  021126  000004        MOV     #2$,@#ERRVEC    ; SET ERR VEC PTR.
021042  005737  177572                TST     @#SR0           ; GOT KT11?
021046  000240                        NOP                     ;  (TRAP IF NO).
021050  013737  002120  003132        MOV     L$MEM,KTFLG     ; YES. SET KT FLAG.
021056  042737  000177  003132        BIC     #177,KTFLG      ;
021064  010037  000004                MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
021070  005000                        CLR     R0              ; R0 = AR DATA.
021072  012701  172340                MOV     #KIPAR0,R1      ; R1 = KI REGS PTR.
021076  012761  077406  177740  1$:   MOV     #77406,-40(R1)  ; SET DESCRIPTOR REG.
021104  010021                        MOV     R0,(R1)+        ; SET KIPAR REG.
021106  062700  000200                ADD     #200,R0         ; BUMP AR DATA BY "4K".
021112  020027  002000                CMP     R0,#2000        ; AT "I/O"?
021116  001367                        BNE     1$              ; NO.
021120  012741  177600                MOV     #177600,-(R1)   ; YES. SET KTPAR7 FOR I/O.
021124  000405                        BR      9$
                                ;
021126  012716  021134        2$:     MOV     #6$,(SP)        ; SET UP RETURN
021132  000002                        RTI                     ; RTI TO NEXT LOCATION
                                ;
021134  010037  000004        6$:     MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
                                ;
021140  000207                9$:     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.
                                        ;
                                        ;

                                        INVERT::

021142  005737  002226                          TST     EXTFEA                  ; IS SWITCH SET?
021146  001020                                  BNE     1$                      ; YES,EXIT STAGE RIGHT!(or the next one outa town!)
021150  012737  100206  021214                  MOV     #100206,CMDPKT          ; WRT SUB-SYS MEM CMD
021156  012737  021224  021216                  MOV     #WSMBK,CMDPKT+2         ; MSG BUF ADDR
021164  012737  000006  021222                  MOV     #6,CMDPKT+6             ; BYTE COUNT
021172  012737  100010  021224                  MOV     #100010,WSMBK           ; INVERT THE SWITCH
021200  012704  021214                          MOV     #CMDPKT,R4              ; SET CMDPKT INTO R4
021204  004737  010662                          JSR     PC,WRTCHR               ; DO IT
021210  000207                        1$:       RTS     PC                      ; RETURN

                                        ;       COMMAND PACKET.

                021214                  .       .       <.+3>&177774    ;MUST BE ON MOD 4 BOUNDRY.

021214  000000                        CMDPKT:: 0                              ;1ST WORD IS TSO5 COMMAND.
021216  000000                                  0                               ;2ND WORD IS THE BUFFER LOW ADDRESS.
021220  000000                                  0                               ;3RD WORD IS THE BUFFER HIGH ADDRESS.
021222  000000                                  0                               ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.

                                        ;       WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.

021224  000000                        WSMBK:: 0                               ;1ST WORD:: SEL 0
021226  000000                                  0                               ;2ND WORD:: SEL 2
021230  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.
                                        ;-

021232                                  MEMCK::

021232                                          SAVREG                          ;SAVE THE REGISTERS
021236  005037  005136                          CLR     NXMFLG                  ;CLEAR THE FLAG
021242  005037  003140                          CLR     NXMLO                   ;CLEAR THE TEST ADDRESS LO
021246  005037  003142                          CLR     NXMHI                   ;CLEAR THE TEST ADDRESS HI
021252  005737  003146                          TST     T238                    ;IS IT A 11/23B?
```

```
2770 021256  001407                          BEQ     1$                      ;NO
2771 021260  023727  002120  007777          CMP     L$HIME,#7777            ; GREATER THAN 128K
2772 021266  103406                          BLO     2$                      ; NO
2773 021270  004737  021406                  JSR     PC,NXMTST               ;SETUP THE ADDRESS
2774 021274  ^00427                          BR      13$                     ;SET THE FLAG AND EXIT
2775 021276  005737  003144          1$:     TST     T23A                    ;IS IT A 11/23A?
2776 021302  001413                          BEQ     4$                      ;NO
2777 021304  023727  002120  005777  2$:     CMP     L$HIME,#5777            ;GREATER THAN 96K
2778 021312  101023                          BHI     14$                     ;YES,23A/23B WITH 128K MEMORY
2779 021314  023727  002120  003777          CMP     L$HIME,#3777            ;GREATER THAN 64K BUT LESS THAN 92K?
2780 021322  103403                          BLO     4$                      ;NO, CHECK 24K
2781 021324  004737  021406                  JSR     PC,NXMTST               ;SETUP THE ADDRESS
2782 021330  000411                          BR      13$                     ;SET THE FLAG AND EXIT
2783 021332  023727  002120  001577  4$:     CMP     L$HIME,#1577            ;GREATER THAN 24K BUT LESS THAN 64K?
2784 021340  103410                          BLO     14$                     ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
2785 021342  004737  021406                  JSR     PC,NXMTST               ;SETUP THE ADDRESS
2786 021346  062737  000077  003142          ADD     #77,NXMHI               ;FOOL THE 11/02 & 11/03
2787 021354  005237  003136          13$:    INC     NXMFLG                  ;SET THE FLAG
2788 021360  000411                          BR      15$                     ;EXIT
2789 021362  000410                  14$:    BR      15$                     ;NOP FOR PRINTOUT
2790 021364                                  PRINTF  #NOMEM                  ;TELL THEM & EXIT ***NO PRINT*****
     021364  012746  005460                  MOV     #NOMEM,-(SP)
     021370  012746  000001                  MOV     #1,-(SP)
     021374  010600                          MOV     SP,R0
     021376  104417                          TRAP    C$PNTF
     021400  062706  000004                  ADD     #4,SP
2791 021404  000207                  15$:    RTS     PC                      ;RETURN
2792
2793                                         ;*
2794                                         ;       SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
2795                                         ;
2796                                         ;OUTPUTS:NXMLO,NXMHI            ;SETUP WITH NXM ADDRESS
2797                                         ;
2798                                         ;-
2799
2800 021406  013701  002120          NXMTST: MOV     L$HIME,R1               ;GET TOP OF MEMORY
2801 021412  062701  000200                  ADD     #200,R1                 ;MAKE IT I/O BLOCK OR OTHER NXM
2802 021416  042701  000177                  BIC     #177,R1
2803 021422  010102                          MOV     R1,R2                   ;RESAVE RESULTS
2804         000006                          .REPT   6
2805                                         ASL     R1                      ;PUT IN PLACE FOR XFER
2806                                         .ENDR
2807 021440  010137  003140                  MOV     R1,NXMLO                ;SAVE TEST ADDRESS LOW
2808         000012                          .REPT   10.
2809                                         ASR     R2                      ;PUT IN PLACE FOR XFER
2810                                         .ENDR
2811 021470  042702  177700                  BIC     #177700,R2              ;DON'T WANT ILA!
2812 021474  010237  003142                  MOV     R2,NXMHI                ;SAVE TEST ADDRESS HIGH
2813 021500  000207                          RTS     PC                      ;RETURN
2814
2815 021502                                  ENDMOD
```

```
TSV4 - MISCELLANEOUS SECTIONS   MACRO M1113  14-JUN-84 15:55                                                    SEQ 0095
KTINIT  - SETUP KT11 MEMORY MANAGEMENT REGISTERS

                                                .TITLE  TSV4   MISCELLANEOUS SECTIONS

 9 021502                                       BGNMOD  TSV4
   021502                               TSV4::
20 021502                                       .SBTTL  PROTECTION TABLE
   021502                                       BGNPROT
   021502                               L$PROT::
21 021502  17 777  177777  177777               .WORD   -1, -1, -1,  1          ;NO DEVICE PROTECTION REQUIRED.
22 021512                                       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
                                ;
021512                                  BGNINIT
021512                          L$INIT::
021512  005037  002226          40$:    CLR     EXTFEA
021516  005037  003136                  CLR     NXMFLG
021522  012737  006360  002200          MOV     #EPRT1,EPRTSW           ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
021530  005037  003154                  CLR     SIFLAG                  ;CLEAR "SOFT INIT" FLAG
021534  005037  003134                  CLR     KTENABLE                ;CLEAR TEST ABOVE 28K FLAG
021540  005037  002302                  CLR     RAMSIZ                  ;CLEAR RAM SIZE FOR RAMERR ROUTINE
021544                                  READEF  #EF.CONTINUE
021544  012700  000036                  MOV     #EF.CONTINUE,R0
021550  104447                          TRAP    C$REFG
021552                                  BNCOMPLETE 1$
021552  103023                          BCC     1$
021554  023737  002202  002012          CMP     UNITN,L$UNIT            ;UNIT IN RANGE?
021562  103070                          BHIS    4$                      ;BR IF NO.
021564  005737  003112                  TST     DUFLG                   ;DROPPED UNIT?
021570  100472                          BMI     NXTU                    ;BR IF YES
021572  013701  002202                  MOV     UNITN,R1
021576  006301                          ASL     R1
021600  005761  003176                  TST     ERTABL(R1)
021604  001516                          BEQ     SETU
021606  032761  040000  003176          BIT     #BIT14,ERTABL(R1)       ;DROPPED?
021614  001060                          BNE     NXTU
021616                                  EXIT    INIT                    ;DO NOTHING IF "CONTINUE".
021616  104432                          TRAP    C$EXIT
021620  000416                          .WORD   L10030-.
021622                          1$:     READEF  #EF.NEW
021622  012700  000035                  MOV     #EF.NEW,R0
021626  104447                          TRAP    C$REFG
021630                                  BNCOMPLETE NXTU                 ;TAKE NEXT UNIT IF NOT NEW PASS.
021630  103052                          BCC     NXTU
021632                                  READEF  #EF.START
021632  012700  000040                  MOV     #EF.START,R0
021636  104447                          TRAP    C$REFG
021640                                  BCOMPLETE 2$
021640  103404                          BCS     2$
021642                                  READEF  #EF.RESTART
021642  012700  000037                  MOV     #EF.RESTART,R0
021646  104447                          TRAP    C$REFG
021650                                  BNCOMPLETE 31$
021650  103031                          BCC     31$
021652                          2$:                                     ;1ST PASS, BUS-INIT...
021652                                  BRESET                          ;BUS RESET.
021652  104433                          TRAP    C$RESET
```

INITIALIZE SECTION

```
 65 021654  005037  002214                    CLR     TSTCNT                  ;NUMBER OF TESTS RUN IN PASS
 66 021660  005037  002222                    CLR     FATFLG                  ;CLEAR FATAL ERROR COUNT
 67 021664  005037  003144                    CLR     T23A                    ;CLEAR 11/23A FLAG
 68 021670  005037  003146                    CLR     T23B                    ;CLEAR 11/23B FLAG
 69                                     ;      MOV     #340,-(SP)
 70                                     ;      MOV     #20$,-(SP)              ;RETURN TO DEBUGGER
 71                                     ;      JMP     O.ODT           ;;@ENTER THE DEBUGGER
 72 021674  005037  003400                    CLR     SKIPT           ;CLEAR THE SUBTEST "SKIPPER"
 73 021700                              20$:
 74 021700  012737  177777  002204            MOV     #-1,QVP                 ;...QUICK VERIFY...
 75 021706  004737  020630                    JSR     PC,ENVIRN               ;SET ENVIRONMENT.
 76 021712  004737  021010                    JSR     PC,KTINIT               ;INITIALIZE KT MEMORY MANAGEMENT
 77 021716  012700  003176                    MOV     #ERTABL,R0
 78 021722  005020                     30$:   CLR     (R0)+                   ;CLEAR THE ERROR TABLE
 79 021724  020027  003376                    CMP     R0,#ERTABE
 80 021730  103774                           BLO     30$
 81 021732  000404                           BR      4$
 82 021734  005037  002204             31$:   CLR     QVP
 83 021740  000137  022010                    JMP     PASRPT                  ;GO REPORT THE STATUS
 84
 85 021744                              4$:
 86 021744  012737  177777  002202    NEWPAS: MOV     #-1,UNITN               ;INIT UNIT NUMBER...
 87 021752  005037  002220                    CLR     DEVCNT                  ;CLEAR COUNT OF DEVICES RUNNING
 88 021756                              NXTU:  BREAK
    021756  104422                           TRAP    C$BRK
 89 021760  005237  002202                    INC     UNITN                   ;...AND SET NEXT UNIT NUMBER.
 90 021764  023737  002202  002012            CMP     UNITN,L$UNIT
 91 021772  103423                           BLO     SETU
 92 021774  012737  177777  003112            MOV     #-1,DUFLG
 93 022002  000401                           BR      11$
 94 022004                                  DOCLN                           ;ABORT, NO MORE UNITS.
    022004  104444                           TRAP    C$DCLN
 95 022006  000240                     11$:   NOP
 96 022010                              PASRPT:
 97 022010  023727  002012  000001            CMP     L$UNIT,#1               ;HOW MANY UNITS SELECTED?
 98 022016  101752                           BLOS    NEWPAS                  ;BR IF ONLY 1
 99 022020  005737  002220                    TST     DEVCNT                  ;ARE ANY STILL RUNNING?
100 022024  001747                           BEQ     NEWPAS                  ;BR IF NO
101 022026                                  RFLAGS  RC
    022026  104421                           TRAP    C$RFLA
102 022030  032700  000100                    BIT     #ISR,R0                 ;SHOULD WE PRINT STATISTICS
103 022034  001343                           BNE     NEWPAS                  ;BR IF NO
104
105 022036                                  DORPT
    022036  104424                           TRAP    C$DRPT
106 022040  000741                           BR      NEWPAS
107 022042                             10$:
108
109 022042                             SETU:  GPHARD  UNITN,R0                ;GET UNIT N P-TABLE POINTER.
    022042  013700  002202                    MOV     UNITN,R0
    022046  104442                           TRAP    C$GPHRD
110 022050                                  BNCOMPLETE NXTU                 ;BR IF UNIT NOT AVAILABLE.
    022050  103342                           BCC     NXTU
111 022052  005037  003112                    CLR     DUFLG                   ;CLEAR "DROPPED" FLAG.
112 022056  005237  002220                    INC     DEVCNT
113 022062  012001                           MOV     (R0)+,R1                ;GET 1ST REGISTER ADDRESS.
114 022064  010137  002206                    MOV     R1,CSRADDR              ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
```

```
    022070  012001                           MOV     (R0)+,R1                  ;GET VECTOR ADDRESS.
                                            ;MOV     (R0),R2                   ;GET INTERRUPT PRIORITY
                                            ;MOV     R2,IPRI                   ;SET INTERRUPT PRIORITY.
    022072  010137  002210                   MOV     R1,IVEC                   ;SET INTERRUPT VECTOR POINTER...
    022076  012721  016216                   MOV     #INTR,(R1)+               ;...VECTOR...
    022102  013721  002212                   MOV     IPRI,(R1)+                ;...AND PRIORITY.

    022106                           1$:
                                     ;       TST     QVP             ;1ST PASS ??
                                     ;       BEQ     5$              ;NO, SKIP THE PASS 1 STUFF.


                                     ;
                                     ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
                                     ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
                                     ;
    022106  013701  002202                   MOV     UNITN,R1
    022112  006301                           ASL     R1
    022114  052761  100000  003176           BIS     #BIT15,ERTABL(R1)         ;SAY DEVICE RUNNING
    022122  005037  005772                   CLR     EXTA                      ;CLEAR ERROR EXTENSION FLAG.
    022126  023727  002012  000001           CMP     L$UNIT,#1                 ;ARE WE TESTING MULTIPLE UNITS?
    022134  101416                           BLOS    10$                       ;BR IF NO.
    022136                                   RFLAGS  R0                        ;YES -- GET OPERATOR FLAGS.
    022136  104421                           TRAP    C$RFLA
    022140  032700  001000                   BIT     #PNT,R0                   ;SHOULD WE PRINT UNIT #?
    022144  001412                           BEQ     10$                       ;BR IF NOT.
    022146                                   PRINTF  #PUNIT,UNITN              ;PRINT THE UNIT #
    022146  013746  002202                   MOV     UNITN,-(SP)
    022152  012746  022240                   MOV     #PUNIT,-(SP)
    022156  012746  000002                   MOV     #2,-(SP)
    022162  010600                           MOV     SP,R0
    022164  104417                           TRAP    C$PNTF
    022166  062706  000006                   ADD     #6,SP
    022172                           10$:
    022172  005037  003114                   CLR     NODEV
    022176  013701  002206                   MOV     CSRADDR,R1                ;ADDRESS OF FIRST REGISTER
    022202  010102                           MOV     R1,R2                     ;START OF REGISTERS
    022204  062702  000002                   ADD     #TSSR,R2                  ;ADDRESS OF TSSR REGISTER
    022210  004737  016376                   JSR     PC,XNXM                   ;TEST BOTH CONTROLLER REGISTERS...
    022214  103005                           BCC     2$                        ;...AND BR IF ALL OK.
    022216  010137  003114                   MOV     R1,NODEV                  ;FLAG DEVICE AS NON-EXISTENT
    022222  012737  177777  003112           MOV     #-1,DUFLG                 ;DROP THIS UNIT.
    022230                           2$:
                                     ;
                                     ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
                                     ;
    022230                           5$:     SETPRI  #PRI00          ;ENABLE INTERRUPTS.
    022230  012700  000000                   MOV     #PRI00,R0
    022234  104441                           TRAP    C$SPRI
    022236                                   ENDINIT
    022236                           L10030:
    022236  104411                           TRAP    C$INIT

    022240     045     116     045   PUNIT:  .ASCIZ  /%N%N%A***** TESTING UNIT %D2%A *****/
                                             .EVEN
```

```
                                        .SBTTL  ADD AND DROP UNITS SECTIONS

                                ;++
                                ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
                                ;    OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
                                ;-
                                        BGNAU
022306                          L$AU::
022306  010001                          MOV     R0,R1                   ; GET UNIT TO BE ADDED (R0)
022310  006301                          ASL     R1                      ; MAKE IT A WORD INDEX
022312  052761  100000  003176          BIS     #100000,ERTABL(R1)      ; SET THE "ACTIVE" BIT
022320  042761  040000  003176          BIC     #40000,ERTABL(R1)       ; CLEAR THE "DROPPED" BIT
022326                                  PRINTF  #1$,R0
022326  010046                          MOV     R0,-(SP)
022330  012746  022354                  MOV     #1$,-(SP)
022334  012746  000002                  MOV     #2,-(SP)
022340  010600                          MOV     SP,R0
022342  104417                          TRAP    C$PNTF
022344  062706  000006                  ADD     #6,SP
022350                                  EXIT    AU
022350  000167                          .WORD   J$JMP
022352  000026                          .WORD   L10031-2-.
022354     045     116     045  1$:     .ASCIZ  /%N%A  UNIT %D%A ADDED/
                                        .EVEN

                                        ENDAU           ; UNUSED.
022402                          L10031:
022402  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.

                                        BGNDU
022404                          L$DU::
022404  012737  177777  003112          MOV     #-1,DUFLG
022412  010001                          MOV     R0,R1
022414  006301                          ASL     R1
022416  052761  140000  003176          BIS     #140000,ERTABL(R1)      ; SAY DROPPED
022424  000240  000240  000240          240,240,240                     ; ?????????
022432                                  PRINTF  #1$,R0
022432  010046                          MOV     R0,-(SP)
022434  012746  022460                  MOV     #1$,-(SP)
022440  012746  000002                  MOV     #2,-(SP)
022444  010600                          MOV     SP,R0
022446  104417                          TRAP    C$PNTF
022450  062706  000006                  ADD     #6,SP
022454                                  EXIT    DU
022454  000167                          .WORD   J$JMP
022456  000030                          .WORD   L10032-2 .
```

```
    197 022460     045     116     045  1$:     .ASCIZ  /%N%A  UNIT %D%A DROPPED/
    198                                         .EVEN
    199 022510                                  ENDDU
        022510                          L10032:
        022510  104453                         TRAP    C$DU
    200                                 ;**
    201                                 ; AUTO-DROP CODE SECTION.
    202                                 ;
    203 022512                                  BGNAUTO
        022512                          L$AUTO::
    204 022512  013705  002206                  MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
    205 022516  012703  000550                  MOV     #360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
    206 022522  004737  016250          10$:    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
    207 022526  103420                          BCS     20$                     ;LEAVE WHEN SSR IS SET
    208 022530                                  DELAY   250.                    ;WAIT FOR .25 SECONDS
        022530  012727  000372                  MOV     #250.,(PC)+
        022534  000000                          .WORD   0
        022536  013727  002116                  MOV     L$DLY,(PC)+
        022542  000000                          .WORD   0
        022544  005367  177772                  DEC     -6(PC)
        022550  001375                          BNE     .-4
        022552  005367  177756                  DEC     -22(PC)
        022556  001367                          BNE     .-20
    209 022560  005303                          DEC     R3                      ;BUMP COUNTER DOWN
    210 022562  001357                          BNE     10$                     ;KEEP GOING
    211 022564  004737  017202                  JSR     PC,CKDROP               ;TRY AND DROP UNIT
    212 022570                          20$:
    213 022570                                  ENDAUTO                 ; UNUSED.
        022570                          L10033:
        022570  104461                         TRAP    C$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.
                                ;
022572                                  BGNCLN
022572                          L$CLEAN::
022572  013705  002206                   MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
022576  005737  003112                   TST     DUFLG           ;"DROPPED" FLAG IS SET ON...
022602  100405                           BMI     1$              ;...AND GROSS CONTROLLER FAULT...
                                                                 ;...DON'T TRY TO XCT CLEANUP CODE.

022604  012765  000000  000002          MOV     #0,TSSR(R5)     ;DO SOFT INIT
022612  004737  016250                   JSR     PC,WAITF
022616                          1$:
022616                          2$:     ENDCLN
022616                          L10034:
022616  104412                          TRAP    C$CLEAN
                                ;++
                                ; THE REPORT CODING SECTION CONTAINS THE
                                ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;-
022620                                  BGNRPT
022620                          L$RPT::
022620                                  PRINTS  #DEVSUM
022620  012746  023062                   MOV     #DEVSUM,-(SP)
022624  012746  000001                   MOV     #1,-(SP)
022630  010600                           MOV     SP,R0
022632  104416                           TRAP    C$PNTS
022634  062706  000004                   ADD     #4,SP
022640  010246                           MOV     R2,-(SP)
022642  010346                           MOV     R3,-(SP)
022644  010446                           MOV     R4,-(SP)
022646  012704  003176                   MOV     #ERTABL,R4      ; GET START OF ERROR TABLE.
022652  005003                           CLR     R3              ; CLEAR UNIT NUMBER
022654  011402                  1$:      MOV     (R4),R2         ; GET ERROR TABLE ENTRY & TEST IT.
022656  001467                           BEQ     4$              ; ZERO IF UNIT NOT RUN
022660  100066                           BPL     4$
022662  032702  040000                   BIT     #BIT14,R2       ; WAS UNIT DROPPED?
022666  001015                           BNE     2$              ; BR IF YES
022670  042702  170000                   BIC     #^C7777,R2      ; GET ERROR COUNT FIELD
022674                                   PRINTS  #DEVONL,R3,R2   ; PRINT
022674  010246                           MOV     R2,-(SP)
022676  010346                           MOV     R3,-(SP)
022700  012746  023117                   MOV     #DEVONL,-(SP)
022704  012746  000003                   MOV     #3,-(SP)
022710  010600                           MOV     SP,R0
022712  104416                           TRAP    C$PNTS
022714  062706  000010                   ADD     #10,SP
022720  000446                           BR      4$
022722  020227  160000          2$:      CMP     R2,#160000      ; WAS UNIT NON-EXISTENT?
022726  001012                           BNE     3$              ; BR IF NO
022730                                   PRINTS  #DEVNXR,R3
022730  010346                           MOV     R3,-(SP)
022732  012746  023167                   MOV     #DEVNXR,-(SP)
```

```
                022736  012746  000002          MOV     #2,-(SP)
                022742  010600                  MOV     SP,R0
                022744  104416                  TRAP    C$PNTS
                022746  062706  000006          ADD     #6,SP
    254         022752  000431                  BR      4$
    255         022754  020227  160001  3$:     CMP     R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
    256         022760  001012                  BNE     30$             ; BR IF NO.
    257         022762                          PRINTS  #DEVNRD,R3
                022762  010346                  MOV     R3,-(SP)
                022764  012746  023251          MOV     #DEVNRD,-(SP)
                022770  012746  000002          MOV     #2,-(SP)
                022774  010600                  MOV     SP,R0
                022776  104416                  TRAP    C$PNTS
                023000  062706  000006          ADD     #6,SP
    258         023004  000414                  BR      4$
    259         023006  042702  170000  30$:    BIC     #^C7777,R2
    260         023012                          PRINTS  #DEVDRO,R3,R2
                023012  010246                  MOV     R2,-(SP)
                023014  010346                  MOV     R3,-(SP)
                023016  012746  023332          MOV     #DEVDRO,-(SP)
                023022  012746  000003          MOV     #3,-(SP)
                023026  010600                  MOV     SP,R0
                023030  104416                  TRAP    C$PNTS
                023032  062706  000010          ADD     #10,SP
    261         023036  062704  000002  4$:     ADD     #2,R4
    262         023042  005203                  INC     R3
    263         023044  020427  003376          CMP     R4,#ERTABE
    264         023050  103701                  BLO     1$
    265         023052  012604                  MOV     (SP)+,R4
    266         023054  012603                  MOV     (SP)+,R3
    267         023056  012602                  MOV     (SP)+,R2
    268         023060                          ENDRPT          ; UNUSED.
                023060                  L10035:
                023060  104425                  TRAP    C$RPT
    269
    270         023062     045     116     045  DEVSUM: .ASCIZ  /<FF><CR><LF>DEVICE STATUS SUMMARY:<LF>/
    271         023117     045     101     040  DEVONL: .ASCIZ  /<CR>  UNIT <D3><CR>   ONLINE,  ERRORS = <D><LF>/
    272         023167     045     101     040  DEVNXR: .ASCIZ  /<CR>  UNIT <D3><CR>   DROPPED, NON-EXISTENT REGISTER<LF>/
    273         023251     045     101     040  DEVNRD: .ASCIZ  /<CR>  UNIT <D3><CR>   DROPPED, NOT READY AT STARTUP<LF>/
    274         023332     045     101     040  DEVDRO: .ASCIZ  /<CR>  UNIT <D3><CR>   DROPPED, ERRORS = <D><LF>/
    275                                             .EVEN
    276
    277         023402                          ENDMOD
    278
```

```
TSV5   HARDWARE TESTS   MACRO M1113  14 JUN-84 15:55                                  SEQ 0103
CLEAN-UP AND REPORT CODING SECTIONS

      1                                       .TITLE  TSV5 - HARDWARE TESTS
      2
      9
     10 023402                                BGNMOD  TSV5
        023402                         TSV5::
     16
     24
```

```
                                        .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
                                ; -


023402                                  BGNTST
023402                                                                          T1::
023402  012700  024042                  MOV     #TST13ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
023406  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
023412  012737  000012  002216          MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
023420                          T13LOOP:
023420  004737  024316                  JSR     PC,T13REST              ;SET PACKET TO START-UP VALUES

023424  012703  002764                  MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
023430  012704  024000                  MOV     #T13PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
023434  012764  000010  000006          MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
023442                          5$:
023442  004737  015774                  JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
023446  103405                          BCS     10$                     ;BR IF SOFT INIT OKAY
023450  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
023452                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
023452  104455                                                                  TRAP    C$ERDF
023454  000144                                                                  .WORD   100
023456  003652                                                                  .WORD   SFIERR
023460  012034                                                                  .WORD   SFIMSG

                                ;Do WRITE CHARACTERISTICS command.
023462  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
023466  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
023472  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
023476                                  FORCERROR       12$                     ;@@FORCE ERROR IF FORCER=1
023512  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
023514  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
023516                                  NEXT.ERRNO
023516                          12$:    ERRDF   ERRNO,T13SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
023516  104455                                                                  TRAP    C$ERDF
023520  000145                                                                  .WORD   101
023522  024227                                                                  .WORD   T13SSR
023524  012046                                                                  .WORD   PKTSSR
023526  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
023532                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

```
                023532  104406                                                          TRAP    C$CLP1
     79 023534  016501  000002          MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
     80 023540  012702  000200          MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
     81 023544  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
     82 023550  001402                  BEQ     25$                     ;BRANCH IF NOT OFF-LINE
     83 023552  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 023556                      25$:
     87 023556                          FORCERROR       27$                     ;BAD
     88 023572  020201                  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
     89 023574  001404                  BEQ     30$                     ;OKAY IF MATCH
     90 023576                          NEXT.ERRNO
     91 023576                      27$:    ERRHRD  ERRNO,T13NBA,PKTSSR     ;NBA NOT ZERO
                023576  104456                                                          TRAP    C$ERHRD
                023600  000146                                                          .WORD   102
                023602  024154                                                          .WORD   T13NBA
                023604  012046                                                          .WORD   PKTSSR
     92 023606                      30$:    CKLOOP                          ;LOOP ON ERROR ?
                023606  104406                                                          TRAP    C$CLP1
     93
     94                              ;Write to TSSR register to soft initialize the controller
     95 023610                      40$:
     96 023610  004737  015774          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
     97 023614                          FORCERROR       42$                     ;BAD
     98 023630  103405                  BCS     50$                     ;BR IF SOFT INIT OKAY
     99 023632  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
    100 023634                          NEXT.ERRNO
    101 023634                      42$:    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
                023634  104455                                                          TRAP    C$ERDF
                023636  000147                                                          .WORD   103
                023640  003652                                                          .WORD   SFIERR
                023642  012034                                                          .WORD   SFIMSG
    102
    103                             ;If controller RAM 310-377 NOT=0 then Print Error
    104 023644  012704  000310      50$:    MOV     #310,R4                 ;START WITH LOC 310
    105 023650  005002                  CLR     R2                      ;MEMORY EXPECTED SHOULD BE 000000
    106 023652  105065  000000          CLRB    TSDB(R5)                ;SET MAINTENANCE MODE
    107 023656  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR READY
    108 023662  010465  000000      60$:    MOV     R4,TSDB(R5)             ;SELECT RAM ADDRESS
    109 023666  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR READY
    110 023672  116501  000000          MOVB    TSBA(R5),R1             ;READ LOC CONTENTS
    111 023676                          FORCERROR       62$,NOTSSR              ;BAD
    112 023706  120102                  CMPB    R1,R2                   ;CHECK MEMORY FOR 000000
    113 023710  001406                  BEQ     70$                     ;BRANCH IF DATA OKAY
    114 023712                          NEXT.ERRNO
    115 023712                      62$:    ERRDF   ERRNO,T13MEM,RAMEXP     ;MEMORY NOT ZERO AFTER INIT.
                023712  104455                                                          TRAP    C$ERDF
                023714  000150                                                          .WORD   104
                023716  024115                                                          .WORD   T13MEM
                023720  015530                                                          .WORD   RAMEXP
    116 023722  005237  002222          INC     FATFLG                  ;SET THE FATAL ERROR FLAG
    117 023726                      70$:    CKLOOP
                023726  104406                                                          TRAP    C$CLP1
    118 023730                          ESCAPE  TST                     ;EXIT ON FATAL ERROR
                023730  104410                                                          TRAP    C$ESCAPE
                023732  000432                                                          .WORD   L10036 .
```

```
023734  005204                 82$:    INC     R4                      ;LOOK AT NEXT RAM LOC.
023736  020427  000400                 CMP     R4,#400                 ;AT TOP OF RAM ADDRESS SPACE
023742  001347                         BNE     60$                     ;BRANCH TILL ALL MEMORY TESTED
023744  005737  002222                 TST     FATFLG                  ;ANY FATAL ERRORS ?
023750  001402                         BEQ     160$                    ;BRANCH IF NOT
023752  004737  017202                 JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
023756  004737  016456         160$:   JSR     PC,TSTLOOP              ;DONE ALL ITERATIONS?
023762  103002                         BCC     165$                    ;BR IF YES
023764  000137  023420                 JMP     T13LOOP                 ;LOOP UNTIL ITERATION COUNT DONE
023770                         165$:
023770                                 EXIT    TST
023770  104432                                                                          TRAP    C$EXIT
023772  000372                                                                          .WORD   L10036-.
                               ;+
                               ;LOCAL STORAGE FOR THIS TEST
                               ;-
        024000                         .=<.+10>&177770
024000                         T13PACKET:                              ;COMMAND PACKET FOR TEST
024000  100004                         .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
024002  024010                         .WORD   T13DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
024004  000000                         .WORD   0
024006  000010                         .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
024010                         T13DATA:                                        ;CHARACTERISTICS DATA BLOCK
024010  024022                         .WORD   T13BFR                  ;ADDRESS OF MESSAGE BUFFER
024012  000000                         .WORD   0
024014  000016                         .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
024016  000000  000000                 .WORD   0,0
024022                         T13BFR: .BLKW   8.                      ;MESSAGE BUFFER
                               ;LOCAL TEXT MESSAGES FOR TEST
                               ;-
024042     111     156     151 TST13ID:        .ASCIZ  'Initialization After WRITE CHARACTERISTICS
024115     111     156     143 T13MEM: .ASCIZ  'Incorrect RAM Data After Init'
                                       .EVEN
024154     127     122     111 T13NBA: .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
024227     103     157     156 T13SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS
                               ;+
                               ;
                               ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
                               ;
                               ;-
                               .EVEN
024316                         T13REST:
024316                                 SAVREG                  ;SAVE THE REGISTERS
024322  012701  024000                 MOV     #T13PACKET,R1   ;START OF THE PACKET
```

```
    176 024326  012721  100004                  MOV     #100004,(R1)+   ;WRITE CHARACTERISTICS WITH ACK
    177 024332  012721  024010                  MOV     #T13DATA,(R1)+  ;ADDRESS OF CHAR DATA BLOCK
    178 024336  005021                          CLR     (R1)+           ;EXTENDED ADDRESS
    179 024340  012721  000010                  MOV     #8.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
    180 024344  012721  024022                  MOV     #T13BFR,(R1)+   ;ADDRESS OF MESSAGE BUFFER
    181 024350  005021                          CLR     (R1)+
    182 024352  012721  000016                  MOV     #14.,(R1)+      ;LENGTH OF MESSAGE BUFFER
    183 024356  005021                          CLR     (R1)+
    184 024360  005011                          CLR     (R1)
    185 024362  000207                          RTS     PC              ;RETURN
    186 024364                                  ENDTST
        024364                                                                      L10036:
        024364  104401                                                                 TRAP    C$ETST
    187
    188                                         .SBTTL  TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND
    189                                 ;+
    190                                 ;
    191                                 ;THIS TEST VERIFIES THAT THE WRITE SUBSYSTEM MEMORY COMMAND WITH A
    192                                 ;BSEL0 SELECT CODE OF 0 (NO-OP) EXECUTES CORRECTLY. IT ALSO
    193                                 ;VERIFIES THAT A WRITE SUBSYSTEM MEMORY COMMAND WITH A NON-ZERO
    194                                 ;MODE FIELD IS REJECTED. THE TEST FURTHER VERIFIES MICROPROGRAM
    195                                 ;COMMAND DECODING AND HANDLING SEQUENCES.
    196                                 ;
    197                                 ;-
    198 024366                                  BGNTST
        024366                                                                      T2::
    203 024366  012700  026201                  MOV     #TST14ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
    204 024372  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
    205 024376  012737  000024  002216          MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
    206 024404                                  T14LOOP:
    207 024404                                  BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
        024404                                                                      T2.1:
        024404  104402                                                              TRAP    C$BSUB
    208 024406  004737  026246                  JSR     PC,T14REST              ;SET PACKET TO INITIAL VALUES
    209 024412  004737  026304                  JSR     PC,T14RST               ;SET PACKET TO INITIAL VALUES
    210 024416                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
        024416  012700  000000                                                      MOV     #PRI00,R0
        024422  104441                                                              TRAP    C$SPRI
    211 024424                                  5$:
    212 024424                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
        024424  104404                                                              TRAP    C$BSEG
    213 024426  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
    214 024432  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
    218 024434  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
    219 024436                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
        024436  104455                                                              TRAP    C$ERDF
        024440  000311                                                              .WORD   201
        024442  003652                                                              .WORD   SFIERR
        024444  012034                                                              .WORD   SFIMSG
    220 024446                                  10$:
    221 024446  012704  025540                  MOV     #T14PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
    222 024452  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
    223 024456  103405                          BCS     11$                     ;BR, IF COMMAND ISSUED OK
    227 024460  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
    228 024462                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
        024462  104456                                                              TRAP    C$ERHRD
        024464  000312                                                              .WORD   202
```

```
                                                                                .WORD   WRTMSG
                                                                                .WORD   SFIMSG
                        1$:
                        CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
                        CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
                        MOV     #T14PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
                        JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                        ERRDF   ERRNO,T14SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                                                                                TRAP    C$ERDF
                                                                                .WORD   203
                                                                                .WORD   T14SSR
                                                                                .WORD   PKTSSR
                        INC     FATFLG                  ;SET FATAL ERROR FLAG
                15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                TRAP    C$CLP1
                        ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
                                                                                TRAP    C$ESCAPE
                                                                                .WORD   10000$ .
                        TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
                        BNE     22$                     ;BRANCH IF YES
                        ERRHRD  ERRNO,T14NINT,PKTSSR
                                                                                TRAP    C$ERHRD
                                                                                .WORD   204
                                                                                .WORD   T14NINT
                                                                                .WORD   PKTSSR
                22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
                        MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
                        BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
                        BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
                25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
                        BEQ     30$                     ;OKAY IF MATCH
                        ERRHRD  ERRNO,T14NBA,PKTSSR     ;NBA NOT ZERO
                                                                                TRAP    C$ERHRD
                                                                                .WORD   205
                                                                                .WORD   T14NBA
                                                                                .WORD   PKTSSR
                30$:
                35$:    JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
                        BCS     59$                     ;RAM OK GO ON
                        ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
                                                                                TRAP    C$ERHRD
                                                                                .WORD   206
                                                                                .WORD   PKTRAM
                                                                                .WORD   RAMERR
                        ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<<
                                                                        10000$:
                                                                                TRAP    C$ESEG

                59$:    ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
                                                                        L10040:
                                                                                TRAP    C$ESUB
```

```
                                TST     FATFLG                  ;ANY FATAL ERRORS ?
                                BEQ     60$                     ;BRANCH IF NOT
                                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
                        60$:
                        ;*
                        ;
                        ;TEST 2, SUBTEST 2
                        ;
                        ;CHECK THAT NON-ZERO MODE BITS BEING SET CAUSE
                        ;WRITE SUBSYSTEM MEMORY COMMAND TO BE REJECTED
                        ;
                        ;

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

                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
                                                                                        MOV     #PRI00,R0
                                                                                        TRAP    C$SPRI
                        5$:     MOV     #T14PK2,R4              ;GET THE ADDRESS OF COMMAND PACKET
                                JSR     PC,T14REST              ;RESTORE PACKET TO STARTING VALUES
                                JSR     PC,T14RST               ;RESTORE PACKET TO STARTING VALUES
                                BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
                                                                                        TRAP    C$BSEG

                                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
                                BCS     10$                     ;BR IF SOFT INIT = OK
                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
                                                                                        TRAP    C$ERDF
                                                                                        .WORD   207
                                                                                        .WORD   SFIERR
                                                                                        .WORD   SFIMSG
                        10$:
                                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
                                BCS     11$                     ;BR, IF COMMAND ISSUED OK
                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE SUBSYSTEM MEMORY FAILED
                                                                                        TRAP    C$ERHRD
                                                                                        .WORD   208
                                                                                        .WORD   WRTMSG
                                                                                        .WORD   SFIMSG
                        11$:
                                CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
                                MOV     #T14PACKET,R4           ;SET-UP WITH WRT SUBSYS MEM PACKET
                                BIS     #007000,(R4)            ;NON-ZERO COMMAND MODE BITS
                                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
                                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
                                BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                ERRDF   ERRNO,T14SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                                                                                        TRAP    C$ERDF
                                                                                        .WORD   209
                                                                                        .WORD   T14SSR
                                                                                        .WORD   PKTSSR
                        15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

```
                025002  104406                                                          TRAP    C$CLP1
322             025004                          ESCAPE  SEG             ;BY-PASS CHECKS IF FATAL ERROR
                025004  104410                                                          TRAP    C$ESCAPE
                025006  000074                                                          .WORD   10000$ .
323             025010  005737  002224          TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
324             025014  001004                  BNE     22$             ;BRANCH IF YES
328             025016                          ERRHRD  ERRNO,T14NINT,PKTSSR
                025016  104456                                                          TRAP    C$ERHRD
                025020  000322                                                          .WORD   210
                025022  026023                                                          .WORD   T14NINT
                025024  012046                                                          .WORD   PKTSSR
329             025026  016501  000002  22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
330             025032  012702  100206          MOV     #SC!SSR!TSREJ,R2 ;EXPECTED CONTENTS OF TSSR
331             025036  032701  000100          BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
332             025042  001402                  BEQ     25$             ;BRANCH IF NOT OFF-LINE
333             025044  052702  000100          BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
334             025050  020201          25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
335             025052  001404                  BEQ     30$             ;OKAY IF MATCH
339             025054                          ERRHRD  ERRNO,T142REJ,PKTSSR ;COMMAND NOT REJECTED
                025054  104456                                                          TRAP    C$ERHRD
                025056  000323                                                          .WORD   211
                025060  025636                                                          .WORD   T142REJ
                025062  012046                                                          .WORD   PKTSSR
340             025064                  30$:
341             025064  004737  011114  35$:    JSR     PC,CKRAM        ;CHECK RAM TO MEMORY
342             025070  103405                  BCS     59$             ;RAM OK GO ON
346             025072                          ERRHRD  ERRNO,PKTRAM,RAMERR ;THEY DON'T MATCH
                025072  104456                                                          TRAP    C$ERHRD
                025074  000324                                                          .WORD   212
                025076  004745                                                          .WORD   PKTRAM
                025100  015510                                                          .WORD   RAMERR
347             025102                          ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                025102                                                          10000$:
                025102  104405                                                          TRAP    C$ESEG
348
349             025104                  59$:    ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                025104                                                          L10041:
                025104  104403                                                          TRAP    C$ESUB
350             025106                          EXIT    TST             ;ALL DONE THIS TEST
                025106  104432                                                          TRAP    C$EXIT
                025110  001246                                                          .WORD   L10037-.
351
352                             ;+
353                             ;LOCAL STORAGE FOR THIS TEST
354                             ;-
355
357             025120                  .=<.+10>&177770
359             025120          T14PACKET:                              ;COMMAND PACKET FOR TEST
360             025120  100206          .WORD   100206                  ;WRITE SUBSYSTEM MEM. COMMAND, WITH IE, ACK
361             025122  025130          .WORD   T14DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
362             025124  000000          .WORD   0
363             025126  000006          .WORD   6.                      ;STARTING VALUE OF BLOCK SIZE
364             025130          T14DATA:                                ;CHARACTERISTICS DATA BLOCK
365             025130     000  T14BS0: .BYTE   0                       ;BSEL0 BYTE
366             025131     000  T14BS1: .BYTE   0                       ;BSEL1 BYTE
367             025132  000000  T14BS2: .WORD   0                       ;BSEL1 WORD
368             025134  000000          .WORD   0                       ;DATA
```

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

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


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


                                ;+
                                ;LOCAL TEXT MESSAGES FOR TEST
                                ;-

                                T14NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
                                T142REJ:        .ASCIZ  'WRITE SUBSYSTEM MEMORY Not Rejected With Non-Zero Mode Field'
                                T14SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
                                T14NINT:        .ASCIZ  'Expected Interrupt Not Received On WRITE SUBSYSTEM MEMORY'
                                T14TSBA:        .ASCIZ  'Incorrect TSBA Address After WRITE SUBSYSTEM MEMORY'
                                TST14ID:        .ASCIZ  'Basic WRITE SUBSYSTEM MEMORY Command'
                                        .EVEN


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

                                T14REST:
                                        SAVREG                          ;SAVE THE REGISTERS
                                        MOV     #T14PACKET,R1           ;START OF THE PACKET
                                        MOV     #100206,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK, IE
                                        MOV     #T14DATA,(R1)+          ;ADDRESS OF DATA BLOCK
                                        CLR     (R1)+                   ;EXTENDED ADDRESS
                                        MOV     #6.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
                                        CLR     (R1)+                   ;CLEAR BSELO AND BSEL1
                                        CLR     (R1)+                   ;CLEAR SEL2
                                        CLR     (R1)                    ;CLEAR DATA AREA
                                        RTS     PC                      ;RETURN


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

```
428 026326  012721  000010          MOV     #8.,(R1)+           ;SIZE OF DATA BLOCK IN BYTES
429 026332  012721  025136          MOV     #T14BFR,(R1)+       ;MESSAGE BUFFER ADDRESS
430 026336  005021                  CLR     (R1)+
431 026340  012721  000400          MOV     #256.,(R1)+         ;LENGTH OF MESSAGE BUFFER
432 026344  005021                  CLR     (R1)+
433 026346  005011                  CLR     (R1)
434 026350  005037  025136          CLR     T14BFR              ;CLEAR 1ST LOC IN MESSAGE BUFFER
435 026354  000207                  RTS     PC                  ;RETURN
436 026356                          ENDTST
    026356                                                          L10037:
    026356  104401                                                          TRAP    C$ETST
```

```
                                        .SBTTL  TEST  3: DMA MEMORY ADDRESSING

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

026360                                  BGNTST
026360                                                                  T3::
026360  012700  030430                  MOV     #TST12ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
026364  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
026370  012737  000012  002216          MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
026376  005237  003150                  INC     T38FLG                  ;SET TEST FLAG
026402  004737  021232                  JSR     PC,MEMCK                ;CHECK MEMORY

026406                          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
                                        ;
                                            BGNSUB                          ;////////////// BEGIN SUBTEST /////////.
026406                                                                                      T3.1:
026406  104402                                                                                TRAP    C$BSUB
                                        ;Write to TSSR to soft initialize
026410  004737  015774                      JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
026414  103405                              BCS     15$                     ;BR IF SOFT INIT = OK
026416                                      NEXT.ERRNO
026416  010001                              MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
026420                                      ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
026420  104455                                                                              TRAP    C$ERDF
026422  000455                                                                              .WORD   301
026424  003652                                                                              .WORD   SFIERR
026426  012034                                                                              .WORD   SFIMSG
                                        ;Do a WRITE CHARACTERISTICS to setup a message buffer
026430                                  15$:
026430  012704  030220                      MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
026434  004737  031600                      JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
026440  005037  003134                      CLR     KTENABLE                ;TURN OFF KT-11
026444  010465  000000                      MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
026450  004737  016336                      JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
026454                                      FORCERROR       17$
026470  103405                              BCS     20$                     ;BR IF SSR SET IN CHKTSSR
026472  010001                              MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
026474                                      NEXT.ERRNO
026474                                  17$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
026474  104455                                                                              TRAP    C$ERDF
026476  000456                                                                              .WORD   302
026500  030532                                                                              .WORD   T12WRTSSR
026502  012046                                                                              .WORD   PKTSSR
                                        ;Verify a Get Status can be fetched from each address
                                        ;Get a valid modulo-4 test address
                                        ;Do a GET STATUS command from the test address
026504  005037  002222                  20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
026510  005037  030270                      CLR     T12KT                   ;TEST ABOVE 28K SWITCH
026514  012702  030274                      MOV     #T12BLK,R2              ;POINT TO TEST PATTERN TABLE
026520                                  T121LOOP:
026520  005037  003134                      CLR     KTENABLE                ;T  N OFF ABOVE 28K TEST FLAG
026524  012201                              MOV     (R2)+,R1                    TEST PATTERN ADDRESS
026526  005000                              CLR     R0                      .  SUME NO TEST ABOVE 28K
026530  005737  030270                      TST     T12KT                   ;tST ABOVE 28K THIS TIME?
026534  001407                              BEQ     25$                     ;BR IF NO
026536  016200  177776                      MOV     -2(R2),R0               ;GET TEST PATTERN AGAIN
026542  042700  177774                      BIC     #^C<A1716>,R0           ;SAVE 18 BIT ADDRESS ONLY
026546  012737  000001  003134              MOV     #1,KTENABLE             ;TURN ON ABOVE 28K TEST FLAG
026554  004737  031276                  25$:    JSR     PC,T12CONVERT           ;CONVERT TEST PATTERN TO TEST ADDRESS
```

```
                        BCC     65$                     ;BR IF INVALID PACKET ADDRESS
                        MOV     T12LOADD,R4             ;COPY CURRENT PACKET LOW ADDRESS
                        MOV     T12HIADD,R3             ;COPY CURRENT PACKET HIGH ADDRESS
                        JSR     PC,T12SETGET            ;SETUP CURRENT PACKET TO GET STATUS
                        BIC     #^C<A1716>,R3           ;SAVE ADDRESS BITS 17-16
                        BIS     R3,R4                   ;SETUP 18 BIT PACKET ADDRESS
                        JSR     PC,KTOFF                ;TURN OFF KT-11
                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
                        JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
                        FORCERROR       32$
                        BCS     40$                     ;BR IF SSR SET IN CHKTSSR
                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                        NEXT.ERRNO
                32$:    ERRDF   ERRNO,T12GETSSR,PKTGETS ;DEVICE FATAL SSR FAILED TO SET
                                                                TRAP    C$ERDF
                                                                .WORD   303
                                                                .WORD   T12GETSSR
                                                                .WORD   PKTGETS
                40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                TRAP    C$CLP1
                65$:
                        FORCEXIT        80$
                        CMP     R2,#T12TBE              ;DONE ALL TSTBLK TEST PATTERNS?
                        BHIS    70$                     ;BR IF YES
                        JMP     T121LOOP                ;DO ANOTHER MODULO  4 ADDRESS
                70$:    TST     T12KT                   ;DONE ABOVE 28K TESTING TOO?
                        BGT     80$                     ;BR IF YES
                        TST     KTFLG                   ;ANY MEMORY ABOVE 28K ON SYST M/
                        BEQ     80$                     ;BR IF NO
                        MOV     #1,T12KT                ;SET SWITCH
                        MOV     #T12BLK,R2              ;RESET TEST PATTERN TABLE
                        JMP     T121LOOP                ;DO ABOVE 28K TESTING
                80$:    JSR     PC,KTOFF                ;TURN OFF KT11
                        ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
                                                                L10043:
                                                                TRAP    C$ESUB
                        TST     FATFLG                  ;ANY FATAL ERRORS ?
                        BEQ     100$                    ;BRANCH IF NOT
                        JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
                100$:

                        .SBTTL  TEST 3: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS

                ;++
                ; TEST 3: SUBTEST 2:
                ;
                ; SUBTEST DESCRIPTION:
                ;
                ;       This subtest verifies the controller can deposit message packets
                ;       to all available memory locations.
                ;       Write Characteristics commands are executed with message
                ;       buffer addresses set to various addresses in each available
                ;       memory location.
                ;       The various addresses are determined by floating a 1 then a 0
                ;       through the address bits.
                ;
                ; TEST STEPS:
                ;
```

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

026746                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
026746                                                                                  T3.2:
026746  104402                                                                          TRAP    C$BSUB

                                ;Write to TSSR to soft initialize
026750  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
026754  103405                          BCS     15$                     ;BR IF SOFT INIT = OK
026756                                  NEXT.ERRNO
026756  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
026760                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
026760  104455                                                                          TRAP    C$ERDF
026762  000460                                                                          .WORD   304
026764  003652                                                                          .WORD   SFIERR
026766  012034                                                                          .WORD   SFIMSG

                                ;Do a WRITE CHARACTERISTICS to setup a message buffer to compare
026770                          15$:
026770  012704  030220                  MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
026774  004737  031600                  JSR     PC,T12SWRT              ;SET PACKET TO WRITE CHARACTERISTICS
027000  004737  017274                  JSR     PC,KTOFF                ;TURN OFF KT-11
027004  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
027010  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
027014                                  FORCERROR       17$
027030  103405                          BCS     20$                     ;BR IF SSR SET IN CHKTSSR
027032  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
027034                                  NEXT.ERRNO
027034                          17$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
027034  104455                                                                          TRAP    C$ERDF
027036  000461                                                                          .WORD   305
027040  030532                                                                          .WORD   T12WRTSSR
027042  012046                                                                          .WORD   PKTSSR

                                ;Get a valid modulo-4 test address
                                ;Set the packet message buffer to the test address
                                ;Do a WRITE CHARACTERISTICS
027044  005037  002222          20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
027050  012703  030274                  MOV     #T12BLK,R3              ;POINT TO TEST PATTERN TABLE
027054                          T122LOOP:
027054  012301                          MOV     (R3)+,R1                ;GET TEST PATTERN ADDRESS
027056  010100                          MOV     R1,R0                   ;GET ADDRESS ALL "18 BITS"
027060  042700  177774                  BIC     #177774,R0              ;LEAVE ONLY A17 AND A16
027064  042701  000003                  BIC     #3,R1                   ;GET RID OF A17 AND A16
```

```
642 027070  004737  031276                  JSR     PC,T12CONVERT           ;CONVERT TEST PATTERN TO TEST ADDRESS
643 027074  103402                          BCS     25$                     ;BR IF VALID MESSAGE BUFFER ADDRESS
644 027076  000137  027174                  JMP     150$                    ;GET ANOTHER TEST PATTERN TO TRY
645 027102  012704  030220          25$:    MOV     #T12PACKET,R4           ;SET THE COMMAND PACKET ADDRESS
646 027106  004737  031600                  JSR     PC,T12SWRT              ;SETUP T12PACKET TO WRITE CHAR.
647 027112  013737  030264  030230          MOV     T12LOADD,T12DATA        ;SETUP LOW ORDER MESSAGE BUFFER ADD.
648 027120  013737  030262  030232          MOV     T12HIADD,T12DATA+2      ;SETUP HIGH ORDER MESSAGE BUFFER ADD.
649 027126  004737  017274                  JSR     PC,KTOFF                ;TURN OFF KT-11
650 027132  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
651 027136  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
652 027142                                  FORCERROR       32$
653 027156  103405                          BCS     50$                     ;BR IF SSR SET IN CHKTSSR
654 027160  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
655 027162                                  NEXT.ERRNO
656 027162                          32$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
    027162  104455                                                                  TRAP    C$ERDF
    027164  000462                                                                  .WORD   306
    027166  030532                                                                  .WORD   T12WRTSSR
    027170  012046                                                                  .WORD   PKTSSR
657 027172                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
    027172  104406                                                                  TRAP    C$CLP1
658 027174                          150$:
659 027174                                  FORCEXIT        160$
660 027204  020327  030426                  CMP     R3,#T12TBE              ;DONE ALL TST12BLK TEST PATTERNS?
661 027210  103002                          BHIS    160$                    ;BR IF YES
662 027212  000137  027054                  JMP     T122LOOP                ;DO ANOTHER MODULO  4 ADDRESS
663 027216  004737  017274          160$:   JSR     PC,KTOFF                ;TURN OFF KT11
664 027222                                  ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
    027222                                                                      L10044:
    027222  104403                                                                  TRAP    C$ESUB
665 027224  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
666 027230  001402                          BEQ     180$                    ;BRANCH IF NOT
667 027232  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
668 027236                          180$:
669
670
671
672                                         .SBTTL  TEST 3: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS
673                                 ;++
674                                 ; TEST 3: SUBTEST 3:
675                                 ;
676                                 ; SUBTEST DESCRIPTION:
677                                 ;
678                                 ;       This subtest verifies the controller can fetch a
679                                 ;       Write Characteristics data block from all available
680                                 ;       memory locations.
681                                 ;       Write Characteristics commands are executed with
682                                 ;       characteristic data blocks at various memory addresses.
683                                 ;       The various memory addresses are determined by floating
684                                 ;       a 1 then a 0 through the address bits.
685                                 ;
686                                 ; TEST STEPS:
687                                 ;
688                                 ;    BEGIN
689                                 ;       Write to TSSR to soft initialize
690                                 ;
691                                 ;       REPEAT FOR SELECTED VALID ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
```

```
                                ;          BEGIN
                                ;               Get a valid test address
                                ;               Set the test packet characteristics data pointer to the
                                ;                 test address.
                                ;               Store expected characteristic data in test address block
                                ;               Do a WRITE CHARACTERISTIC command
                                ;          END
                                ;     END
                                ;
                                        BGNSUB                          ;////////////// BEGIN SUBTEST //////////
027236                                                                                      T3.3:
027236                                                                                          TRAP    C$BSUB
027236  104402
                                ;Write to TSSR to soft initialize
027240  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
027244  103405                          BCS     20$                     ;BR IF SOFT INIT = OK
027246                                  NEXT.ERRNO
027246  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
027250                                  ERROF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
027250  104455                                                                              TRAP    C$EROF
027252  000463                                                                              .WORD   307
027254  003652                                                                              .WORD   SFIERR
027256  012034                                                                              .WORD   SFIMSG
                                ;Get a valid test address
027260  005037  002222          20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
027264  005037  030270                  CLR     T12KT                   ;TEST ABOVE 28K SWITCH
027270  012703  030274                  MOV     #T12BLK,R3              ;POINT TO TEST PATTERN TABLE
027274                          T123LOOP:
027274  005037  003134                  CLR     KTENABLE                ;TURN OFF ABOVE 28K TEST FLAG
027300  012301                          MOV     (R3)+,R1                ;GET TEST PATTERN ADDRESS
027302  010100                          MOV     R1,R0                   ;GET ADDRESS ALL "18 BITS"
027304  042700  177774                BIC     #177774,R0              ;LEAVE ONLY A17 AND A16
027310  042701  000003                BIC     #3,R1                   ;GET RID OF A17 AND A16
027314  005737  030270                TST     T12KT                   ;TEST ABOVE 28K THIS TIME?
027320  001407                          BEQ     25$                     ;BR IF NO
027322  016300  177776                MOV     -2(R3),R0               ;GET TEST PATTERN AGAIN
027326  042700  177774                BIC     #^C<A1716>,R0           ;SAVE 18 BIT ADDRESS ONLY
027332  012737  000001  003134        MOV     #1,KTENABLE             ;TURN ON ABOVE 28K TEST FLAG
027340  004737  031276          25$:    JSR     PC,T12CONVERT           ;CONVERT TEST PATTERN TO TEST ADDRESS
027344  103402                          BCS     30$                     ;BR IF VALID TEST ADDRESS
027346  000137  027450                  JMP     60$                     ;GET NEXT TEST PATTERN
                                ;Set the test packet characteristics data pointer to the test address
027352  012704  030220          30$:    MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
027356  004737  031600                  JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
027362  013764  030264  000002        MOV     T12LOADD,PKLOW(R4)      ;STORE CHAR. DATA PTR LOW ADDRESS
027370  013764  030262  000004        MOV     T12HIADD,PKHI(R4)       ;STORE CHAR. DATA PTR HIGH ADDRESS
027376  004737  031710                JSR     PC,T12CHAR              ;STORE EXPECTED DATA IN DATA BLOCK
                                ;Do a WRITE CHARACTERISTIC command
027402  004737  017274                JSR     PC,KTOFF                ;TURN OFF KT-11
027406  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
027412  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
027416                                  FORCERROR       32$
027432  103405                          BCS     40$                     ;BR IF SSR SET IN CHKTSSR
027434  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

```
743 027436                                  NEXT.ERRNO
744 027436                          32$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
    027436  104455                                                          TRAP    C$ERDF
    027440  000464                                                          .WORD   306
    027442  030532                                                          .WORD   T12WRTSSR
    027444  012046                                                          .WORD   PKTSSR
745 027446                          40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
    027446  104406                                                          TRAP    C$CLP1
746 027450                          60$:
747 027450  020327  030426                     CMP     R3,@T12TBE          ;DONE ALL TSTBLK TEST PATTERNS?
748 027454  103002                             BHIS    65$                 ;BR IF YES
749 027456  000137  027274                     JMP     T123LOOP            ;DO ANOTHER MODULO- 4 ADDRESS
750 027462  005737  030270          65$:       TST     T12KT               ;DONE ABOVE 28K TESTING TOO?
751 027466  003012                             BGT     70$                 ;BR IF YES
752 027470  005737  003132                     TST     KTFLG               ;ANY MEMORY ABOVE 28K ON SYSTEM?
753 027474  001407                             BEQ     70$                 ;BR IF NO
754 027476  012737  000001  030270             MOV     #1,T12KT            ;SET SWITCH
755 027504  012703  030274                     MOV     #T12BLK,R3          ;RESET TEST PATTERN TABLE
756 027510  000137  027274                     JMP     T123LOOP            ;DO ABOVE 28K TESTING
757 027514  004737  017274          70$:       JSR     PC,KTOFF            ;TURN OFF KT11
758 027520                                     ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\
    027520                                                                      L10045:
    027520  104403                                                          TRAP    C$ESUB
759 027522  005737  002222                     TST     FATFLG              ;ANY FATAL ERRORS ?
760 027526  001402                             BEQ     75$                 ;BRANCH IF NOT
761 027530  004737  017202                     JSR     PC,CKDROP           ;TRY TO DROP THE UNIT
762 027534                          75$:
763
764                                         .SBTTL  TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES
765                                 ;**
766                                 ; TEST 3: SUBTEST 4:
767                                 ;
768                                 ; SUBTEST DESCRIPTION:
769                                 ;
770                                 ;       This subtest verifies the NXM error bit in the TSSR
771                                 ;       register is set when attempting to fetch data (a characteristic
772                                 ;       data block) from selected nonexistent locations.
773                                 ;       If NXM fails to set it is likely that an LSI-11 Bus driver is
774                                 ;       failing to assert an address line.
775                                 ;       Addresses tested include all combinations of high-order address
776                                 ;       bits (i.e bits 16-21).
777                                 ;       ************************************************************************
778                                 ;                                   CAUTION
779                                 ;
780                                 ;               The LSI BUS drivers for all available address lines(16-21)
781                                 ;       are only checked when running on a 11/23B system with more than
782                                 ;       128K words of memory!
783                                 ;       ************************************************************************
784                                 ;
785                                 ; TEST STEPS:
786                                 ;
787                                 ;    BEGIN
788                                 ;       Write to TSSR to soft initialize
789                                 ;       Do a write characteristic command
790                                 ;       Invert the extended features switch
791                                 ;
792                                 ;       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
                                ;
                                
027534                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
027534                                                                                  T3.4:
027534  104402                                                                          TRAP    C$BSUB

027536  005737  003144                  TST     T23A                    ;26-APR-83 REV B - CHK FOR 23A CPU
027542  001406                          BEQ     5$                      ;26-APR-83 REV B - BR, IF NOT 23A
027544  023727  002120  007777          CMP     L$HIME,#7777            ;26-APR-83 REV B - CHK FOR > 256KB
027552  103402                          BLO     5$                      ;26-APR-83 REV B - BR, IF < 256KB
027554  000137  030146                  JMP     NOEXTF                  ;26-APR-83 REV B   JMP OVER 256KB
027560                          5$:
027560  005737  003136                  TST     NXMFLG                  ;GOT ENOUGH MEMORY?
027564  001002                          BNE     10$                     ;IF SET STAY
027566  000137  030146                  JMP     NOEXTF                  ;LEAVE IF NOT SET

                                ;Write to TSSR to soft initialize

027572  004737  015774          10$:    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
027576  103405                          BCS     11$                     ;BR IF SOFT INIT = OK
027600                                  NEXT.ERRNO
027600  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
027602                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
027602  104455                                                                          TRAP    C$ERDF
027604  000465                                                                          .WORD   309
027606  003652                                                                          .WORD   SFIERR
027610  012034                                                                          .WORD   SFIMSG

                                ;Do a WRITE CHARACTERISTIC command so to invert switch

027612                          11$:    CKLOOP                          ;LOOP IF SELECTED
027612  104406                                                                          TRAP    C$CLP1
027614  012704  030220                  MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
027620  004737  031600                  JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
027624  005037  003134                  CLR     KTENABLE                ;TURN OFF KT-11
027630  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
027634  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
027640                                  FORCERROR       15$
027654  103405                          BCS     17$                     ;BR IF SSR SET IN CHKTSSR
027656  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
027660                                  NEXT.ERRNO
027660                          15$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
027660  104455                                                                          TRAP    C$ERDF
027662  000466                                                                          .WORD   310
027664  030532                                                                          .WORD   T12WRTSSR
027666  012046                                                                          .WORD   PKTSSR
027670                          17$:    CKLOOP                          ;LOOP IF SELECTED
027670  104406                                                                          TRAP    C$CLP1
```

```
                                        JSR     PC,INVERT               ;INVERT THE SWITCH

                                ;Get an invalid test address

                        20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
                        25$:
                                MOV     NXMHI,T12HIADD          ;SAVE TEST ADDRESS HIGH
                                MOV     NXMLO,T12LOADD          ;SAVE TEST ADDRESS LOW
                        T124LOOP:

                                ;Set the test packet characteristics data pointer to the
                                ;  test address.

                        30$:    MOV     #T12PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
                                JSR     PC,T12SWRT              ;RESTORE PACKET TO STARTING VALUES
                                MOV     T12LOADD,PKLOW(R4)      ;STORE CHAR. DATA PTR LOW ADDRESS
                                MOV     T12HIADD,PKHI(R4)       ;STORE CHAR. DATA PTR HIGH ADDRESS

                                ;Do a WRITE CHARACTERISTIC command
                                JSR     PC,KTOFF                ;TURN OFF KT-11
                                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
                                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
                                FORCERROR       32$
                                BCS     40$                     ;BR IF SSR SET IN CHKTSSR
                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                NEXT.ERRNO
                        32$:    ERRDF   ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
                                                                        TRAP    C$ERDF
                                                                        .WORD   311
                                                                        .WORD   T12WRTSSR
                                                                        .WORD   PKTSSR
                                INC     FATFLG                  ;SET FATAL ERROR FLAG
                        40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                        TRAP    C$CLP1
                                FORCERROR       45$,NOTSSR
                                ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
                                                                        TRAP    C$ESCAPE
                                                                        .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$:    MOV     T12LOADD,ERRLO          ;MEMORY TEST ADDRESS LOW
                                MOV     T12HIADD,ERRHI          ;MEMORY TEST ADDRESS HIGH
                                ERRHRD  ERRNO,T12NXM,ADDSSR     ;REPORT ADDRESS AND TSSR ERROR
                                                                        TRAP    C$ERHRD
                                                                        .WORD   312
                                                                        .WORD   T12NXM
                                                                        .WORD   ADDSSR

                        60$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                        TRAP    C$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:
                                                                        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$:
                        JSR     PC,KTOFF                ;TURN OFF MEMORY MANAGEMENT
                        CLR     T3BFLG                  ;CLEAR TEST FLAG
                        EXIT    TST                     ;ALL DONE THIS TEST
                                                                TRAP    C$EXIT
                                                                .WORD   L10042-.


                ;*
                ;LOCAL STORAGE FOR THIS TEST
                ;
                        .=<.+10>&177770
                T12PACKET:                              ;COMMAND PACKET FOR TEST
                        .WORD   100004                  ;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
```

```
    938 030276  000002                                .WORD   000002
    939 030300  000003                                .WORD   000003
    940 030302  000005                                .WORD   000005
    941 030304  000006                                .WORD   000006
    942 030306  000007                                .WORD   000007
    943 030310  000011                                .WORD   000011
    944 030312  000012                                .WORD   000012
    945 030314  000013                                .WORD   000013
    946 030316  000021                                .WORD   000021
    947 030320  000022                                .WORD   000022
    948 030322  000023                                .WORD   000023
    949 030324  000041                                .WORD   000041
    950 030326  000042                                .WORD   000042
    951 030330  000043                                .WORD   000043
    952 030332  000101                                .WORD   000101
    953 030334  000102                                .WORD   000102
    954 030336  000103                                .WORD   000103
    955 030340  000201                                .WORD   000201
    956 030342  000202                                .WORD   000202
    957 030344  000203                                .WORD   000203
    958 030346  000401                                .WORD   000401
    959 030350  000402                                .WORD   000402
    960 030352  000403                                .WORD   000403
    961 030354  001001                                .WORD   001001
    962 030356  001002                                .WORD   001002
    963 030360  001003                                .WORD   001003
    964 030362  002001                                .WORD   002001
    965 030364  002002                                .WORD   002002
    966 030366  002003                                .WORD   002003
    967 030370  004001                                .WORD   004001
    968 030372  004002                                .WORD   004002
    969 030374  004003                                .WORD   004003
    970 030376  010001                                .WORD   010001
    971 030400  010002                                .WORD   010002
    972 030402  010003                                .WORD   010003
    973 030404  020001                                .WORD   020001
    974 030406  020002                                .WORD   020002
    975 030410  020003                                .WORD   020003
    976 030412  040001                                .WORD   040001
    977 030414  040002                                .WORD   040002
    978 030416  040003                                .WORD   040003
    979 030420  100001                                .WORD   100001
    980 030422  100002                                .WORD   100002
    981 030424  100003                                .WORD   100003
    982 030426  177777                    T12TBE:     .WORD           177777
    983                                   ;+
    984                                   ;LOCAL TEXT MESSAGES FOR TEST
    985                                   ;-
    986
    987 030430      104     115     101   TST12ID:          .ASCIZ  'DMA Memory Addressing'
    988 030456      103     157     156   T12GETSSR:        .ASCIZ  'Contents of TSSR Incorrect After GET STATUS'
    989 030532      103     157     156   T12WRTSSR:        .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
    990 030621      115     145     163   T12MSGBUF:        .ASCIZ  'Message Buffer Contents Incorrect After WRITE CHARACTERISTICS'
    991 030717      102     141     143   T12BKGND:         .ASCIZ  'Background Pattern Disturbed By WRITE CHARACTERISTICS'
    992 031005      105     170     160   T12NINT:          .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
    993 031076      127     162     151   T12DPR:           .ASCIZ  'Write Characteristic data in ram does not match expected'
    994 031167      124     123     123   T12NXM:           .ASCIZ  'TSSR NXM bit failed to set when non-existent memory address specifi
ed'
```

```
                                        .EVEN


                                ;+
                                ;ROUTINE TO CONVERT A TEST PATTERN TO A VALID ADDRESS IN DIAGNOSTIC FREE SPACE
                                ;       DIAGNOSTIC FREE SPACE IS BETWEEN THE END OF THE DIAGNOSTIC AND THE
                                ;       BEGINNING OF THE SUPERVISOR. THIS IS ALWAYS BELOW 24K.
                                ;       IF MEMORY ABOVE 28K SPECIFIED (VIA R1) THEN PAR 6 IS SET
                                ;       TO THE RELOCATION BASE.
                                ;
                                ; INPUTS:
                                ;
                                ;       R0      HIGH ORDER ADDRESS BITS
                                ;       R1      LOW ORDER ADDRESS BITS
                                ;
                                ; OUPTUTS:
                                ;       T12PAR6 - ADDRESS BIASED TO PAR6 IF >28K UNDER TEST
                                ;       T12HIADD - HIGH ORDER ADDRESS IN NON PAR6 FORMAT
                                ;       T12LOADD - LOW ORDER ADDRESS IN NON PAR6 FORMAT
                                ;       C BIT = 1 IF GOOD ADDRESS RETURNED
                                ;       C BIT = 0 IF TEST PATTERN DID NOT YIELD A VALID ADDRESS
                                ;-
031276                          T12CONVERT:
031276                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
031302  005037  030264                  CLR     T12LOADD        ;CLEAR LOW ADDRESS
031306  005037  030262                  CLR     T12HIADD        ;CLEAR HIGH ADDRESS
031312  005037  030266                  CLR     T12PAR6         ;CLEAR PAR6 BIASED ADDRESS
031316  042701  170000                  BIC     #^C<7777>,R1    ;FORCE TO LOWER 12 BITS OF ADDRESS
031322  010005                          MOV     R0,R5           ;SAVE HIGH ORDER ADDRESS BITS
031324  004737  017274                  JSR     PC,KTOFF        ;SHUTOFF MEMORY MANAGEMENT
031330  013702  003124                  MOV     FREE,R2         ;GET FIRST FREE ADDRESS
031334  062702  000020                  ADD     #16.,R2         ;IN CASE TEST PATTERN=0
031340  060102                          ADD     R1,R2           ;ADD IN TEST PATTERN
031342  042702  000003                  BIC     #3,R2           ;MAKE IT MODULO-4
031346  013703  003130          25$:    MOV     FREEHI,R3       ;GET LAST FREE ADDRESS
031352  162703  000020                  SUB     #16.,R3         ;SAVE AT LEAST 8 WORDS (IN CASE MESSAGE BUFFER)
031356  010237  030264                  MOV     R2,T12LOADD     ;SAVE POSSIBLE LOW ADDRESS
031362  010237  030266                  MOV     R2,T12PAR6      ;SAVE IT IN PAR6 BIASED TOO
031366  020203                          CMP     R2,R3           ;IS THIS ADDRESS ABOVE FREE SPACE?
031370  101007                          BHI     35$             ;BR IF YES
031372  020237  003124                  CMP     R2,FREE         ;IS IT IN FREE SPACE?
031376  103007                          BHIS    50$             ;BR IF YES- ITS GOOD
031400  005737  003134                  TST     KTENABLE        ;TESTING ABOVE 28K?
031404  001004                          BNE     50$             ;BR IF YES
031406  000424                          BR      90$             ;BR IF NOT IN FREE SPACE
031410  162702  000020          35$:    SUB     #16.,R2         ;FORCE FIT THE TEST PATTERN
031414  000754                          BR      25$             ;TRY THIS TEST PATTERN ADDRESS
031416                          50$:
031416  005737  003134                  TST     KTENABLE        ;TESTING ABOVE 28K?
031422  001420                          BEQ     100$            ;BR IF NO
031424  005737  003132                  TST     KTFLG           ;ANY MEMORY ABOVE 28K?
031430  001413                          BEQ     90$             ;BR IF NO
031432  004737  017256                  JSR     PC,KTON         ;TURN ON MEMORY MANAGEMENT
031436  010500                          MOV     R5,R0           ;GET HIGH ORDER ADDRESS
031440  010037  030262                  MOV     R0,T12HIADD     ;SAVE POSSIBLE HIGH ADDRESS
031444  010201                          MOV     R2,R1           ;GET COMPUTED LOW ORDER ADDRESS
031446  004737  017316                  JSR     PC,SETMAP       ;RETURN PAR6 BIASED ADDRESS IN R0
```

```
1052 031452  010037  030266                    MOV     R0,T12PAR6      ;COPY PAR6 BIASED ADDRESS
1053 031456  103403                            BCS     105$            ;BR IF VALID ADDRESS
1054 031460  000241                  90$:      CLC                     ;CLR C BIT FOR FAILURE
1055 031462  000401                            BR      105$            ;
1056 031464  000261                  100$:     SEC                     ;SET SUCCESS
1057 031466  000207                  105$:     RTS     PC              ;RETURN
1058
1059
1060                                  ;*
1061                                  ;
1062                                  ;ROUTINE TO READ THE FIRST 2 BYTES FROM RAM
1063                                  ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
1064                                  ;
1065                                  ;INPUT:
1066                                  ;
1067                                  ;       R4      ADDRESS OF THE COMMAND PACKET
1068                                  ;       R5      FIRST DEVICE UNIBUS ADDRESS
1069                                  ;
1070                                  ;OUTPUT:
1071                                  ;
1072                                  ;       CARRY   SET - RAM MATCHES PACKET
1073                                  ;               CLR - RAM DOES NOT MATCH PACKET
1074                                  ;
1075                                  ;IMPLICIT OUTPUT:
1076                                  ;
1077                                  ;       THE TABLE RAMDATA IS FILLED WITH THE
1078                                  ;       DATA HELD IN RAM.
1079                                  ;       RAMSIZ  SET TO 2 FOR PRAMPKT ROUTINE
1080                                  ;
1081                                  ;SIDE EFFECTS:
1082                                  ;
1083                                  ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1084                                  ;
1085                                  ;-
1086 031470                          T12CKRAM::
1087 031470                                  SAVREG                  ;SAVE THE GENERAL REGISTERS
1088 031474  012701  002242                  MOV     #RAMDATA,R1     ;ADDRESS TO SAVE THE RAM DATA
1089 031500  012702  000201                  MOV     #RMPKTBEG,R2    ;BYTE ADDRESS OF FIRST RAM DATA
1090 031504  005003                          CLR     R3              ;CLEAR THE ERROR FLAG
1091 031506  004737  016336                  JSR     PC,CHKTSSR      ;WAIT FOR SSR
1092 031512  112765  000000  000000          MOVB    #0,TSDB(R5)     ;SET MAINTENANCE MODE
1093 031520  004737  016336          10$:    JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
1094 031524  010265  000000                  MOV     R2,TSDB(R5)     ;SELECT NEXT RAM ADDRESS
1095 031530  004737  016336                  JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
1096 031534  116511  000000                  MOVB    TSBA(R5),(R1)   ;READ THE RAM DATA
1097 031540  122124                          CMPB    (R1)+,(R4)+     ;COMPARE TO EXPECTED
1098 031542  001401                          BEQ     20$             ;BRANCH IF OK
1099 031544  005203                          INC     R3              ;SET ERROR FLAG
1100 031546  005202                  20$:    INC     R2              ;ADDRESS OF NEXT RAM LOCATION
1101 031550  020227  000203                  CMP     R2,#RMPKTBEG+2  ;DONE 2 BYTES?
1102 031554  002761                          BLT     10$             ;BR IF NO
1103 031556  005703                          TST     R3              ;WAS AN ERROR FOUND ?
1104 031560  001402                          BEQ     30$             ;BRANCH IF NOT
1105 031562  000241                          CLC                     ;CLEAR CARRY TO SHOW ERROR
1106 031564  000401                          BR      50$             ;AND EXIT
1107 031566  000261                  30$:    SEC                     ;SHOW GOOD COMPARE
1108 031570  012737  000002  002302  50$:    MOV     #2,RAMSIZ       ;SETUP RAMSIZ
```

```
1109 031576  000207                        RTS     PC                      ;RETURN
1110
1111                                ;+
1112                                ;
1113                                ;ROUTINE TO SETUP PACKET TO WRITE CHARACTERISTICS
1114                                ;-
1115
1116 031600                         T12SWRT:
1117 031600                                 SAVREG                          ;SAVE THE REGISTERS
1118 031604  012701  030220                 MOV     #T12PACKET,R1           ;START OF THE PACKET
1119 031610  012721  100004                 MOV     #100004,(R1)+           ;WRITE CHARACTERISTICS WITH ACK
1120 031614  012721  030230                 MOV     #T12DATA,(R1)+          ;ADDRESS OF CHAR DATA BLOCK
1121 031620  005021                         CLR     (R1)+                   ;EXTENDED ADDRESS
1122 031622  012721  000010                 MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
1123 031626  012721  030242                 MOV     #T12BFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
1124 031632  005021                         CLR     (R1)+
1125 031634  012721  000016                 MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
1126 031640  005021                         CLR     (R1)+
1127 031642  005011                         CLR     (R1)
1128 031644  000207                         RTS     PC                      ;RETURN
1129                                ;+
1130                                ;
1131                                ;ROUTINE TO SETUP A GET STATUS COMMAND PACKET AT CURRENT PACKET ADDRESS
1132                                ;
1133                                ;       R3      HIGH ORDER PACKET ADDRESS
1134                                ;       R4      LOW ORDER PACKET ADDRESS
1135                                ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG CLEAR
1136                                ;
1137                                ;
1138
1139 031646                         T12SETGET:
1140 031646                                 SAVREG                          ;SAVE THE REGISTERS
1141 031652  010401                         MOV     R4,R1                   ;GET LOW ORDER ADDRESS
1142 031654  005737  003134                 TST     KTENABLE                ;TESTING ABOVE 28K?
1143 031660  001404                         BEQ     10$                     ;BR IF NO
1144 031662  010300                         MOV     R3,R0                   ;GET HIGH ORDER ADDRESS
1145 031664  004737  017316                 JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
1146 031670  010001                         MOV     R0,R1                   ;GET ADDRESS
1147 031672  012700  000017         10$:    MOV     #P.GETSTATUS,R0         ;GET STATUS COMMAND CODE NO IE
1148 031676  052700  100000                 BIS     #P.ACK,R0               ;SET ACK
1149 031702  010021                         MOV     R0,(R1)+                ;STORE GET STATUS IN PACKET
1150 031704  005021                         CLR     (R1)+                   ;CLEAR UNUSED WORD
1151 031706  000207                         RTS     PC                      ;RETURN
1152
1153
1154                                ;+
1155                                ;
1156                                ;ROUTINE TO SETUP A CHARACTERISTIC DATA BLOCK AT A TEST ADDRESS
1157                                ;
1158                                ;-
1159 031710                         T12CHAR:
1160 031710                                 SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
1161 031714  012700  030230                 MOV     #T12DATA,R0             ;GET T12PACKET DATA POINTER
1162 031720  013701  030264                 MOV     T12LOADD,R1             ;ASSUME NOT ABOVE 28K
1163 031724  005737  003134                 TST     KTENABLE                ;TESTING ABOVE 28K?
1164 031730  001402                         BEQ     10$                     ;BR IF NO
1165 031732  013701  030266                 MOV     T12PAR6,R1              ;SET TEST ADDRESS ABOVE 28K
```

```
1166 031736  012021          10$:    MOV     (R0)+,(R1)+     ;STORE DATA WORD 1
1167 031740  012021                  MOV     (R0)+,(R1)+     ;STORE DATA WORD 2
1168 031742  012021                  MOV     (R0)+,(R1)+     ;STORE DATA WORD 3
1169 031744  012021                  MOV     (R0)+,(R1)+     ;STORE DATA WORD 4
1170 031746  012021                  MOV     (R0)+,(R1)+     ;STORE DATA WORD 5
1171 031750  000207                  RTS     PC              ;RETURN
1172
1173 031752                          ENDTST
     031752                                                              L10042:
     031752  104401                                                              TRAP    C$ETST
```

```
                                        .SBTTL  TEST  4: RAM EXERCISER TEST
                                ;+
                                ;
                                ;THIS TEST USES THE READ AND WRITE RAM (BOTH SINGLE AND 256
                                ;LOCATIONS) SELECT CODES OF THE WRITE SUBSYSTEM MEMORY COMMAND
                                ;TO EXERCISE THE CONTROLLER'S RAM MEMORY AND DMA LOGIC
                                ;
                                ;-
031754                                  BGNTST
031754                                                                          T4::
                                
                                
031754  005737  002214                  TST     TSTCNT                  ;CHECK FOR RUN MODE
031760  001402                          BEQ     10$                     ;BR, IF NOT ONLY PROGRAM RUN
031762  005237  003400                  INC     SKIPT                   ;SET SKIP SW
031766  012700  034413          10$:    MOV     #TST15ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
031772  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
031776  012737  000005  002216          MOV     #5,LOOPCNT              ;PERFORM 5 ITERATIONS
032004                          T15LOOP:
                                ;
                                ;
                                ;TEST 4, SUBTEST 1
                                ;
                                ;       THIS SUBTEST WRITES THE ADDRESS (8 BITS) INTO THE
                                ;       RAM MEMORY SINGLE WORD (8 BITS) MODE
                                ;
                                ;-
032004                                  BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
032004                                                                          T4.1:
032004  104402                                                                   TRAP    C$BSUB
032006                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
032006  012700  000000                                                          MOV     #PR100,R0
032012  104441                                                                   TRAP    C$SPRI
032014  005737  003400                  TST     SKIPT                   ;SHOULD WE SKIP THIS SUBTEST
032020  001402                          BEQ     10$                     ;BR, IF NOW SKIP REQUIRED
032022  000137  032304                  JMP     50$                     ;SKIP SUBTEST
032026  004737  034432          10$:    JSR     PC,T15REST              ;SET COMMAND PACKET
032032  004737  034504                  JSR     PC,T15RT2               ;SET UP OTHER COMMAND PACKET
032036  004737  015774                  JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
032042  103405                          BCS     20$                     ;BR IF INIT WAS OK
032044  010001                          MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
032046                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
032046  104455                                                                   TRAP    C$ERDF
032050  000621                                                                   .WORD   401
032052  003652                                                                   .WORD   SFIERR
032054  012034                                                                   .WORD   SFIMSG
032056                          20$:
032056  012704  033330                  MOV     #T15PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
032062  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
032066  103405                          BCS     23$                     ;BR, IF COMMAND ISSUED OK
032070  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
032072                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
032072  104456                                                                   TRAP    C$ERHRD
032074  000622                                                                   .WORD   402
032076  005056                                                                   .WORD   WRTMSG
032100  012034                                                                   .WORD   SFIMSG
```

```
1229 032102  012703  000400          23$:   MOV     #256.,R3              ;STARTING ADDRESS FOR RAM WRITE
1230 032106  112737  000001  034041         MOVB    #1,T15BS1             ;SIZE OF TRANSFER
1231 032114  112737  000002  034040         MOVB    #2,T15BS0             ;WRITE RAM "COMMAND"
1232 032122                             25$:
1233 032122  010337  034042                 MOV     R3,T15S2              ;ADDRESS FOR RAM
1234 032126  012704  034030                 MOV     #T15PK2,R4            ;WRITE SUBSYS MEM PACKET
1235 032132  110337  034044                 MOVB    R3,T15S3              ;DATA FOR WRITE (ADDRESS)
1236 032136  010465  000000                 MOV     R4,TSDB(R5)           ;ISSUE COMMAND
1237 032142  004737  016336                 JSR     PC,CHKTSSR            ;WAIT FOR SSR
1238 032146  103407                         BCS     30$                   ;BR, IF NO ERROR
1239 032150  010001                         MOV     R0,R1                 ;ERROR, SAVE TSSR
1243 032152                                 ERRHRD  ERRNO,T15SSR,PKTSSR   ;TSSR NOT CORRECT AFTER WRITE SUB MEM
     032152  104456                                                       TRAP    C$ERHRD
     032154  000623                                                       .WORD   403
     032156  034046                                                       .WORD   T15SSR
     032160  012046                                                       .WORD   PKTSSR
1244 032162                                 ESCAPE  SUB                   ;DON'T CONTINUE IF ERROR ON WRITE
     032162  104410                                                       TRAP    C$ESCAPE
     032164  000122                                                       .WORD   L10050-.
1245 032166                             30$:   CKLOOP                     ;SCOPE LOOP
     032166  104406                                                       TRAP    C$CLP1
1246
1247
1248 032170  005203                         INC     R3                    ;NEXT ADDRESS
1249 032172  020327  010000                 CMP     R3,#10000             ;END OF RAM MEMORY CHECK
1250 032176  001351                         BNE     25$                   ;LOOP TILL ALL RAM WRITTEN
1251 032200  005002                         CLR     R2                    ;CLEAR OUT R2 HIGH BITS
1252 032202  005303                         DEC     R3                    ;SET BACK TO 7777
1253 032204  110337  034044             40$:   MOVB    R3,T15S3              ;GET DATA PATTERN BACK IN SHAPE
1254 032210  010337  034042                 MOV     R3,T15S2              ;ADDRESS FOR RAM READ
1255 032214  112737  000001  034040         MOVB    #1,T15BS0             ;READ RAM COMMAND
1256 032222  010465  000000                 MOV     R4,TSDB(R5)           ;SEND OUT PACKET ADDRESS TO CONTR.
1257 032226  004737  016336                 JSR     PC,CHKTSSR            ;WAIT FOR READY, NON-AMBIGUOUS
1258 032232  103405                         BCS     43$                   ;BR, IF NO PROBLEM
1259 032234  010001                         MOV     R0,R1                 ;SAVE TSSR
1263 032236                                 ERRDF   ERRNO,T15SSR,PKTSSR   ;TSSR NOT CORRECT
     032236  104455                                                       TRAP    C$ERDF
     032240  000624                                                       .WORD   404
     032242  034046                                                       .WORD   T15SSR
     032244  012046                                                       .WORD   PKTSSR
1264 032246                             43$:   CKLOOP                     ;SCOPE LOOP
     032246  104406                                                       TRAP    C$CLP1
1265 032250  013701  033372                 MOV     T15BFR+20,R1          ;GET RAM READ DATA
1266 032254  010302                         MOV     R3,R2                 ;SET UP FOR COMPARE
1267 032256  120102                         CMPB    R1,R2                 ;CHECK WITH DATA WRITTEN
1268 032260  001404                         BEQ     45$                   ;BR IF OK, DATA IN = DATA OUT
1272 032262                                 ERRHRD  ERRNO,T15AM4,EXPBREC  ;WRITTEN DATA NOT = TO READ
     032262  104456                                                       TRAP    C$ERHRD
     032264  000625                                                       .WORD   405
     032266  034325                                                       .WORD   T15AM4
     032270  015502                                                       .WORD   EXPBREC
1273 032272                             45$:   CKLOOP                     ;SCOPE LOOP
     032272  104406                                                       TRAP    C$CLP1
1274 032274  005303                         DEC     R3                    ;DROP DATA COUNTER (PATTERN)
1275 032276  020327  000377                 CMP     R3,#255.              ;AT BOTTOM YET
1276 032302  001340                         BNE     40$                   ;BR, IF MORE TO CHECK
1277 032304                             50$:   CKLOOP                     ;SCOPE LOOP
```

```
032304  104406                                                              TRAP    C$CLP1
032306                          ENDSUB                  ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
032306                                                              L10050:
032306  104403                                                              TRAP    C$ESUB


032310                          BGNSUB                  ;//////////// BEGIN SUBTEST ////////////
032310                                                              T4.2:
032310  104402                                                              TRAP    C$BSUB

                        ;*
                        ;
                        ;TEST 4, SUBTEST 2
                        ;
                        ;
                        ;       THIS SUBTEST WRITES RAM WITH ALL ZEROS
                        ;       THEN WALKS AN ALL ONES WORD DOWN THROUGH MEMORY
                        ;
032312  004737  034432          JSR     PC,T15REST              ;RESTORE PACKET FOR WRITE CHARA
032316  004737  034504          JSR     PC,T15RT2               ;RESTORE PACKET FOR WRT SUB SYS MEM
032322  004737  015774          JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
032326  103405                  BCS     20$                     ;BR IF INIT WAS OK
032330  010001                  MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
032332                          ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
032332  104455                                                              TRAP    C$ERDF
032334  000626                                                              .WORD   406
032336  003652                                                              .WORD   SFIERR
032340  012034                                                              .WORD   SFIMSG
032342                  20$:
032342  012704  033330          MOV     #T15PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
032346  004737  010662          JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
032352  103405                  BCS     25$                     ;BR, IF COMMAND ISSUED OK
032354  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
032356                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
032356  104456                                                              TRAP    C$ERHRD
032360  000627                                                              .WORD   407
032362  005056                                                              .WORD   WRTMSG
032364  012034                                                              .WORD   SFIMSG
032366                  25$:
032366  112737  000001  034041  MOVB    #1,T15BS1               ;SET SIZE OF TRANSFER 1 BYTE
032374  012704  034030          MOV     #T15PK2,R4              ;SET NEW PACKET ADDRESS
032400  012703  000400          MOV     #256.,R3                ;STARTING ADDRESS IN RAM
032404  112737  000002  034040  MOVB    #2,T15BS0               ;WRITE RAM COMMAND
032412  105037  034044          CLRB    T15S3                   ;SET DATA TO 000
032416  010337  034042  30$:    MOV     R3,T15S2                ;ADDRESS TO PACKET DATA AREA
032422  010465  000000          MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
032426  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR
032432  103405                  BCS     33$                     ;BR, IF NO PROBLEM
032434  010001                  MOV     R0,R1                   ;SAVE TSSR
032436                          ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
032436  104456                                                              TRAP    C$ERHRD
032440  000630                                                              .WORD   408
032442  034046                                                              .WORD   T15SSR
032444  012046                                                              .WORD   PKTSSR
032446                  33$:    CKLOOP                          ;SCOPE LOOP
032446  104406                                                              TRAP    C$CLP1
```

```
1326
1327 032450  005203                          INC     R3                      ;NEXT ADDRESS
1328 032452  020327  010000                  CMP     R3,#10000               ;END OF RAM MEMORY CHECK
1329 032456  001357                          BNE     30$                     ;BR, MORE RAM TO GO
1330 032460  005303                  35$:    DEC     R3                      ;SET BACK TO 7777
1331 032462  005002                  40$:    CLR     R2                      ;SET TO ALL ZEROS
1332 032464  112737  000001  034040          MOVB    #1,T15BS0               ;READ RAM COMMAND
1333 032472  010337  034042                  MOV     R3,T15S2                ;ADDRESS TO BE READ TO PACKET DATA
1334 032476  010465  000000                  MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
1335 032502  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1336 032506  103405                          BCS     41$                     ;BR, IF ALL IS WELL
1337 032510  010001                          MOV     R0,R1                   ;SAVE TSSR
1341 032512                                  ERRMRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032512  104456                                                          TRAP    C$ERMRD
     032514  000631                                                          .WORD   409
     032516  034046                                                          .WORD   T15SSR
     032520  012046                                                          .WORD   PKTSSR
1342 032522                          41$:    CKLOOP                          ;SCOPE LOOP
     032522  104406                                                          TRAP    C$CLP1
1343 032524  013701  033372                  MOV     T15BFR+20,R1            ;PICK UP READ DATA
1344 032530  120102                          CMPB    R1,R2                   ;BOTH SHOULD BE 00000000 BINARY
1345 032532  001404                          BEQ     42$                     ;BR, IF DATA IS GOOD
1349 032534                                  ERRMRD  ERRNO,T15AM3,EXPBREC    ;CHARACTERISTICS DATA NOT CORRECT
     032534  104456                                                          TRAP    C$ERMRD
     032536  000632                                                          .WORD   410
     032540  034223                                                          .WORD   T15AM3
     032542  015502                                                          .WORD   EXPBREC
1350 032544                          42$.    CKLOOP                          ;SCOPE LOOPER
     032544  104406                                                          TRAP    C$CLP1
1351 032546  012702  000377                  MOV     #000377,R2              ;SET ALL ONES WORD
1352 032552  112737  000002  034040          MOVB    #2,T15BS0               ;WRITE RAM COMMAND
1353 032560  112737  000377  034044          MOVB    #000377,T15S3           ;ALL ONES PATTERN
1354 032566  010465  000000                  MOV     R4,TSDB(R5)             ;PASS PACKET ADDRESS TO CONTR.
1355 032572  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
1356 032576  103405                          BCS     43$                     ;BR, IF OK (NO ERROR)
1357 032600  010001                          MOV     R0,R1                   ;SAVE TSSR
1361 032602                                  ERRMRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032602  104456                                                          TRAP    C$ERMRD
     032604  000633                                                          .WORD   411
     032606  034046                                                          .WORD   T15SSR
     032610  012046                                                          .WORD   PKTSSR
1362 032612                          43$:    CKLOOP                          ;SCOPE LOOP
     032612  104406                                                          TRAP    C$CLP1
1363 032614  112737  000001  034040          MOVB    #1,T15BS0               ;SET UP FOR RAM READ
1364 032622  010465  000000                  MOV     R4,TSDB(R5)             ;ISSUE RAM READ
1365 032626  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1366 032632  103405                          BCS     44$                     ;BR, IF OK (NO ERROR)
1367 032634  010001                          MOV     R0,R1                   ;SAVE TSSR
1371 032636                                  ERRDF   ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     032636  104455                                                          TRAP    C$ERDF
     032640  000634                                                          .WORD   412
     032642  034046                                                          .WORD   T15SSR
     032644  012046                                                          .WORD   PKTSSR
1372 032646  013701  033372          44$:    MOV     T15BFR+20,R1            ;PICK UP REC'D DATA
1373 032652  120102                          CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
1374 032654  001404                          BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
1378 032656                                  ERRMRD  ERRNO,T15AM2,EXPBREC    ;WRITTEN DATA NOT = TO READ
```

```
                                                                                    TRAP    C$ERHRD
                                                                                    .WORD   413
                                                                                    .WORD   T15AM2
                                                                                    .WORD   EXPBREC
1379 032666                         45$:    CKLOOP                          ;SCOPE LOOP
     032666 104406                                                                  TRAP    C$CLP1
1380 032670 005303                          DEC     R3                      ;DROP RAM ADDRESS POINTER
1381 032672 020327 000377                   CMP     R3,#255.                ;AT START YET
1382 032676 001271                          BNE     40$                     ;BR, IF MORE RAM TO CHECK
1383
1384 032700                                 ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     032700                                                                         L10051:
     032700 104403                                                                  TRAP    C$ESUB
1385
1386 032702                                 BGNSUB                          ;///////////// BEGIN SUBTEST /////////////
     032702                                                                         T4.3:
     032702 104402                                                                  TRAP    C$BSUB
1387                                ;*
1388                                ;
1389                                ;TEST 4, SUBTEST 3
1390                                ;
1391                                ;
1392                                ;       THIS SUBTEST WRITES RAM WITH ALL ONES
1393                                ;       THEN WALKS AN ALL ZEROS WORD DOWN THROUGH MEMORY
1394                                ;
1395 032704 005737 003400                  TST     SKIPT                   ;CHECK RUN MODE
1396 032710 001402                          BEQ     10$                     ;BR, IF NO SKIP
1397 032712 000137 033306                   JMP     50$                     ;SKIP SUBTEST
1398 032716 004737 034432           10$:    JSR     PC,T15REST              ;RESTORE PACKET FOR WRITE CHARA
1399 032722 004737 034504                   JSR     PC,T15RT2               ;RESTORE PACKET FOR WRT SUB SYS MEM
1400 032726 004737 015774                   JSR     PC,SDFINIT              ;DO INITIALIZE ON CONTROLLER
1401 032732 103405                          BCS     20$                     ;BR IF INIT WAS OK
1405 032734 010001                          MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
1406 032736                                 ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
     032736 104455                                                                  TRAP    C$ERDF
     032740 000636                                                                  .WORD   414
     032742 003652                                                                  .WORD   SFIERR
     032744 012034                                                                  .WORD   SFIMSG
1407 032746                         20$:
1408 032746 012704 033330                  MOV     #T15PACKET,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
1409 032752 004737 010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
1410 032756 103405                         BCS     25$                     ;BR, IF COMMAND ISSUED OK
1414 032760 010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1415 032762                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     032762 104456                                                                  TRAP    C$ERHRD
     032764 000637                                                                  .WORD   415
     032766 005056                                                                  .WORD   WRTMSG
     032770 012034                                                                  .WORD   SFIMSG
1416 032772                         25$:
1417 032772 112737 000001 034041           MOVB    #1,T15BS1               ;SET SIZE TO 1 BYTE
1418 033000 012704 034030                  MOV     #T15PK2,R4              ;SET NEW PACKET ADDRESS
1419 033004 012703 000400                  MOV     #256.,R3                ;STARTING ADDRESS IN RAM
1420 033010 112737 000002 034040           MOVB    #2,T15BS0               ;WRITE RAM COMMAND
1421 033016 112737 000377 034044           MOVB    #377,T15S3              ;SET DATA TO 377
1422 033024 010337 034042           30$:    MOV     R3,T15S2                ;ADDRESS TO PACKET DATA AREA
1423 033030 010465 000000                  MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
1424 033034 004737 016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
```

```
1425 033040  103405                        BCS     33$                     ;BR, IF NO PROBLEM
1426 033042  010001                        MOV     R0,R1                   ;SAVE TSSR
1430 033044                                ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     033044  104456                                                                TRAP    C$ERHRD
     033046  000640                                                                .WORD   416
     033050  034046                                                                .WORD   T15SSR
     033052  012046                                                                .WORD   PKTSSR
1431 033054                         33$:    CKLOOP                          ;SCOPE LOOP
     033054  104406                                                                TRAP    C$CLP1
1432
1433
1434 033056  005203                        INC     R3                      ;NEXT ADDRESS
1435 033060  020327  010000                CMP     R3,#10000               ;END OF RAM MEMORY CHECK
1436 033064  001357                        BNE     30$                     ;BR, MORE RAM TO GO
1437 033066  005303                 35$:   DEC     R3                      ;SET BACK TO 7777
1438 033070  112702  000377          40$:   MOVB    #377,R2                 ;SET TO ALL ONES
1439 033074  112737  000001  034040        MOVB    #1,T15BS0               ;READ RAM COMMAND
1440 033102  010337  034042                MOV     R3,T15S2                ;ADDRESS TO BE READ TO PACKET DATA
1441 033106  010465  000000                MOV     R4,TSDB(R5)             ;SEND OUT PACKET ADDRESS
1442 033112  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1443 033116  103405                        BCS     41$                     ;BR, IF ALL IS WELL
1444 033120  010001                        MOV     R0,R1                   ;SAVE TSSR
1448 033122                                ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     033122  104456                                                                TRAP    C$ERHRD
     033124  000641                                                                .WORD   417
     033126  034046                                                                .WORD   T15SSR
     033130  012046                                                                .WORD   PKTSSR
1449 033132                         41$:    CKLOOP                          ;SCOPE LOOP
     033132  104406                                                                TRAP    C$CLP1
1450 033134  013701  033372                MOV     T15BFR+20,R1            ;PICK UP READ DATA
1451 033140  120102                        CMPB    R1,R2                   ;BOTH SHOULD BE 11111111 BINARY
1452 033142  001404                        BEQ     42$                     ;BR, IF DATA IS GOOD
1456 033144                                ERRHRD  ERRNO,T15AM3,EXPBREC    ;CHARACTERISTICS DATA NOT CORRECT
     033144  104456                                                                TRAP    C$ERHRD
     033146  000642                                                                .WORD   418
     033150  034223                                                                .WORD   T15AM3
     033152  015502                                                                .WORD   EXPBREC
1457 033154  012702  000377          42$:   MOV     #000377,R2              ;SET ALL ONES WORD
1458 033160  012737  000002  034040        MOV     #2,T15BS0               ;WRITE RAM COMMAND
1459 033166  112737  000377  034044        MOVB    #000377,T15S3           ;ALL ONES PATTERN
1460 033174  010465  000000                MOV     R4,TSDB(R5)             ;PASS PACKET ADDRESS TO CONTR.
1461 033200  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR
1462 033204  103405                        BCS     43$                     ;BR, IF OK (NO ERROR)
1463 033206  010001                        MOV     R0,R1                   ;SAVE TSSR
1467 033210                                ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
     033210  104456                                                                TRAP    C$ERHRD
     033212  000643                                                                .WORD   419
     033214  034046                                                                .WORD   T15SSR
     033216  012046                                                                .WORD   PKTSSR
1468 033220                         43$:    CKLOOP                          ;SCOPE LOOP
     033220  104406                                                                TRAP    C$CLP1
1469 033222  112737  000001  034040        MOVB    #1,T15BS0               ;SET UP FOR RAM READ
1470 033230  010465  000000                MOV     R4,TSDB(R5)             ;ISSUE RAM READ
1471 033234  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1472 033240  103405                        BCS     44$                     ;BR, IF OK (NO ERROR)
1473 033242  010001                        MOV     R0,R1                   ;SAVE TSSR
1477 033244                                ERRHRD  ERRNO,T15SSR,PKTSSR     ;TSSR NOT CORRECT
```

```
                                                                                    TRAP    C$ERMRD
                                                                                    .WORD   420
                                                                                    .WORD   T15SSR
                                                                                    .WORD   PKTSSR
1478 033254  013701  033372        44$:    MOV     T15BFR+20,R1        ;PICK UP REC'D DATA
1479 033260  120102                CMPB    R1,R2               ;CHECK WITH DATA WRITTEN
1480 033262  001404                BEQ     45$                 ;BR IF OK, DATA IN = DATA OUT
1484 033264                        ERRHRD  ERRNO,T15AM2,EXPBREC    ;WRITTEN DATA NOT = TO READ
     033264  104456                                                                 TRAP    C$ERMRD
     033266  000645                                                                 .WORD   421
     033270  034122                                                                 .WORD   T15AM2
     033272  015502                                                                 .WORD   EXPBREC
1485 033274                        45$:    CKLOOP                  ;SCOPE LOOP
     033274  104406                                                                 TRAP    C$CLP1
1486 033276  005303                DEC     R3                  ;DROP RAM ADDRESS POINTER
1487 033300  020327  000377        CMP     R3,#255.            ;AT START YET
1488 033304  001271                BNE     40$                 ;BR, IF MORE RAM TO CHECK
1489
1490 033306                        50$:
1491 033306                        ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     033306                                                                     L10052:
     033306  104403                                                                 TRAP    C$ESUB
1492
1493 033310  004737  016456        JSR     PC,TSTLOOP          ;DO WE NEED TO ITERATE TEST ?
1494 033314  103002                BCC     63$                 ;BRANCH IF NOT
1495 033316  000137  032004        JMP     T15LOOP             ;EXECUTE AGAIN
1496 033322                        63$:    EXIT    TST         ;ALL DONE THIS TEST
     033322  104432                                                                 TRAP    C$EXIT
     033324  001216                                                                 .WORD   L10047-.
1497                               ;+
1498                               ;LOCAL STORAGE FOR THIS TEST
1499                               ;-
1500
1502         033330                        .=<.+10>&177770
1504 033330                        T15PACKET:                      ;COMMAND PACKET FOR TEST
1505 033330  100204                       .WORD   100204          ;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
1506 033332  033340                       .WORD   T15DATA         ;ADDRESS OF CHARACTERISTICS BLOCK
1507 033334  000000                       .WORD   0
1508 033336  000010                       .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
1509 033340                        T15DATA:                        ;CHARACTERISTICS DATA BLOCK
1510 033340  033352                       .WORD   T15BFR          ;ADDRESS OF MESSAGE BUFFER
1511 033342  000000                       .WORD   0
1512 033344  000400                       .WORD   256.            ;LENGTH OF MESSAGE BUFFER
1513 033346  000000  000000               .WORD   0,0
1514 033352                        T15BFR: .BLKW   150.            ;MESSAGE BUFFER
1515                               ;
1516                               ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
1517                               ;
1519         034030                        .=<.+10>&177770
1521 034030                        T15PK2:
1522 034030  100206                       .WORD   100206          ;WRITE SUB SYS MEM COMMAND, IE AND ACK
1523 034032  034040                       .WORD   T15BF2          ;ADDRESS OF SELECT BLOCK DATA
1524 034034  000000                       .WORD   0
1525 034036  000006                       .WORD   6.              ;SIZE OF DATA PACKET
1526
1527                                      .EVEN
1528 034040                        T15BF2:
```

```
1529 034040     000                 T15BS0: .BYTE   0                       ;BSEL0 AREA
1530 034041     000                 T15BS1: .BYTE   0                       ;BSEL1 AREA
1531 034042  000000                 T15S2:  .WORD   0                       ;SEL 2 AREA
1532 034044  000000                 T15S3:  .WORD   0                       ;DATA AREA
1533
1534
1535
1536
1537                                ;+
1538                                ;LOCAL TEXT MESSAGES FOR TEST
1539                                ;-
1540
1541 034046     127     122     111 T15SSR: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
1542 034122     127     122     111 T15AM2: .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Ones Word Read Back'
1543 034223     127     122     111 T15AM3: .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Zeros Word Read Back'
1544 034325     127     122     111 T15AM4: .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On Address Test'
1545 034413     122     101     115 TST15ID:        .ASCIZ  'RAM Exerciser'
1546                                        .EVEN
1547                                ;+
1548                                ;
1549                                ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
1550                                ;WRITE SUBSYSTEM MEMORY COMMAND
1551                                ;
1552                                ;-
1553
1554 034432                         T15REST:
1555 034432                                 SAVREG                          ;SAVE THE REGISTERS
1556 034436  012701  033330                MOV     #T15PACKET,R1           ;START OF THE PACKET
1557 034442  012721  100204                MOV     #100204,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK. IE
1558 034446  012721  033340                MOV     #T15DATA,(R1)+          ;ADDRESS OF CHARAISTICS DATA BLOCK
1559 034452  005021                        CLR     (R1)+                   ;EXTENDED ADDRESS
1560 034454  012721  000010                MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
1561 034460  012721  033352                MOV     #T15BFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
1562 034464  005021                        CLR     (R1)+
1563 034466  012721  000400                MOV     #256.,(R1)+             ;LENGTH OF MESSAGE BUFFER
1564 034472  005021                        CLR     (R1)+
1565 034474  005011                        CLR     (R1)
1566 034476  005037  033352                CLR     T15BFR                  ;CLEAR 1ST LOC IN MESSAGE BUFFER
1567 034502  000207                        RTS     PC                      ;RETURN
1568
1569
1570 034504                         T15RT2:
1571 034504                                 SAVREG                          ;SAVE THE REGISTERS
1572 034510  012701  034030                MOV     #T15PK2,R1              ;START OF THE PACKET
1573 034514  012721  100206                MOV     #100206,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK. IE
1574 034520  012721  034040                MOV     #T15BF2,(R1)+           ;ADDRESS OF DATA BLOCK
1575 034524  005021                        CLR     (R1)+                   ;EXTENDED ADDRESS
1576 034526  012721  000006                MOV     #6.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
1577 034532  005021                        CLR     (R1)+
1578 034534  005021                        CLR     (R1)+
1579 034536  005011                        CLR     (R1)
1580 034540  000207                        RTS     PC                      ;RETURN
1581 034542                                 ENDTST
     034542                                                         L10047:
     034542  104401                                                       TRAP    C$ETST
```

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


034544                                  BGNTST
034544                                                                          T5:.
034544  012700  036622                    MOV     #TST16ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
034550  004737  016510                    JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
034554  012737  000012  002216            MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
034562                          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
                                ;--
1659 034562                             BGNSUB                          ;////////////// BEGIN SUBTEST //////////
     034562                                                                     T5.1:
     034562  104402                                                                     TRAP    C$BSUB
1660
1661
1662 034564                     5$:
1663                            ;       Write to TSSR register to soft initialize the controller
1664 034564  004737  015774               JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
1665 034570  103405                       BCS     10$                     ;BR IF SOFT INIT OKAY
1666 034572  010001                       MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1667 034574                               ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     034574  104455                                                                     TRAP    C$ERDF
     034576  000764                                                                     .WORD   500
     034600  003652                                                                     .WORD   SFIERR
     034602  012034                                                                     .WORD   SFIMSG
1668                            ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
1669 034604  004737  037770     10$:    JSR     PC,T16REST              ;RESTORE PACKET DEFAULTS
1670 034610  005037  002222               CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1671 034614  012704  040150               MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
1672 034620  004737  010662               JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
1673 034624                               FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
1674 034640  103407                       BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1675 034642  010001                       MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1676 034644                               NEXT.ERRNO
1677 034644                     12$:      ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     034644  104455                                                                     TRAP    C$ERDF
     034646  000765                                                                     .WORD   501
     034650  036672                                                                     .WORD   T16SSR
     034652  012046                                                                     .WORD   PKTSSR
1678 034654  005237  002222               INC     FATFLG                  ;SET FATAL ERROR FLAG
1679 034660                     15$:      CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034660  104406                                                                     TRAP    C$CLP1
1680
1681                            ;       If Extended Features Hardware Switch Clear then:
1682                            ;         (* Verify Extended Features switch can be Inverted to SET *)
1683                            ;         REPEAT FOR TEST PATTERNS IN TSTBLK TABLE
1684 034662  012701  040172               MOV     #T16BFR,R1              ;MESSAGE BUFFER ADDRESS
1685 034666  032761  000200  000012       BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH CLEAR?
1686 034674  001402                       BEQ     20$                     ;BR IF YES
1687 034676  000137  035246               JMP     200$                    ;
1688 034702  012703  002764     20$:      MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
```

```
1689                                        ;       Do Write Subsystem Write Miscellaneous to SET Extended Features.
1690
1691 034706  004737  040130                          JSR     PC,T16SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
1692 034712  012704  040220                          MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
1693 034716  010465  000000                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
1694 034722  004737  016336                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1695 034726                                          FORCERROR       32$                     ;@@DFORCE ERROR IF FORCER=1
1696 034742  103407                                  BCS     40$                     ;BR IF CARRY SET (GOOD RETURN)
1697 034744  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1698 034746                                          NEXT.ERRNO
1699 034746                                  32$:    ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     034746  104455                                                                          TRAP    C$ERDF
     034750  000766                                                                          .WORD   502
     034752  036727                                                                          .WORD   T162SSR
     034754  012046                                                                          .WORD   PKTSSR
1700 034756  005237  002222                          INC     FATFLG                  ;SET FATAL ERROR FLAG
1701 034762                                  40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034762  104406                                                                          TRAP    C$CLP1
1702
1703                                        ;         DO a WRITE CHARACTERISTICS with an extended characteristic word
1704 034764  012737  125252  002312                  MOV     #125252,DATA            ;SETUP TEST DATA FOR EXTENDED WORD
1705 034772  012704  040150                          MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
1706 034776  012764  000020  000006                  MOV     #16.,PKBCNT(R4)         ;STORE MESSAGE PACKET SIZE
1707 035004  013737  002312  040170                  MOV     DATA,T16DATA+10         ;STORE TEST DATA IN EXTENDED WORD
1708 035012  004737  010662                          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
1709 035016                                          FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
1710 035032  103407                                  BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
1711 035034  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1712 035036                                          NEXT.ERRNO
1713 035036                                  42$:    ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     035036  104455                                                                          TRAP    C$ERDF
     035040  000767                                                                          .WORD   503
     035042  036672                                                                          .WORD   T16SSR
     035044  012046                                                                          .WORD   PKTSSR
1714 035046  005237  002222                          INC     FATFLG                  ;SET FATAL ERROR FLAG
1715 035052                                  50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035052  104406                                                                          TRAP    C$CLP1
1716                                        ;         If the TSBA Address Register NOT= Expected  Then Print Error
1717 035054  016501  000000                          MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
1718 035060  012702  040172                          MOV     #T16BFR,R2              ;START OF THE DATA BUFFER
1719 035064  062702  000020                  62$:    ADD     #16.,R2                 ;EXPECTED CONTENTS OF TSBA
1720 035070                                          FORCERROR       72$,NOTSSR              ;@@DFORCE ERROR IF FORCER=1
1721 035100  020102                                  CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1722 035102  001404                                  BEQ     80$                     ;ERROR IF NOT EQUAL
1723 035104                                          NEXT.ERRNO
1724 035104                                  72$:    ERRHRD  ERRNO,T16TSBA,EXPREC    ;PRINT THE ERROR & EXPD/RECV
     035104  104456                                                                          TRAP    C$ERHRD
     035106  000770                                                                          .WORD   504
     035110  037040                                                                          .WORD   T16TSBA
     035112  015474                                                                          .WORD   EXPREC
1725 035114                                  80$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035114  104406                                                                          TRAP    C$CLP1
1726                                        ;         Compare the controller ram to the extended characteristic word
1727                                        ;         If Data word in controller ram NOT= to word sent  Then Print Error
1728 035116  012704  040160                          MOV     #T16DATA,R4             ;GET CHARACTERISTIC DATA ADDRESS
1729 035122  004737  011224                          JSR     PC,CKRAM2               ;DOES RAM DATA EQUAL DATA SENT?
1730 035126                                          FORCERROR       92$                     ;@@DFORCE ERROR IF FORCER=1
```

```
1731 035142  103404                    BCS     100$                    ;BR IF YES
1732 035144                            NEXT.ERRNO
1733 035144                    92$:    ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
     035144  104456                                                            TRAP    C$ERHRD
     035146  000771                                                            .WORD   505
     035150  004745                                                            .WORD   PKTRAM
     035152  015510                                                            .WORD   RAMERR
1734 035154                    100$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035154  104406                                                            TRAP    C$CLP1
1735                           ;         If Message Buffer Data Length NOT= 12.  Then Print Error
1736 035156  012702  040172              MOV     #T16BFR,R2              ;GET MESSAGE BUFFER ADDRESS
1737 035162  016201  000002              MOV     2(R2),R1                ;GET RECV DATA FIELD LENGTH
1738 035166  012702  000014              MOV     #12.,R2                 ;GET EXPD DATA FIELD LENGTH
1739 035172                              FORCERROR       112$,NOTSSR             ;BSDFORCE ERROR IF FORCER=1
1740 035202  020102                      CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1741 035204  001404                      BEQ     120$                    ;ERROR IF NOT EQUAL
1742 035206                              NEXT.ERRNO
1743 035206                    112$:   ERRHRD  ERRNO,T16LEN,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     035206  104456                                                            TRAP    C$ERHRD
     035210  000772                                                            .WORD   506
     035212  037142                                                            .WORD   T16LEN
     035214  015474                                                            .WORD   EXPREC
1744 035216                    120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035216  104406                                                            TRAP    C$CLP1
1745
1746 035220  004737  015774              JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
1747 035224  103405                      BCS     125$                    ;BR IF SOFT INIT OKAY
1748 035226  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1749 035230                              ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     035230  104455                                                            TRAP    C$ERDF
     035232  000772                                                            .WORD   506
     035234  003652                                                            .WORD   SFIERR
     035236  012034                                                            .WORD   SFIMSG
1750 035240                    125$:   CKLOOP                          ;LOOP IF SELECTED
     035240  104406                                                            TRAP    C$CLP1
1751 035242  000137  035426              JMP     300$                    ;
1752
1753                           ;    (* Verify Extended Features switch can be Inverted to CLEAR *)
1754 035246                    200$:
1755                           ;       Do Write Subsystem Write Miscellaneous to CLEAR Extended Features.
1756 035246  004737  040130              JSR     PC,T16SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
1757 035252  012704  040220              MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
1758 035256  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
1759 035262  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1760 035266                              FORCERROR       232$                    ;BSDFORCE ERROR IF FORCER=1
1761 035302  103407                      BCS     240$                    ;BR IF CARRY SET (GOOD RETURN)
1762 035304  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1763 035306                              NEXT.ERRNO
1764 035306                    232$:   ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     035306  104455                                                            TRAP    C$ERDF
     035310  000773                                                            .WORD   507
     035312  036727                                                            .WORD   T162SSR
     035314  012046                                                            .WORD   PKTSSR
1765 035316  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
1766 035322                    240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035322  104406                                                            TRAP    C$CLP1
1767
```

```
1768                                    ;       DO a WRITE CHARACTERISTICS without an extended characteristic word
1769 035324  012704  040150              MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
1770 035330  012764  000016  000006      MOV     #14.,PKBCNT(R4)         ;STORE MESSAGE PACKET SIZE
1771 035336  004737  010662              JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
1772 035342                              FORCERROR       242$                    ;@@DFORCE ERROR IF FORCER=1
1773 035356  103407                      BCS     250$                    ;BR IF CARRY SET (GOOD RETURN)
1774 035360  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1775 035362                              NEXT.ERRNO
1776 035362                      242$:   ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     035362  104455                                                              TRAP    C$ERDF
     035364  000774                                                              .WORD   508
     035366  036672                                                              .WORD   T16SSR
     035370  012046                                                              .WORD   PKTSSR
1777 035372  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
1778 035376                      250$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035376  104406                                                              TRAP    C$CLP1
1779                                    ;       If Message Buffer Data Length NOT= 10.  Then Print Error
1780 035400  013701  040174              MOV     T16BFR+2,R1             ;GET RECV DATA FIELD LENGTH
1781 035404  012702  000012              MOV     #10.,R2                 ;GET EXPD DATA FIELD LENGTH
1782 035410  020102                      CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1783 035412  001404                      BEQ     270$                    ;ERROR IF NOT EQUAL
1784 035414                              NEXT.ERRNO
1785 035414                      262$:   ERRHRD  ERRNO,T16LEN,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     035414  104456                                                              TRAP    C$ERHRD
     035416  000775                                                              .WORD   509
     035420  037142                                                              .WORD   T16LEN
     035422  015474                                                              .WORD   EXPREC
1786 035424                      270$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035424  104406                                                              TRAP    C$CLP1
1787
1788
1789                                    ;       (* Verify Function Reject when Message Buffer 21-19 are non-zero *)
1790                                    ;       Write to TSSR register to soft initialize the controller
1791 035426                      300$:
1792                                    ;       REPEAT FOR MESSAGE BUFFER ADDRESS bits <21:19> FROM 0 TO 7
1793 035426  012737  000001  002312  320$:   MOV     #1,DATA                 ;START AT BITS<21:19>=001
1794                                    ;       DO a WRITE CHARACTERISTICS with a message address bit<21:19> non-zero
1795 035434                      325$:
1796 035434  012704  040150              MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
1797 035440  012764  000016  000006      MOV     #14.,PKBCNT(R4)         ;STORE MESSAGE PACKET SIZE
1798 035446  013700  002312              MOV     DATA,R0                 ;GET TEST DATA
1799         000003                      .REPT   3
1800                                     ASL     R0                      ;SHIFT INTO BITS 21:19
1801                                     .ENDR
1802 035460  010037  040162              MOV     R0,T16DATA+2            ;STORE BUFFER ADDRESS BITS 21:19
1803 035464  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
1804 035470  004737  016250              JSR     PC,WAITF                ;WAIT FOR SSR
1805 035474                              FORCERROR       342$                    ;@@DFORCE ERROR IF FORCER=1
1806 035510  103407                      BCS     350$                    ;BR IF CARRY SET (GOOD RETURN)
1807 035512  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1808 035514                              NEXT.ERRNO
1809 035514                      342$:   ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     035514  104455                                                              TRAP    C$ERDF
     035516  000776                                                              .WORD   510
     035520  036672                                                              .WORD   T16SSR
     035522  012046                                                              .WORD   PKTSSR
1810 035524  005237  002222              INC     FATFLG                  ;SET FATAL ERROR FLAG
```

```
1811 035530                          350$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035530  104406                                                                  TRAP    C$CLP1
1812
1813                                 ;       If TSSR termination code NOT= Function Reject  Then Print Error
1814 035532  016501  000002                  MOV     TSSR(R5),R1             ;GET RECV TSSR
1815 035536  010102                          MOV     R1,R2                   ;COPY RECV TSSR
1816 035540  042702  000016                  BIC     #TERCLS,R2              ;CLEAR TC<2:0> EXPD
1817 035544  052702  000006                  BIS     #TSREJ,R2               ;SET EXPD TC<2:0>= FUNCTION REJECT
1818 035550                                  FORCERROR       352$,NOTSSR             ;@@FORCE ERROR IF FORCER=1
1819 035560  020102                          CMP     R1,R2                   ;EXPD EQUAL RECV?
1820 035562  001404                          BEQ     360$                    ;BR IF YES
1821 035564                                  NEXT.ERRNO
1822 035564                          352$:   ERRHRD  ERRNO,T16REJ,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     035564  104456                                                                  TRAP    C$ERHRD
     035566  000777                                                                  .WORD   511
     035570  037254                                                                  .WORD   T16REJ
     035572  012046                                                                  .WORD   PKTSSR
1823 035574                          360$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035574  104406                                                                  TRAP    C$CLP1
1824 035576                                  FORCEXIT        370$
1825 035606  005237  002312                  INC     DATA                    ;GET NEXT TST PATTERN
1826 035612  023727  002312  000007          CMP     DATA,#7                 ;DONE ALL DATA?
1827 035620  101002                          BHI     370$                    ;BR IF YES
1828 035622  000137  035434                  JMP     325$                    ;DO ANOTHER TEST PATTERN
1829                                 ;       END-REPEAT
1830 035626                          370$:
1831 035626                                  ENDSUB                          ;////////////// END SUBTEST ///////////
     035626                                                                          L10054:
     035626  104403                                                                  TRAP    C$ESUB
1832
1833 035630  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
1834 035634  001402                          BEQ     460$                    ;BRANCH IF NOT
1835 035636  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
1836 035642                          460$:
1837
1838
1839
1840
1841
1842                                         .SBTTL  TEST  5: SUBTEST 2: VERIFY TIMERS A,B
1843
1844                                 ;**
1845                                 ; TEST 5: SUBTEST 2:
1846                                 ;
1847                                 ; SUBTEST DESCRIPTION:
1848                                 ;
1849                                 ;       This subtest verifies that timers A,B can be reset
1850                                 ;       and that Timer A is twice the frequency of Timer B.
1851                                 ;       Timer A has a period of 25 microseconds and Timer B
1852                                 ;       has a period of 50 microseconds. The timers are
1853                                 ;       checked at 1, 28, 53, and 78 micoseconds.
1854                                 ;
1855                                 ; TEST STEPS:
1856                                 ;
1857                                 ;
1858                                 ;       Write to TSSR register to soft initialize the controller
1859                                 ;       Do WRITE CHARACTERISTICS to setup a Message Buffer
```

```
1860                                    ;       (* Verify Timers A,B after RESET TIMER with 0 microsecond delay *)
1861                                    ;       Do a Write Control RESET TIMER with 1 microsecond delay
1862                                    ;       Do a Write Subsystem READ STATUS
1863                                    ;       If Timer A NOT= 0  Then Print Error
1864                                    ;       If Timer B NOT= 0  Then Print Error
1865                                    ;       (* Verify Timers A,B after RESET TIMER with 28 microsecond delay *)
1866                                    ;       Do a Write Control RESET TIMER with 28 microsecond delay
1867                                    ;       If Timer A NOT= 1  Then Print Error
1868                                    ;       If Timer B NOT= 1  Then Print Error
1869                                    ;       Do a Write Control RESET TIMER with 53 microsecond delay
1870                                    ;       If Timer A NOT= 0  Then Print Error
1871                                    ;       If Timer B NOT= 1  Then Print Error
1872                                    ;       Do a Write Control RESET TIMER with 78 microsecond delay
1873                                    ;       If Timer A NOT= 1  Then Print Error
1874                                    ;       If Timer B NOT= 0  Then Print Error
1875                                    ;--
1876 035642                             BGNSUB                          ;////////////// BEGIN SUBTEST //////////
      035642                                                                    T5.2:
      035642  104402                                                           TRAP    C$BSUB
1877                                    ;       Write to TSSR register to soft initialize the controller
1878 035644                             5$:
1879 035644  004737  015774             JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
1880 035650  103405                     BCS     10$                     ;BR IF SOFT INIT OKAY
1881 035652  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1882 035654                             ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      035654  104455                                                           TRAP    C$ERDF
      035656  000777                                                           .WORD   511
      035660  003652                                                           .WORD   SFIERR
      035662  012034                                                           .WORD   SFIMSG
1883                                    ;       Do WRITE CHARACTERISTICS to setup a Message Buffer
1884 035664  004737  037770             10$:    JSR     PC,T16REST              ;RESTORE PACKET DEFAULTS
1885 035670  005037  002222                     CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1886 035674  012704  040150                     MOV     #T16PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
1887 035700  012764  000010  000006             MOV     #8.,PKBCNT(R4)          ;MESSAGE PACKET SIZE NO EXTEND
1888 035706  004737  010662                     JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
1889 035712                                     FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
1890 035726  103407                     BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1891 035730  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1892 035732                             NEXT.ERRNO
1893 035732                             12$:    ERRDF   ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      035732  104455                                                                   TRAP    C$ERDF
      035734  001000                                                                   .WORD   512
      035736  036672                                                                   .WORD   T16SSR
      035740  012046                                                                   .WORD   PKTSSR
1894 035742  005237  002222                     INC     FATFLG                  ;SET FATAL ERROR FLAG
1895 035746                             15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      035746  104406                                                                   TRAP    C$CLP1
1896
1897                                    ;       (* Verify Timers A,B after RESET TIMER with 1 microsecond delay *)
1898                                    ;       Do a Write Control RESET TIMER with 1 microsecond delay
1899 035750  012700  000001                     MOV     #MS.RSD,R0              ;RESET TIMER COMMAND
1900 035754  013701  036612                     MOV     T16D01,R1               ;1 MICROSECOND DELAY
1901 035760  004737  040102                     JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
1902 035764  012704  040220                     MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
1903 035770  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
1904 035774  004737  016336                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1905 036000                                     FORCERROR       32$                     ;@@DFORCE ERROR IF FORCER=1
```

```
1906 036014  103407                    BCS     40$                     ;BR IF CARRY SET (GOOD RETURN)
1907 036016  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1908 036020                            NEXT.ERRNO
1909 036020                    32$:    ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     036020  104455                                                            TRAP    C$ERDF
     036022  001001                                                            .WORD   513
     036024  036727                                                            .WORD   T162SSR
     036026  012046                                                            .WORD   PKTSSR
1910 036030  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
1911 036034                    40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036034  104406                                                            TRAP    C$CLP1
1912                           ;       If Timer A NOT= 0  Then Print Error
1913                           ;       If Timer B NOT= 0  Then Print Error
1914 036036  005002                    CLR     R2                      ;INIT EXPD
1915 036040  042702  000010            BIC     #S2.ATIM,R2             ;TIMER A EXPD=0
1916 036044  042702  000004            BIC     #S2.BTIM,R2             ;TIMER B EXPD=0
1917 036050  012700  040212            MOV     #T168FSTA,R0            ;GET RECV READ STATUS
1918 036054  016001  000002            MOV     2(R0),R1                ;GET RECV BYTE 2
1919 036060  042701  177763            BIC     #^C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
1920 036064                            FORCERROR       72$,NOTSSR      ;@@D
1921 036074  020201                    CMP     R2,R1                   ;EXPD EQUAL RECV?
1922 036076  001404                    BEQ     80$                     ;BR IF YES
1923 036100                            NEXT.ERRNO
1924 036100                    72$:    ERRHRD  ERRNO,T16T01,TIMEXP     ;REPORT ERROR
     036100  104456                                                            TRAP    C$ERHRD
     036102  001002                                                            .WORD   514
     036104  037371                                                            .WORD   T16T01
     036106  015552                                                            .WORD   TIMEXP
1925 036110                    80$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036110  104406                                                            TRAP    C$CLP1
1926
1927                           ;       Do a Write Control RESET TIMER with 28 microsecond delay
1928 036112  012700  000001            MOV     #MS.RSD,R0              ;RESET TIMER COMMAND
1929 036116  013701  036614            MOV     T16D28,R1               ;28 MICROSECOND DELAY
1930 036122  004737  040102            JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
1931 036126  012704  040220            MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
1932 036132  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
1933 036136  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1934 036142                            FORCERROR       112$                    ;@@DFORCE ERHOR IF FORCER=1
1935 036156  103407                    BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
1936 036160  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1937 036162                            NEXT.ERRNO
1938 036162                    112$:   ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     036162  104455                                                            TRAP    C$ERDF
     036164  001003                                                            .WORD   515
     036166  036727                                                            .WORD   T162SSR
     036170  012046                                                            .WORD   PKTSSR
1939 036172  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
1940 036176                    120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036176  104406                                                            TRAP    C$CLP1
1941                           ;       If Timer A NOT= 1  Then Print Error
1942                           ;       If Timer B NOT= 1  Then Print Error
1943 036200  005002                    CLR     R2                      ;INIT EXPD
1944 036202  052702  000010            BIS     #S2.ATIM,R2             ;TIMER A EXPD=1
1945 036206  052702  000004            BIS     #S2.BTIM,R2             ;TIMER B EXPD=1
1946 036212  012700  040212            MOV     #T168FSTA,R0            ;GET RECV READ STATUS
1947 036216  016001  000002            MOV     2(R0),R1                ;GET RECV BYTE 2
```

```
1948 036222  042701  177763          BIC     #^C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
1949 036226                          FORCERROR       1728,NOTSSR     ;BRD
1950 036236  020201                  CMP     R2,R1                   ;EXPD EQUAL RECV?
1951 036240  001404                  BEQ     1808                    ;BR IF YES
1952 036242                          NEXT.ERRNO
1953 036242                  1728:   ERRHRD  ERRNO.T16T28.TIMEXP     ;REPORT ERROR
     036242  104456                                                      TRAP    C$ERHRD
     036244  001004                                                      .WORD   516
     036246  037470                                                      .WORD   T16T28
     036250  015552                                                      .WORD   TIMEXP
1954 036252                  1808:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036252  104406                                                      TRAP    C$CLP1
1955
1956                         ;       Do a Write Control RESET TIMER with 53 microsecond delay
1957 036254  012700  000001          MOV     #MS.RSD,R0              ;RESET TIMER COMMAND
1958 036260  013701  036616          MOV     T16053,R1               ;53 MICROSECOND DELAY
1959 036264  004737  040102          JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
1960 036270  012704  040220          MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
1961 036274  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
1962 036300  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1963 036304                          FORCERROR       2128                    ;BRDFORCE ERROR IF FORCER=1
1964 036320  103407                  BCS     2208                    ;BR IF CARRY SET (GOOD RETURN)
1965 036322  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1966 036324                          NEXT.ERRNO
1967 036324                  2128:   ERRDF   ERRNO,T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     036324  104455                                                      TRAP    C$ERDF
     036326  001005                                                      .WORD   517
     036330  036727                                                      .WORD   T162SSR
     036332  012046                                                      .WORD   PKTSSR
1968 036334  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
1969 036340                  2208:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036340  104406                                                      TRAP    C$CLP1
1970                         ;       If Timer A NOT= 0  Then Print Error
1971                         ;       If Timer B NOT= 1  Then Print Error
1972 036342  005002                  CLR     R2                      ;INIT EXPD
1973 036344  042702  000010          BIC     #S2.ATIM,R2             ;TIMER A EXPD=0
1974 036350  052702  000004          BIS     #S2.BTIM,R2             ;TIMER B EXPD=1
1975 036354  012700  040212          MOV     #T16BFSTA,R0            ;GET RECV READ STATUS
1976 036360  016001  000002          MOV     2(R0),R1                ;GET RECV BYTE 2
1977 036364  042701  177763          BIC     #^C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
1978 036370                          FORCERROR       2728,NOTSSR     ;BRD
1979 036400  020201                  CMP     R2,R1                   ;EXPD EQUAL RECV?
1980 036402  001404                  BEQ     2808                    ;BR IF YES
1981 036404                          NEXT.ERRNO
1982 036404                  2728:   ERRHRD  ERRNO,T16T53,TIMEXP     ;REPORT ERROR
     036404  104456                                                      TRAP    C$ERHRD
     036406  001006                                                      .WORD   518
     036410  037570                                                      .WORD   T16T53
     036412  015552                                                      .WORD   TIMEXP
1983 036414                  2808:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036414  104406                                                      TRAP    C$CLP1
1984                         ;       Do a Write Control RESET TIMER with 78 microsecond delay
1985 036416  012700  000001          MOV     #MS.RSD,R0              ;RESET TIMER COMMAND
1986 036422  013701  036620          MOV     T16078,R1               ;78 MICROSECOND DELAY
1987 036426  004737  040102          JSR     PC,T16WMISC             ;SETUP T16PK2 COMMAND PACKET
1988 036432  012704  040220          MOV     #T16PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
1989 036436  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
```

```
1990 036442  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1991 036446                                 FORCERROR       312$            ;@@FORCE ERROR IF FORCER=1
1992 036462  103407                         BCS     320$                    ;BR IF CARRY SET (GOOD RETURN)
1993 036464  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1994 036466                                 NEXT.ERRNO
1995 036466                         312$:   ERROF   ERRNO.T162SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     036466  104455                                                                TRAP    C$ERDF
     036470  001007                                                                .WORD   519
     036472  036727                                                                .WORD   T162SSR
     036474  012046                                                                .WORD   PKTSSR
1996 036476  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
1997 036502                         320$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036502  104406                                                                TRAP    C$CLP1
1998                                ;       If Timer A NOT= 1  Then Print Error
1999                                ;       If Timer B NOT= 0  Then Print Error
2000 036504  005002                         CLR     R2                      ;INIT EXPD
2001 036506  052702  000010                 BIS     @S2.ATIM,R2             ;TIMER A EXPD=1
2002 036512  042702  000004                 BIC     @S2.BTIM,R2             ;TIMER B EXPD=0
2003 036516  012700  040212                 MOV     @T16BFSTA,R0            ;GET RECV READ STATUS
2004 036522  016001  000002                 MOV     2(R0),R1                ;GET RECV BYTE 2
2005 036526  042701  177763                 BIC     @^C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
2006 036532                                 FORCERROR       372$,NOTSSR     ;@@@
2007 036542  020201                         CMP     R2,R1                   ;EXPD EQUAL RECV?
2008 036544  001404                         BEQ     380$                    ;BR IF YES
2009 036546                                 NEXT.ERRNO
2010 036546                         372$:   ERRHRD  ERRNO,T16T78,TIMEXP     ;REPORT ERROR
     036546  104456                                                                TRAP    C$ERHRD
     036550  001010                                                                .WORD   520
     036552  037670                                                                .WORD   T16T78
     036554  015552                                                                .WORD   TIMEXP
2011 036556                         380$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036556  104406                                                                TRAP    C$CLP1
2012
2013 036560                                 ENDSUB                          ;////////////// END SUBTEST //////////
     036560                                                                        L10055:
     036560  104403                                                                TRAP    C$ESUB
2014
2015 036562  005737  002222                 TST     FATFLG                  ;ANY FATAL ERRORS ?
2016 036566  001402                         BEQ     460$                    ;BRANCH IF NOT
2017 036570  004737  017202                 JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2018 036574  004737  016456         460$:   JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS?
2019 036600  103002                         BCC     465$                    ;BR IF NO
2020 036602  000137  034562                 JMP     T16LOOP                 ;LOOP UNTIL ITERATIONS DONE
2021 036606                         465$:
2022
2023
2024 036606                         EXIT    TST                             ;////////// EXIT TEST //////////
     036606  104432                                                                TRAP    C$EXIT
     036610  001524                                                                .WORD   L10053 .
2025
2026
2027                                ;+
2028                                ;LOCAL STORAGE FOR THIS TEST
2029                                ;-
2030 036612  000001                 T16D01:         .WORD   1               ;1 MICROSECOND DELAY (ACTUALLY .8 MIC)
2031 036614  000040                 T16D28:         .WORD   40              ;28 MICROSECOND DELAY (.8 MICROS PER)
2032 036616  000076                 T16D53:         .WORD   76              ;53 MICROSECOND
```

```
2033 036620  000142                  T16D78:         .WORD   142             ;78 MICROSECOND
2034                                  ;+
2035                                  ;LOCAL TEXT MESSAGES FOR TEST
2036                                  ;
2037
2038 036622      105     170     164  T5T16ID:        .ASCIZ  'Extended Features Switch and Timers A,B
2039 036672      127     122     111  T16SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
2040 036727      127     122     111  T162SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
2041 036773      127     122     111  T163SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
2042 037040      102     165     163  T16TSBA:.ASCIZ  'Bus Address Register (TSBA) Incorrect after Write Characteristics
2043 037142      104     141     164  T16LEN:.ASCIZ   'Data Field Length in Message Buffer Incorrect after Write Characteristics
2044 037254      124     123     123  T16REJ:.ASCIZ   'TSSR Function Reject Not Returned When Non-Existent Buffer Address Specifie
d
2045 037371      124     151     155  T16T01:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 1 microsecond Delay'
2046 037470      124     151     155  T16T28:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 28 microsecond Delay
2047 037570      124     151     155  T16T53:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 53 microsecond Delay'
2048 037670      124     151     155  T16T78:.ASCIZ   'Timer A,B Incorrect after Reset Timer with 78 microsecond Delay
2049                                          .EVEN
2050
2051                                  ;+
2052                                  ; SET DEFAULT PACKET
2053                                  ;-
2054 037770                           T16REST:
2055 037770  012700  040150                    MOV     #T16PACKET,R0           ;PACKET ADDRESS
2056 037774  012720  100004                    MOV     #100004,(R0)+           ;WRITE CHARACTERISTICS WITH ACK
2057 040000  012720  040160                    MOV     #T16DATA,(R0)+          ;ADDRESS OF CHAR DATA BLOCK
2058 040004  005020                            CLR     (R0)+                   ;EXTENDED ADDRESS
2059 040006  012720  000012                    MOV     #10.,(R0)+              ;SIZE OF MESSAGE PACKET
2060 040012  012720  040172                    MOV     #T16BFR,(R0)+           ;MESSAGE BUFFER ADDRESS
2061 040016  005020                            CLR     (R0)+                   ;CLEAR EXTENDED BUFFER ADDRESS
2062 040020  012720  000024                    MOV     #20.,(R0)+              ;LENGTH OF MESSAGE BUFFER
2063 040024  005020                            CLR     (R0)+                   ;CLEAR ESS,ENB,EAI,ERI
2064 040026  005010                            CLR     (R0)                    ;CLEAR EXTENDED FEATURES WORD
2065 040030  005037  040172                    CLR     T16BFR                  ;CLEAR 1ST LOCATION IN MESSAGE BUFFER
2066 040034  000207                            RTS     PC                      ;
2067
2068                                  ;+
2069                                  ; CLEAR MESSAGE BUFFER
2070                                  ;-
2071 040036                           T16CLRBUF:                              ;
2072 040036                                    SAVREG                        ;SAVE R1-R5 UNTIL NEXT RETURN
2073 040042  012701  040172                    MOV     #T16BFR,R1              ;GET MESSAGE BUFFER ADDRESS
2074 040046  012702  000026                    MOV     #T16BEND-T16BFR,R2      ;SIZE OF MESSAGE BUFFER IN BYTES
2075 040052  105021                    10$:    CLRB    (R1)+                   ;CLEAR A BYTE
2076 040054  005302                            DEC     R2                      ;DONE?
2077 040056  003375                            BGT     10$                     ;BR IF NO
2078 040060  000207                            RTS     PC                      ;RETURN
2079
2080                                  ;+
2081                                  ; SETUP T16PK2 PACKET FOR READ STATUS
2082                                  ;-
2083 040062                           T16SRD:
2084 040062  004737  040036                    JSR     PC,T16CLRBUF            ;CLEAR MESSAGE BUFFER
2085 040066  012700  040230                    MOV     #T16DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
2086 040072  112720  000005                    MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
2087 040076  105010                            CLRB    (R0)                    ;CLEAR BSEL1
2088 040100  000207                            RTS     PC                      ;RETURN
2089
```

```
2090
2091                                    ;*
2092                                    ; SETUP T16PK2 PACKET FOR WRITE MISC.
2093                                    ;
2094                                    ; INPUT:
2095                                    ;       R0      CONTAINS WRITE MISC FUNCTION CODE (BSEL1)
2096                                    ;       R1      CONTAINS DELAY (TIMES 800 NS) FOR BSEL2
2097                                    ;-
2098 040102                            T16WMISC:
2099 040102                                    SAVREG                          ;SAVE R1-R5 UNITL NEXT RETURN
2100 040106  004737  040036                    JSR     PC,T16CLRBUF            ;CLEAR MESSAGE BUFFER
2101 040112  012702  040230                    MOV     #T16DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
2102 040116  112722  000010                    MOVB    #PW.WMISC,(R2)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
2103 040122  110022                            MOVB    R0,(R2)+                ;STORE WRITE MISC CODE IN BSEL1
2104 040124  110112                            MOVB    R1,(R2)                 ;STORE DELAY (RESET TIMER) IN BSEL2
2105 040126  000207                            RTS     PC                      ;RETURN
2106                                    ;*
2107                                    ; SETUP T16PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
2108                                    ;-
2109 040130                            T16SEXT:
2110 040130  012700  040230                    MOV     #T16DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
2111 040134  112720  000010                    MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
2112 040140  112710  000200                    MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
2113 040144  000207                            RTS     PC                      ;RETURN
2114
2115
2116
2117
2119         040150                            .=<.+10>&177770
2121                                    ;
2122                                    ;WRITE CHARACTERISTICS COMMAND PACKET
2123                                    ;
2124 040150                            T16PACKET:                              ;COMMAND PACKET FOR TEST
2125 040150  100004                            .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
2126 040152  040160                            .WORD   T16DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
2127 040154  000000                            .WORD   0
2128 040156  000012                            .WORD   10.                     ;MESSAGE PACKET SIZE
2129
2130 040160                            T16DATA:                                ;CHARACTERISTICS DATA BLOCK
2131 040160  040172                            .WORD   T168FR                  ;ADDRESS OF MESSAGE BUFFER
2132 040162  000000                            .WORD   0
2133 040164  000024                            .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
2134 040166  000000                            .WORD   0                       ;ESS,ENB,EAI,ERI
2135 040170  000000                            .WORD   0                       ;EXTENDED FEATURES WORD
2136
2137
2138                                    ;MESSAGE BUFFER
2139
2140 040172                            T168FR:                                 ;BEGIN MESSAGE BUFFER
2141 040172  000000                            .WORD   0                       ;MESSAGE TYPE
2142 040174  000000                            .WORD   0                       ;DATA FIELD LENGTH
2143 040176  000000                            .WORD   0                       ;RBPCR
2144 040200  000000                            .WORD   0                       ;XST0
2145 040202  000000                            .WORD   0                       ;XST1
2146 040204  000000                            .WORD   0                       ;XST2
2147 040206  000000                            .WORD   0                       ;XST3
2148 040210  000000                            .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
```

```
 2149 040212                        T16BFSTA: .BLKB 6.                      ;READ STATUS AND WRITE FIFO BUFFER
 2150 040220                        T16BEND:                                ;END OF MESSAGE BUFFER
 2151                               ;
 2152                               ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
 2153                               ;
 2157 040220                        T16PK2:
 2158 040220  100006                        .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
 2159 040222  040230                        .WORD   T16DT2                  ;LOW ADDRESS OF DATA BLOCK
 2160 040224  000000                        .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
 2161 040226  000012                        .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
 2162
 2163 040230                        T16DT2:                                 ;DATA BLOCK
 2164 040230     000                        .BYTE   0                       ;BSEL0
 2165 040231     000                        .BYTE   0                       ;BSEL1
 2166 040232  000000                        .WORD   0                       ;SEL2
 2167 040234                                .BLKB   64.                     ;WRITE FIFO DATA OUTPUT BUFFER
 2168
 2169
 2170 040334                        ENDTST
      040334                                                          L10053:
      040334  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
                                ;

040336                                  BGNTST
040336                                                                  T6::
040336  012700  046566                  MOV     #TST17ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
040342  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
040346  012737  000012  002216          MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
040354  004737  017274                  JSR     PC,KTOFF                ;SHUT OFF MEMORY MANAGEMENT
040360  005037  003134                  CLR     KTENABLE                ;REALLY SHUT DOWN KT-11
040364                          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
```

```
                                ; -
                                        BGNSUB                          ;////////////// BEGIN SUBTEST //////////
                                                                                T6.1:
040364  104402                                                                    TRAP    C$BSUB
                                ;       Write to TSSR register to soft initialize the controller
                                5$:
040366  004737  015774                  JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
040372  103405                          BCS     10$                     ;BR IF SOFT INIT OKAY
040374  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040376                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
040376  104455                                                                  TRAP    C$ERDF
040400  001130                                                                  .WORD   600
040402  003652                                                                  .WORD   SFIERR
040404  012034                                                                  .WORD   SFIMSG
                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
040406  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
040412  012704  050160                  MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
040416  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
                                        FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
040436  103407                          BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
040440  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040442                                  NEXT.ERRNO
040442                          42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
040442  104455                                                                  TRAP    C$ERDF
040444  001131                                                                  .WORD   601
040446  046605                                                                  .WORD   T17SSR
040450  012046                                                                  .WORD   PKTSSR
040452  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
040456                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
040456  104406                                                                  TRAP    C$CLP1
                                ;       Do a Write Subsystem READ STATUS
040460  004737  047744                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
040464  012704  050330                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
040470  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
040474  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
                                        FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
040514  103407                          BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
040516  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040520                                  NEXT.ERRNO
040520                          62$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
040520  104455                                                                  TRAP    C$ERDF
040522  001132                                                                  .WORD   602
040524  046706                                                                  .WORD   T173SSR
040526  012046                                                                  .WORD   PKTSSR
040530  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
040534                          70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
040534  104406                                                                  TRAP    C$CLP1
                                ;       Set WORDS 0-7  of expd message buffer = to recv since not testing
040536  004737  050126                  JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
040542  012701  046362                  MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
040546  012702  050222                  MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
040552  012221                          MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
040554  011211                          MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
040556  052711  000020                  BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= TRUE
040562  042711  000040                  BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= FALSE
```

```
2273 040566  042711  000200                    BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = FALSE
2274                                    ;       If Input Ready NOT=1 then Print Error
2275                                    ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2276 040572  005000                             CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2277 040574  012701  050202                     MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2278 040600  012702  046342                     MOV     #T17EXP,R2              ;EXPD ADDRESS
2279 040604  012703  000024                     MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2280 040610  004737  011500                     JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2281 040614                                     FORCERROR       82$,NOTSSR      ;BAD
2282 040624  103404                             BCS     90$                     ;BR IF YES
2283 040626                                     NEXT.ERRNO
2284 040626                             82$:    ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
     040626  104456                                                                     TRAP    C$ERHRD
     040630  001133                                                                     .WORD   603
     040632  047125                                                                     .WORD   T171CMP
     040634  012350                                                                     .WORD   MSGSTAT
2285 040636                             90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     040636  104406                                                                     TRAP    C$CLP1
2286
2287 040640                                     ENDSUB                          ;////////////// END SUBTEST //////////
     040640                                                                     L10057:
     040640  104403                                                                     TRAP    C$ESUB
2288
2289 040642  005737  002222                     TST     FATFLG                  ;ANY FATAL ERRORS ?
2290 040646  001402                             BEQ     160$                    ;BRANCH IF NOT
2291 040650  004737  017202                     JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2292 040654                             160$:
2293
2294                                            .SBTTL  TEST  6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST
2295
2296                                    ;++
2297                                    ; TEST 6: SUBTEST 2:
2298                                    ;
2299                                    ; SUBTEST DESCRIPTION:
2300                                    ;
2301                                    ;       This subtest verifies the ability of the FIFO to correctly
2302                                    ;       pass a single data byte from input to output. For each
2303                                    ;       of 256 data values (0-377 octal) the following is done:
2304                                    ;               1. Initial FIFO status is checked
2305                                    ;               2. The Write FIFO function, specifying a count of
2306                                    ;                  one byte to be written is executed.
2307                                    ;               3. Read Status is executed and FIFO status is checked.
2308                                    ;               4. Read FIFO is executed and the data and final status
2309                                    ;                  is checked.
2310                                    ;
2311                                    ; TEST STEPS:
2312                                    ;
2313                                    ; BEGIN
2314                                    ;       Write to TSSR to soft initialize
2315                                    ;       Do a WRITE CHARACTERISTICS to setup a message buffer
2316                                    ;       Do a Write Subsystem READ STATUS
2317                                    ;       If Input Ready  NOT=1   Then Print Error
2318                                    ;       If Output Ready NOT=0   Then Print Error
2319                                    ;       If Data In Miss NOT=0   Then Print Error
2320                                    ;
2321                                    ;  REPEAT FOR DATA FROM 0 TO 377 OCTAL
2322                                    ;    BEGIN
```

```
                                ;       Do a Write Subsystem WRITE NPR to set tape direction out
                                ;       Do a Write Subsystem WRITE FIFO with byte count equal to 1
                                ;       Do a Write Subsystem READ STATUS
                                ;       If Input Ready  NOT=1   Then Print Error
                                ;       If Output Ready NOT=1   Then Print Error
                                ;       If Data In Miss NOT=0   Then Print Error
                                ;       Do Write Subsystem READ FIFO with byte count equal to 1
                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
                                ;       Do a Write Subsystem READ STATUS
                                ;       If Input Ready  NOT=1   Then Print Error
                                ;       If Output Ready NOT=0   Then Print Error
                                ;       If Data In Miss NOT=0   Then Print Error
                                ;    END
                                ;  END
                                ;
040654                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
040654                                                                                  T6.2:
040654  104402                                                                          TRAP    C$BSUB

                                ;       Write to TSSR register to soft initialize the controller
040656                          5$:
040656  004737  015774                  JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
040662  103405                          BCS     10$                     ;BR IF SOFT INIT OKAY
040664  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040666                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
040666  104455                                                                          TRAP    C$ERDF
040670  001133                                                                          .WORD   603
040672  003652                                                                          .WORD   SFIERR
040674  012034                                                                          .WORD   SFIMSG

                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
040676  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
040702  012704  050160                  MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
040706  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
040712                                  FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
040726  103407                          BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
040730  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040732                                  NEXT.ERRNO
040732                          42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
040732  104455                                                                          TRAP    C$ERDF
040734  001134                                                                          .WORD   604
040736  046605                                                                          .WORD   T17SSR
040740  012046                                                                          .WORD   PKTSSR
040742  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
040746                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
040746  104406                                                                          TRAP    C$CLP1

                                ;       Do a Write Subsystem READ STATUS
040750  004737  047744                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
040754  012704  050330                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
040760  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
040764  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
040770                                  FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
041004  103407                          BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
041006  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
041010                                  NEXT.ERRNO
041010                          62$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
041010  104455                                                                          TRAP    C$ERDF
041012  001135                                                                          .WORD   605
```

```
                                                                                .WORD   T173SSR
                                                                                .WORD   PKTSSR
2367 041020 005237 002222               INC     FATFLG                  ;SET FATAL ERROR FLAG
2368 041024                     70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041024 104406                                                              TRAP    C$CLP1
2369                            ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2370 041026 004737 050126               JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2371 041032 012701 046362               MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2372 041036 012702 050222               MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2373 041042 012221                      MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2374 041044 011211                      MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2375 041046 052711 000020               BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= TRUE
2376 041052 042711 000040               BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= FALSE
2377 041056 042711 000200               BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = FALSE
2378                            ;       If Input Ready NOT=1 then Print Error
2379                            ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2380 041062 005000                      CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2381 041064 012701 050202               MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2382 041070 012702 046342               MOV     #T17EXP,R2              ;EXPD ADDRESS
2383 041074 012703 000024               MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2384 041100 004737 011500               JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2385 041104                             FORCERROR       82$,NOTSSR      ;@@D
2386 041114 103404                      BCS     90$                     ;BR IF YES
2387 041116                             NEXT.ERRNO
2388 041116                     82$:    ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
     041116 104456                                                              TRAP    C$ERHRD
     041120 001136                                                              .WORD   606
     041122 047125                                                              .WORD   T171CMP
     041124 012350                                                              .WORD   MSGSTAT
2389 041126                     90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041126 104406                                                              TRAP    C$CLP1
2390
2391                            ;  Repeat for DATA from 0 to 377
2392 041130 012737 000000 002312        MOV     #0,DATA                 ;GET FIRST DATA
2393 041136                     100$:                                   ;REPEAT LABEL
2394                            ;       Do a Write Subsystem WRITE NPR to set tape direction out
2395 041136 012700 000100               MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
2396 041142 004737 050006               JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
2397 041146 012704 050330               MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2398 041152 010465 000000               MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2399 041156 004737 016336               JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2400 041162                             FORCERROR       102$            ;@@DFORCE ERROR IF FORCER=1
2401 041176 103407                      BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
2402 041200 010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2403 041202                             NEXT.ERRNO
2404 041202                     102$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041202 104455                                                              TRAP    C$ERDF
     041204 001137                                                              .WORD   607
     041206 046753                                                              .WORD   T174SSR
     041210 012046                                                              .WORD   PKTSSR
2405 041212 005237 002222               INC     FATFLG                  ;SET FATAL ERROR FLAG
2406 041216                     105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041216 104406                                                              TRAP    C$CLP1
2407                            ;       Do a Write Subsystem WRITE FIFO with byte count equal to 1
2408 041220 012700 000001               MOV     #1,R0                   ;WRITE 1 BYTE
2409 041224 012701 002312               MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
2410 041230 004737 050032               JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
```

```
2411 041234  012704  050330                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2412 041240  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2413 041244  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2414 041250                                FORCERROR       107$                    ;@@DFORCE ERROR IF FORCER=1
2415 041264  103407                        BCS     110$                    ;BR IF CARRY SET (GOOD RETURN)
2416 041266  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2417 041270                                NEXT.ERRNO
2418 041270                        107$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041270  104455                                                                      TRAP    C$ERDF
     041272  001140                                                                      .WORD   608
     041274  047016                                                                      .WORD   T175SSR
     041276  012046                                                                      .WORD   PKTSSR
2419 041300  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
2420 041304                        110$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041304  104406                                                                      TRAP    C$CLP1
2421
2422                               ;       Do a Write Subsystem READ STATUS
2423 041306  004737  047744                JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2424 041312  012704  050330                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2425 041316  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2426 041322  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2427 041326                                FORCERROR       112$                    ;@@DFORCE ERROR IF FORCER=1
2428 041342  103407                        BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
2429 041344  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2430 041346                                NEXT.ERRNO
2431 041346                        112$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041346  104455                                                                      TRAP    C$ERDF
     041350  001141                                                                      .WORD   609
     041352  046706                                                                      .WORD   T173SSR
     041354  012046                                                                      .WORD   PKTSSR
2432 041356  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
2433 041362                        120$:   CKLOOP                          ;LOOP ON ERROR. IF FLAG SET
     041362  104406                                                                      TRAP    C$CLP1
2434                               ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2435 041364  004737  050126                JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2436 041370  012701  046362                MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2437 041374  012702  050222                MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2438 041400  012221                        MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2439 041402  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2440 041404  052711  000020                BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
2441 041410  052711  000040                BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
2442 041414  042711  000200                BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
2443                               ;       If Input Ready NOT=1 then Print Error
2444                               ;       If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
2445 041420  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2446 041422  012701  050202                MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2447 041426  012702  046342                MOV     #T17EXP,R2              ;EXPD ADDRESS
2448 041432  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2449 041436  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2450 041442                                FORCERROR       132$,NOTSSR             ;@@D
2451 041452  103404                        BCS     140$                    ;BR IF YES
2452 041454                                NEXT.ERRNO
2453 041454                        132$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
     041454  104456                                                                      TRAP    C$ERHRD
     041456  001142                                                                      .WORD   610
     041460  047303                                                                      .WORD   T173CMP
     041462  012350                                                                      .WORD   MSGSTAT
```

```
2454 041464                          140$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041464  104406                                                                TRAP    C$CLP1
2455
2456                                 ;      Do Write Subsystem READ FIFO with byte count equal to 1
2457 041466  012700  000001                 MOV     #1,R0                   ;SET READ BYTE COUNT
2458 041472  004737  050066                 JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
2459 041476  012704  050330                 MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2460 041502  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2461 041506  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2462 041512                                 FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
2463 041526  103407                         BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
2464 041530  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2465 041532                                 NEXT.ERRNO
2466 041532                          142$:  ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041532  104455                                                                TRAP    C$ERDF
     041534  001143                                                                .WORD   611
     041536  047062                                                                .WORD   T176SSR
     041540  012046                                                                .WORD   PKTSSR
2467 041542  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
2468 041546                          150$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041546  104406                                                                TRAP    C$CLP1
2469                                 ;      Set WORDS 0-7 of expd message buffer = to recv since not testing
2470 041550  004737  050126                 JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2471 041554  012701  046362                 MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2472 041560  012702  050222                 MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2473 041564  013721  002312                 MOV     DATA,(R1)+              ;SET EXPD WORD #8 = COUNT DATA
2474 041570  011211                         MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV (NOT TESTING)
2475                                 ;      If Data read from FIFO NOT= to Data sent Then Print Error
2476                                 ;      The data is in WORD #8 of the message buffer
2477 041572  005000                         CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2478 041574  012701  050202                 MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2479 041600  012702  046342                 MOV     #T17EXP,R2              ;EXPD ADDRESS
2480 041604  012703  000022                 MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
2481 041610  004737  011500                 JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2482 041614                                 FORCERROR       152$,NOTSSR     ;@@D
2483 041624  103404                         BCS     160$                    ;BR IF YES
2484 041626                                 NEXT.ERRNO
2485 041626                          152$:  ERRHRD  ERRNO,T172CMP,MSGSUB    ;REPORT ERROR
     041626  104456                                                                TRAP    C$ERHPD
     041630  001144                                                                .WORD   612
     041632  047207                                                                .WORD   T172CMP
     041634  013742                                                                .WORD   MSGSUB
2486 041636                          160$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041636  104406                                                                TRAP    C$CLP1
2487
2488                                 ;      Do a Write Subsystem READ STATUS
2489 041640  004737  047744                 JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2490 041644  012704  050330                 MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2491 041650  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2492 041654  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2493 041660                                 FORCERROR       162$                    ;@@DFORCE ERROR IF FORCER=1
2494 041674  103407                         BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
2495 041676  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2496 041700                                 NEXT.ERRNO
2497 041700                          162$:  ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     041700  104455                                                                TRAP    C$ERDF
     041702  001145                                                                .WORD   613
```

```
                041704  046706                                                          .WORD   T173SSR
                041706  012046                                                          .WORD   PKTSSR
2498            041710  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
2499            041714                          170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                041714  104406                                                          TRAP    C$CLP1
2500                                            ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2501            041716  004737  050126                  JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2502            041722  012701  046362                  MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2503            041726  012702  050222                  MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2504            041732  012221                          MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2505            041734  011211                          MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2506            041736  052711  000020                  BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
2507            041742  042711  000040                  BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
2508            041746  042711  000200                  BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
2509                                            ;       If Input Ready NOT=1 then Print Error
2510                                            ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2511            041752  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2512            041754  012701  050202                  MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2513            041760  012702  046342                  MOV     #T17EXP,R2              ;EXPD ADDRESS
2514            041764  012703  000024                  MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2515            041770  004737  011500                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2516            041774                                  FORCERROR       172$,NOTSSR     ;@@D
2517            042004  103404                          BCS     180$                    ;BR IF YES
2518            042006                                  NEXT.ERRNO
2519            042006                          172$:   ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
                042006  104456                                                          TRAP    C$ERHRD
                042010  001146                                                          .WORD   614
                042012  047367                                                          .WORD   T174CMP
                042014  012350                                                          .WORD   MSGSTAT
2520            042016                          180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                042016  104406                                                          TRAP    C$CLP1
2521            042020                                  FORCEXIT        205$            ;@@D
2522            042030  005237  002312                  INC     DATA                    ;GET NEXT TEST DATA
2523            042034  023727  002312  000377          CMP     DATA,#377               ;DONE 0 TO 377?
2524            042042  101002                          BHI     205$                    ;BR IF YES
2525            042044  000137  041136                  JMP     100$                    ;DO ANOTHER TEST PATTERN
2526            042050                          205$:
2527
2528            042050                                  ENDSUB                          ;////////////// END SUBTEST //////////
                042050                                                                          L10060:
                042050  104403                                                                  TRAP    C$ESUB
2529
2530            042052  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
2531            042056  001402                          BEQ     260$                    ;BRANCH IF NOT
2532            042060  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2533            042064                          260$:
2534
2535                                            .SBTTL  TEST  6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST
2536
2537                                            ;++
2538                                            ; TEST 6: SUBTEST 3:
2539                                            ;
2540                                            ; SUBTEST DESCRIPTION:
2541                                            ;
2542                                            ;       This subtest verifies the ability of the FIFO to correctly
2543                                            ;       pass a multiple data bytes from input to output.
2544                                            ;       The following sequence is done with various data patterns
```

```
2545                                    ;       and byte counts from 2 to 64.
2546                                    ;               1. Initial FIFO status is checked
2547                                    ;               2. The Write FIFO function.
2548                                    ;               3. Read Status is executed and FIFO status is checked
2549                                    ;               4. Read FIFO is executed and the data and final status
2550                                    ;                  is checked
2551                                    ;
2552                                    ; TEST STEPS:
2553                                    ;
2554                                    ; BEGIN
2555                                    ;       Write to TSSR to soft initialize
2556                                    ;       Do a WRITE CHARACTERISTICS to setup a message buffer
2557                                    ;       Do a Write Subsystem READ STATUS
2558                                    ;       If Input Ready  NOT=1   Then Print Error
2559                                    ;       If Output Ready NOT=0   Then Print Error
2560                                    ;       If Data In Miss NOT=0   Then Print Error
2561                                    ;       If Last Word    NOT=0   Then Print Error
2562                                    ; REPEAT FOR DATA 0 TO 377, 377 TO 0, FLOATING 1'S,0'S AND ALL 1'S/0 S
2563                                    ; REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
2564                                    ;   BEGIN
2565                                    ;       Do a Write Subsystem WRITE MPR to set tape direction out
2566                                    ;       Do a Write Subsystem WRITE FIFO
2567                                    ;       Do a Write Subsystem READ STATUS
2568                                    ;       If Input Ready  NOT=1   Then Print Error
2569                                    ;       If Output Ready NOT=1   Then Print Error
2570                                    ;       If Data In Miss NOT=0   Then Print Error
2571                                    ;       If Last Word    NOT=0   Then Print Error
2572                                    ;       Do Write Subsystem READ FIFO
2573                                    ;       If Data read from FIFO NOT= to Data sent Then Print Error
2574                                    ;       Do a Write Subsystem READ STATUS
2575                                    ;       If Input Ready  NOT=1   Then Print Error
2576                                    ;       If Output Ready NOT=0   Then Print Error
2577                                    ;       If Data In Miss NOT=0   Then Print Error
2578                                    ;       If Last Word    NOT=0   Then Print Error
2579                                    ;   END
2580                                    ; END
2581                                    ;
2582 042064                            T17:    BGNSUB                          ;////////////// BEGIN SUBTEST //////////
      042064                                                                            T6.3:
      042064  104402                                                                       TRAP    C$BSUB
2583
2584                                    ;       Write to TSSR register to soft initialize the controller
2585 042066                            5$:
2586 042066  004737  015774                    JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
2587 042072  103405                            BCS     10$                     ;BR IF SOFT INIT OKAY
2588 042074  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2589 042076                                    ERROF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      042076  104455                                                                   TRAP    C$EROF
      042100  001146                                                                   .WORD   614
      042102  003652                                                                   .WORD   SFIERR
      042104  012034                                                                   .WORD   SFIMSG
2590                                    ;       Do a WRITE CHARACTERISTICS to setup a message buffer
2591 042106  005037  002222            10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
2592 042112  012704  050160                    MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
2593 042116  004737  010662                    JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
2594 042122                                    FORCERROR       42$                     ;##FORCE ERROR IF FORCER=1
2595 042136  103407                            BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
```

```
2596 042140  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2597 042142                             NEXT.ERRNO
2598 042142                     42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     042142  104455                                                            TRAP    C$ERDF
     042144  001147                                                            .WORD   615
     042146  046605                                                            .WORD   T17SSR
     042150  012046                                                            .WORD   PKTSSR
2599 042152  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
2600 042156                     50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042156  104406                                                            TRAP    C$CLP1
2601                            ;       Do a Write Subsystem READ STATUS
2602 042160  004737  047744            JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2603 042164  012704  050330            MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2604 042170  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2605 042174  004737  016336            JSR     PC,CMKTSSR              ;WAIT FOR SSR TO SET
2606 042200                             FORCERROR       62$             ;BBDFORCE ERROR IF FORCER=1
2607 042214  103407                    BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
2608 042216  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2609 042220                             NEXT.ERRNO
2610 042220                     62$:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042220  104455                                                            TRAP    C$ERDF
     042222  001150                                                            .WORD   616
     042224  046706                                                            .WORD   T173SSR
     042226  012046                                                            .WORD   PKTSSR
2611 042230  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
2612 042234                     70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042234  104406                                                            TRAP    C$CLP1
2613                            ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2614 042236  004737  050126            JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2615 042242  012701  046362            MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2616 042246  012702  050222            MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2617 042252  012221                    MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2618 042254  011211                    MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2619 042256  052711  000020            BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY=  1
2620 042262  042711  000040            BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY=  0
2621 042266  042711  000200            BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
2622 042272  042711  000100            BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
2623                            ;       If Input Ready NOT=1 then Print Error
2624                            ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2625                            ;       If Last Word    NOT=0   Then Print Error
2626 042276  005000                    CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2627 042300  012701  050202            MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2628 042304  012702  046342            MOV     #T17EXP,R2              ;EXPD ADDRESS
2629 042310  012703  000024            MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2630 042314  004737  011500            JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2631 042320                             FORCERROR       82$,NOTSSR      ;BBD
2632 042330  103404                    BCS     90$                     ;BR IF YES
2633 042332                             NEXT.ERRNO
2634 042332                     82$:    ERRHRD  ERRNO,T171CMP,MSGSTAT   ;REPORT ERROR
     042332  104456                                                            TRAP    C$ERHRD
     042334  001151                                                            .WORD   617
     042336  047125                                                            .WORD   T171CMP
     042340  012350                                                            .WORD   MSGSTAT
2635 042342                     90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042342  104406                                                            TRAP    C$CLP1
2636
2637
```

```
2638
2639                                    ;  REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
2640                                    ;  TSTFLAG =1 FOR INCREMENT TEST PATTERN
2641                                    ;           =2 FOR DECREMENT TEST PATTERN
2642                                    ;           =3 FOR TSTBLK TABLE PATTERN
2643 042344  012737  000001  002314            MOV     #1,TSTFLAG              ;TEST PATTERN FLAG
2644 042352                             95$:
2645 042352  012737  000002  002310            MOV     #2,COUNT                ;GET FIRST BYTE COUNT
2646 042360                             100$:
2647                                    ;       Do a Write Subsystem WRITE NPR to set tape direction out
2648 042360  012700  000100                    MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
2649 042364  004737  050006                    JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
2650 042370  012704  050330                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2651 042374  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2652 042400  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2653 042404                                    FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
2654 042420  103407                            BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
2655 042422  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2656 042424                                    NEXT.ERRNO
2657 042424                             102$:  ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042424  104455                                                                    TRAP    C$ERDF
     042426  001152                                                                    .WORD   618
     042430  046753                                                                    .WORD   T174SSR
     042432  012046                                                                    .WORD   PKTSSR
2658 042434  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
2659 042440                             105$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042440  104406                                                                    TRAP    C$CLP1
2660                                    ;       Do a Write Subsystem WRITE FIFO
2661 042442  004737  050106                    JSR     PC,T17CLEXP             ;CLEAR EXPD BUFFER
2662 042446  012701  046464                    MOV     #T17WFDATA,R1           ;EXPD WRITE FIFO DATA BUFFER
2663 042452  013702  002310                    MOV     COUNT,R2                ;TEST PATTERN SIZE
2664 042456  022737  000001  002314            CMP     #1,TSTFLAG              ;INCREMENT PATTERN THIS TIME THRU?
2665 042464  001005                            BNE     115$                    ;BR IF NO
2666 042466  005000                            CLR     R0                      ;INCREMENT TEST PATTERN
2667 042470  110021                     110$:  MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
2668 042472  005200                            INC     R0                      ;SET NEXT PATTERN
2669 042474  005302                            DEC     R2                      ;DONE?
2670 042476  003374                            BGT     110$                    ;BR IF NO
2671 042500  022737  000002  002314     115$:  CMP     #2,TSTFLAG              ;DECREMENT PATTERN THIS TIME THRU?
2672 042506  001006                            BNE     125$                    ;BR IF NO
2673 042510  012700  000377                    MOV     #377,R0                 ;DECREMENT TEST PATTERN
2674 042514  110021                     120$:  MOVB    R0,(R1)+                ;STORE DECREMENT TEST BYTE
2675 042516  005300                            DEC     R0                      ;SET NEXT PATTERN
2676 042520  005302                            DEC     R2                      ;DONE?
2677 042522  003374                            BGT     120$                    ;BR IF NO
2678 042524  022737  000003  002314     125$:  CMP     #3,TSTFLAG              ;TSTBLK PATTERNS THIS TIME THRU?
2679 042532  001005                            BNE     135$                    ;BR IF NO
2680 042534  012700  002752                    MOV     #TSTBLK,R0              ;FLOAT 1'S/0'S ETC. TEST TABLE
2681 042540  112021                     130$:  MOVB    (R0)+,(R1)+             ;STORE A TSTBLK BYTE
2682 042542  005302                            DEC     R2                      ;DONE?
2683 042544  003375                            BGT     130$                    ;BR IF NO
2684 042546                             135$:
2685 042546  013700  002310                    MOV     COUNT,R0                ;FIFO BYTE COUNT
2686 042552  012701  046464                    MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
2687 042556  004737  050032                    JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
2688 042562  012704  050330                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2689 042566  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
```

```
2690 042572  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2691 042576                                FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
2692 042612  103407                        BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
2693 042614  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2694 042616                                NEXT.ERRNO
2695 042616                        142$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042616  104455                                                                TRAP    C$ERDF
     042620  001153                                                                .WORD   619
     042622  047016                                                                .WORD   T175SSR
     042624  012046                                                                .WORD   PKTSSR
2696 042626  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
2697 042632                        150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042632  104406                                                                TRAP    C$CLP1
2698
2699                               ;       Do a Write Subsystem READ STATUS
2700 042634  004737  047744                JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2701 042640  012704  050330                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2702 042644  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2703 042650  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2704 042654                                FORCERROR       157$                    ;@@DFORCE ERROR IF FORCER=1
2705 042670  103407                        BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
2706 042672  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2707 042674                                NEXT.ERRNO
2708 042674                        157$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     042674  104455                                                                TRAP    C$ERDF
     042676  001154                                                                .WORD   620
     042700  046706                                                                .WORD   T173SSR
     042702  012046                                                                .WORD   PKTSSR
2709 042704  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
2710 042710                        160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042710  104406                                                                TRAP    C$CLP1
2711
2712                               ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2713 042712  004737  050126                JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2714 042716  012701  046362                MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2715 042722  012702  050222                MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2716 042726  012221                        MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2717 042730  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2718 042732  052711  000020                BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
2719 042736  052711  000040                BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
2720 042742  042711  000200                BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
2721 042746  042711  000100                BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
2722                               ;       If Input Ready NOT=1 then Print Error
2723                               ;       If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
2724 042752  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2725 042754  012701  050202                MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2726 042760  012702  046342                MOV     #T17EXP,R2              ;EXPD ADDRESS
2727 042764  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2728 042770  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2729 042774                                FORCERROR       162$,NOTSSR             ;@@D
2730 043004  103404                        BCS     170$                    ;BR IF YES
2731 043006                                NEXT.ERRNO
2732 043006                        162$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
     043006  104456                                                                TRAP    C$ERHRD
     043010  001155                                                                .WORD   621
     043012  047303                                                                .WORD   T173CMP
     043014  012350                                                                .WORD   MSGSTAT
```

```
2733 043016                          170$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043016  104406                                                                 TRAP    C$CLP1
2734
2735                                 ;      Do Write Subsystem READ FIFO
2736 043020  013700  002310                 MOV     COUNT,R0                ;SET READ BYTE COUNT
2737 043024  004737  050066                 JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
2738 043030  012704  050330                 MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2739 043034  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2740 043040  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2741 043044                                 FORCERROR       172$                    ;BBDFORCE ERROR IF FORCER=1
2742 043060  103407                         BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
2743 043062  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2744 043064                                 NEXT.ERRNO
2745 043064                          172$:  ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     043064  104455                                                                 TRAP    C$ERDF
     043066  001156                                                                 .WORD   622
     043070  047062                                                                 .WORD   T176SSR
     043072  012046                                                                 .WORD   PKTSSR
2746 043074  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
2747 043100                          180$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043100  104406                                                                 TRAP    C$CLP1
2748
2749                                 ;      If Data read from FIFO NOT= to Data sent Then Print Error
2750 043102  005000                         CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2751 043104  012702  046464                 MOV     #T17WFDATA,R2           ;GET EXPECTED ADDRESS FOR CKMSG2
2752 043110  012701  050222                 MOV     #T17BFSTA,R1            ;GET RECEIVED ADDRESS FOR CKMSG2
2753 043114  013703  002310                 MOV     COUNT,R3                ;NUMBER OF BYTES TO COMPARE
2754 043120  004737  011500                 JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2755 043124                                 FORCERROR       192$,NOTSSR     ;BBD
2756 043134  103406                         BCS     200$                    ;BR IF YES
2757 043136                                 NEXT.ERRNO
2758 043136  013701  002310          192$:  MOV     COUNT,R1                ;GET BYTE COUNT
2759 043142                                 ERRHRD  ERRNO,T175CMP,FIFEXP    ;REPORT ERROR
     043142  104456                                                                 TRAP    C$ERHRD
     043144  001157                                                                 .WORD   623
     043146  047452                                                                 .WORD   T175CMP
     043150  012170                                                                 .WORD   FIFEXP
2760 043152                          200$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043152  104406                                                                 TRAP    C$CLP1
2761
2762                                 ;      Do a Write Subsystem READ STATUS
2763 043154  004737  047744                 JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2764 043160  012704  050330                 MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2765 043164  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2766 043170  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2767 043174                                 FORCERROR       212$                    ;BBDFORCE ERROR IF FORCER=1
2768 043210  103407                         BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
2769 043212  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2770 043214                                 NEXT.ERRNO
2771 043214                          212$:  ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     043214  104455                                                                 TRAP    C$ERDF
     043216  001160                                                                 .WORD   624
     043220  046706                                                                 .WORD   T173SSR
     043222  012046                                                                 .WORD   PKTSSR
2772 043224  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
2773 043230                          220$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043230  104406                                                                 TRAP    C$CLP1
```

```
2774                                    ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
2775 043232  004737  050126             ;       JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2776 043236  012701  046362             ;       MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2777 043242  012702  050222             ;       MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2778 043246  012221                     ;       MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2779 043250  011211                     ;       MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2780 043252  052711  000020             ;       BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
2781 043256  042711  000040             ;       BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
2782 043262  042711  000200             ;       BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
2783 043266  042711  000100             ;       BIC     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=0
2784                                    ;       If Input Ready NOT=1 then Print Error
2785                                    ;       If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2786 043272  005000                     ;       CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2787 043274  012701  050202             ;       MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2788 043300  012702  046342             ;       MOV     #T17EXP,R2              ;EXPD ADDRESS
2789 043304  012703  000024             ;       MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2790 043310  004737  011500             ;       JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2791 043314                             ;       FORCERROR       232$,NOTSSR     ;SSD
2792 043324  103404                     ;       BCS     240$                    ;BR IF YES
2793 043326                             ;       NEXT.ERRNO
2794 043326                             232$:   ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
     043326  104456                                                                     TRAP    C$ERHRD
     043330  001161                                                                     .WORD   625
     043332  047367                                                                     .WORD   T174CMP
     043334  012350                                                                     .WORD   MSGSTAT
2795 043336                             240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043336  104406                                                                     TRAP    C$CLP1
2796 043340                             ;       FORCEXIT        250$            ;SSD
2797 043350  005237  002310             ;       INC     COUNT                   ;GET NEXT BYTE COUNT
2798 043354  023727  002310  000077     ;       CMP     COUNT,#77               ;DONE 0 TO 77
2799 043362  101002                     ;       BHI     250$                    ;BR IF YES
2800 043364  000137  042360             ;       JMP     100$                    ;DO ANOTHER BYTE COUNT
2801 043370  005237  002314             250$:   INC     TSTFLAG                 ;GET NEXT TEST PATTERN CODE
2802 043374  023727  002314  000003     ;       CMP     TSTFLAG,#3              ;DONE INC,DEC,TSTBLK PATTERNS?
2803 043402  101002                     ;       BHI     255$                    ;BR IF YES
2804 043404  000137  042352             ;       JMP     95$                     ;DO ANOTHER TEST PATTERN
2805 043410                             255$:
2806 043410                             ;       ENDSUB                          ;////////////// END SUBTEST //////////
     043410                                                                             L10061:
     043410  104403                                                                             TRAP    C$ESUB
2807
2808 043412  005737  002222             ;       TST     FATFLG                  ;ANY FATAL ERRORS ?
2809 043416  001402                     ;       BEQ     260$                    ;BRANCH IF NOT
2810 043420  004737  017202             ;       JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2811 043424                             260$:
2812
2813
2814
2815                                            .SBTTL  TEST  6: SUBTEST 4: FIFO Verify ILW Status
2816
2817                                    ;++
2818                                    ; TEST 6: SUBTEST 4:
2819                                    ;
2820                                    ; SUBTEST DESCRIPTION:
2821                                    ;
2822                                    ;       This subtest verifies that reading the FIFO when it is
2823                                    ;       empty causes the Last Word (ILW) status to assert.
```

```
                                ;
                                ; TEST STEPS:
                                ;
                                ; BEGIN
                                ;       Write to TSSR to soft initialize
                                ;       Do Write Subsystem READ FIFO with byte count equal to 1
                                ;       Do a Write Subsystem READ STATUS
                                ;       If Input Ready  NOT=1   Then Print Error
                                ;       If Output Ready NOT=0   Then Print Error
                                ;       If Data In Miss NOT=0   Then Print Error
                                ;       If Last Word (ILW) NOT=1 Then Print Error
                                ;    END
                                ;--
043424                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
043424                                                                  T6.4:
043424  104402                                                                  TRAP    C$BSUB

                                ;       Write to TSSR register to soft initialize the controller
043426                          5$:
043426  004737  015774          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
043432  103405                  BCS     10$                     ;BR IF SOFT INIT OKAY
043434  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043436                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
043436  104455                                                                  TRAP    C$ERDF
043440  001161                                                                  .WORD   625
043442  003652                                                                  .WORD   SFIERR
043444  012034                                                                  .WORD   SFIMSG

                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
043446  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
043452  012704  050160                  MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
043456  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
043462                                  FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
043476  103407                          BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
043500  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043502                                  NEXT.ERRNO
043502                          42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
043502  104455                                                                  TRAP    C$ERDF
043504  001162                                                                  .WORD   626
043506  046605                                                                  .WORD   T17SSR
043510  012046                                                                  .WORD   PKTSSR
043512  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
043516                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
043516  104406                                                                  TRAP    C$CLP1

                                ;       Do Write Subsystem READ FIFO with byte count equal to 1
043520  012700  000001          MOV     #1,R0                   ;SET READ BYTE COUNT
043524  004737  050066          JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
043530  012704  050330          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
043534  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
043540  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
043544                          FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
043560  103407                  BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
043562  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043564                          NEXT.ERRNO
043564                  142$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
043564  104455                                                                  TRAP    C$ERDF
043566  001163                                                                  .WORD   627
```

```
                                                                                .WORD   T176SSR
                                                                                .WORD   PKTSSR
2868 043574  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
2869 043600                          150$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043600  104406                                                            TRAP    C$CLP1
2870
2871                                 ;      Do a Write Subsystem READ STATUS
2872 043602  004737  047744                JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
2873 043606  012704  050330                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2874 043612  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2875 043616  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2876 043622                                FORCERROR       162$                    ;@@DFORCE ERROR IF FORCER=1
2877 043636  103407                        BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
2878 043640  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2879 043642                                NEXT.ERRNO
2880 043642                          162$: ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     043642  104455                                                            TRAP    C$ERDF
     043644  001164                                                            .WORD   628
     043646  046706                                                            .WORD   T173SSR
     043650  012046                                                            .WORD   PKTSSR
2881 043652  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
2882 043656                          170$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043656  104406                                                            TRAP    C$CLP1
2883                                 ;      Set WORDS 0-7 of expd message buffer = to recv since not testing
2884 043660  004737  050126                JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
2885 043664  012701  046362                MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
2886 043670  012702  050222                MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
2887 043674  012221                        MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
2888 043676  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
2889 043700  052711  000020                BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
2890 043704  042711  000040                BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
2891 043710  042711  000200                BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
2892 043714  052711  000100                BIS     #S2.ILW,(R1)            ;SET EXP LAST WORD (ILW)=1
2893                                 ;      If Input Ready NOT=1 then Print Error
2894                                 ;      If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2895                                 ;      If Last Word (ILW) NOT=1 Then Print Error
2896 043720  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
2897 043722  012701  050202                MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
2898 043726  012702  046342                MOV     #T17EXP,R2              ;EXPD ADDRESS
2899 043732  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
2900 043736  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
2901 043742                                FORCERROR       172$,NOTSSR             ;@@D
2902 043752  103404                        BCS     180$                    ;BR IF YES
2903 043754                                NEXT.ERRNO
2904 043754                          172$: ERRHRD  ERRNO,T176CMP,MSGSTAT   ;REPORT ERROR
     043754  104456                                                            TRAP    C$ERHRD
     043756  001165                                                            .WORD   629
     043760  047526                                                            .WORD   T176CMP
     043762  012350                                                            .WORD   MSGSTAT
2905 043764                          180$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043764  104406                                                            TRAP    C$CLP1
2906
2907 043766                                ENDSUB                          ;////////////// END SUBTEST //////////
     043766                                                                  L10062:
     043766  104403                                                            TRAP    C$ESUB
2908
2909 043770  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
```

```
2910 043774  001402                    BEQ     260$                    ;BRANCH IF NOT
2911 043776  004737  017202             JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2912 044002                         260$:
2913
2914
2915                                    .SBTTL  TEST  6: SUBTEST 5: FIFO Verify Input Ready
2916
2917                                ;++
2918                                ; TEST 6: SUBTEST 5:
2919                                ;
2920                                ; SUBTEST DESCRIPTION:
2921                                ;
2922                                ;       This subtest verifies that writing 64. bytes into the FIFO
2923                                ;       without reading any out causes the Input Ready status to
2924                                ;       negate. The Subtest then verfiies that writing a 65th byte
2925                                ;       into the FIFO causes the Data In Miss status to assert.
2926                                ;       Next it is verified that the original 64 bytes can be read
2927                                ;       out correctly and that the data has not been corrupted.
2928                                ;
2929                                ; TEST STEPS:
2930                                ;
2931                                ; BEGIN
2932                                ;       Write to TSSR to soft initialize
2933                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
2934                                ;       Do a Write Subsystem WRITE NPR to set tape direction out
2935                                ;       Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
2936                                ;       Do a Write Subsystem READ STATUS
2937                                ;       If Input Ready  NOT=0   Then Print Error
2938                                ;       If Output Ready NOT=1   Then Print Error
2939                                ;       If Data In Miss NOT=0   Then Print Error
2940                                ;       Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
2941                                ;       Do a Write Subsystem READ STATUS
2942                                ;       If Input Ready  NOT=0   Then Print Error
2943                                ;       If Output Ready NOT=1   Then Print Error
2944                                ;       If Data In Miss NOT=1   Then Print Error
2945                                ;       Do Write Subsystem READ FIFO
2946                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
2947                                ;       Do a Write Subsystem READ STATUS
2948                                ;       If Input Ready  NOT=1   Then Print Error
2949                                ;       If Output Ready NOT=0   Then Print Error
2950                                ;       If Data In Miss NOT=1   Then Print Error
2951                                ;  END
2952                                ;--
2953 044002                                BGNSUB                          ;////////////// BEGIN SUBTEST /////////
      044002                                                                      T6.5:
      044002  104402                                                                TRAP    C$BSUB
2954
2955                                ;       Write to TSSR register to soft initialize the controller
2956 044004                         5$:
2957 044004  004737  015774                JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
2958 044010  103405                        BCS     10$                     ;BR IF SOFT INIT OKAY
2959 044012  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2960 044014                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      044014  104455                                                                TRAP    C$ERDF
      044016  001165                                                                .WORD   629
      044020  003652                                                                .WORD   SFIERR
      044022  012034                                                                .WORD   SFIMSG
```

```
2961                                    ;       Do a WRITE CHARACTERISTICS to setup a message buffer
2962 044024  005037  002222             10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
2963 044030  012704  050160                     MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
2964 044034  004737  010662                     JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
2965 044040                                     FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
2966 044054  103407                             BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
2967 044056  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2968 044060                                     NEXT.ERRNO
2969 044060                             42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     044060  104455                                                                  TRAP    C$ERDF
     044062  001166                                                                 .WORD   630
     044064  046605                                                                 .WORD   T17SSR
     044066  012046                                                                 .WORD   PKTSSR
2970 044070  005237  002222                     INC     FATFLG                  ;SET FATAL ERROR FLAG
2971 044074                             50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044074  104406                                                                 TRAP    C$CLP1
2972
2973                                    ;       Do a Write Subsystem WRITE NPR to set tape direction out
2974 044076  012700  000100             100$:   MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
2975 044102  004737  050006                     JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
2976 044106  012704  050330                     MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
2977 044112  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
2978 044116  004737  016336                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2979 044122                                     FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
2980 044136  103407                             BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
2981 044140  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2982 044142                                     NEXT.ERRNO
2983 044142                             102$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044142  104455                                                                  TRAP    C$ERDF
     044144  001167                                                                 .WORD   631
     044146  046753                                                                 .WORD   T174SSR
     044150  012046                                                                 .WORD   PKTSSR
2984 044152  005237  002222                     INC     FATFLG                  ;SET FATAL ERROR FLAG
2985 044156                             105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044156  104406                                                                 TRAP    C$CLP1
2986
2987                                    ;       Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
2988 044160  012737  000100  002310             MOV     #64.,COUNT              ;WRITE 64 BYTES
2989 044166  012701  046464                     MOV     #T17WFDATA,R1           ;EXPD WRITE FIFO DATA BUFFER
2990 044172  012702  000100                     MOV     #64.,R2                 ;TEST PATTERN SIZE
2991 044176  005000                             CLR     R0                      ;INCREMENT TEST PATTERN
2992 044200  110021                     110$:   MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
2993 044202  005200                             INC     R0                      ;SET NEXT PATTERN
2994 044204  005302                             DEC     R2                      ;DONE?
2995 044206  003374                             BGT     110$                    ;BR IF NO
2996 044210  013700  002310                     MOV     COUNT,R0                ;FIFO BYTE COUNT
2997 044214  012701  046464                     MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
2998 044220  004737  050032                     JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
2999 044224  012704  050330                     MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3000 044230  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3001 044234  004737  016336                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3002 044240                                     FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
3003 044254  103407                             BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
3004 044256  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3005 044260                                     NEXT.ERRNO
3006 044260                             142$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044260  104455                                                                  TRAP    C$ERDF
```

```
                044262  001170                                                          .WORD   632
                044264  047016                                                          .WORD   T175SSR
                044266  012046                                                          .WORD   PKTSSR
3007            044270  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
3008            044274                          150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                044274  104406                                                          TRAP    C$CLP1
3009
3010                                            ;       Do a Write Subsystem READ STATUS
3011                                            ;       If Input Ready  NOT=0   Then Print Error
3012                                            ;       If Output Ready NOT=1   Then Print Error
3013                                            ;       If Data In Miss NOT=0   Then Print Error
3014            044276  004737  047744                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
3015            044302  012704  050330                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3016            044306  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3017            044312  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3018            044316                                  FORCERROR       157$            ;@@FORCE ERROR IF FORCER=1
3019            044332  103407                          BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
3020            044334  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3021            044336                                  NEXT.ERRNO
3022            044336                          157$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                044336  104455                                                          TRAP    C$ERDF
                044340  001171                                                          .WORD   633
                044342  046706                                                          .WORD   T173SSR
                044344  012046                                                          .WORD   PKTSSR
3023            044346  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
3024            044352                          160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                044352  104406                                                          TRAP    C$CLP1
3025                                            ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
3026            044354  004737  050126                  JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
3027            044360  012701  046362                  MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
3028            044364  012702  050222                  MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
3029            044370  012221                          MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
3030            044372  011211                          MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
3031            044374  042711  000020                  BIC     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 0
3032            044400  052711  000040                  BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
3033            044404  042711  000200                  BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
3034            044410  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
3035            044412  012701  050202                  MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
3036            044416  012702  046342                  MOV     #T17EXP,R2              ;EXPD ADDRESS
3037            044422  012703  000024                  MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
3038            044426  004737  011500                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3039            044432                                  FORCERROR       162$,NOTSSR     ;@@D
3040            044442  103404                          BCS     170$                    ;BR IF YES
3041            044444                                  NEXT.ERRNO
3042            044444                          162$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
                044444  104456                                                          TRAP    C$ERHRD
                044446  001172                                                          .WORD   634
                044450  047303                                                          .WORD   T173CMP
                044452  012350                                                          .WORD   MSGSTAT
3043            044454                          170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                044454  104406                                                          TRAP    C$CLP1
3044
3045
3046                                            ;       Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
3047            044456  012700  000001                  MOV     #1,R0                   ;FIFO BYTE COUNT
3048            044462  012701  046464                  MOV     #T17WFDATA,R1           ;FIFO WRITE DATA ADDRESS
3049            044466  004737  050032                  JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
```

```
3050 044472  012704  050330                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3051 044476  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3052 044502  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3053 044506                                FORCERROR       172$                    ;@@DFORCE ERROR IF FORCER=1
3054 044522  103407                        BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
3055 044524  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3056 044526                                NEXT.ERRNO
3057 044526                        172$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044526  104455                                                                TRAP    C$ERDF
     044530  001173                                                                .WORD   635
     044532  047016                                                                .WORD   T175SSR
     044534  012046                                                                .WORD   PKTSSR
3058 044536  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
3059 044542                        180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044542  104406                                                                TRAP    C$CLP1
3060
3061                               ;       Do a Write Subsystem READ STATUS
3062                               ;       If Input Ready  NOT=0   Then Print Error
3063                               ;       If Output Ready NOT=1   Then Print Error
3064                               ;       If Data In Miss NOT=1   Then Print Error
3065 044544  004737  047744                JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
3066 044550  012704  050330                MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3067 044554  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3068 044560  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3069 044564                                FORCERROR       187$                    ;@@DFORCE ERROR IF FORCER=1
3070 044600  103407                        BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
3071 044602  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3072 044604                                NEXT.ERRNO
3073 044604                        187$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     044604  104455                                                                TRAP    C$ERDF
     044606  001174                                                                .WORD   636
     044610  046706                                                                .WORD   T173SSR
     044612  012046                                                                .WORD   PKTSSR
3074 044614  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
3075 044620                        190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044620  104406                                                                TRAP    C$CLP1
3076                               ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
3077 044622  004737  050126                JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
3078 044626  012701  046362                MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
3079 044632  012702  050222                MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
3080 044636  012221                        MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
3081 044640  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
3082 044642  042711  000020                BIC     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 0
3083 044646  052711  000040                BIS     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 1
3084 044652  052711  000200                BIS     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 1
3085 044656  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
3086 044660  012701  050202                MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
3087 044664  012702  046342                MOV     #T17EXP,R2              ;EXPD ADDRESS
3088 044670  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
3089 044674  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3090 044700                                FORCERROR       192$,NOTSSR             ;@@D
3091 044710  103404                        BCS     200$                    ;BR IF YES
3092 044712                                NEXT.ERRNO
3093 044712                        192$:   ERRHRD  ERRNO,T173CMP,MSGSTAT   ;REPORT ERROR
     044712  104456                                                                TRAP    C$ERHRD
     044714  001175                                                                .WORD   637
     044716  047303                                                                .WORD   T173CMP
```

```
044720  012350                                                                  .WORD   MSGSTAT
044722                          200$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
044722  104406                                                                  TRAP    C$CLP1
                                ;       Do Write Subsystem READ FIFO
044724  013700  002310                  MOV     COUNT,R0                ;SET READ BYTE COUNT
044730  004737  050066                  JSR     PC,T17RFIF              ;SETUP T17PK2 FOR READ FIFO
044734  012704  050330                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
044740  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
044744  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
044750                                  FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
044764  103407                          BCS     220$                    ;BR IF CARRY SET (GOOD RETUR
044766  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
044770                                  NEXT.ERRNO
044770                          212$:   ERRDF   ERRNO,T176SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
044770  104455                                                                  TRAP    C$ERDF
044772  001176                                                                  .WORD   638
044774  047062                                                                  .WORD   T176SSR
044776  012046                                                                  .WORD   PKTSSR
045000  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
045004                          220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
045004  104406                                                                  TRAP    C$CLP1

                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
045006  005000                          CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
045010  012702  046464                  MOV     #T17WFDATA,R2           ;GET EXPECTED ADDRESS FOR CKMSG2
045014  012701  050222                  MOV     #T17BFSTA,R1            ;GET RECEIVED ADDRESS FOR CKMSG2
045020  013703  002310                  MOV     COUNT,R3                ;NUMBER OF BYTES TO COMPARE
045024  004737  011500                  JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
045030                                  FORCERROR       232$,NOTSSR     ;@@D
045040  103406                          BCS     240$                    ;BR IF YES
045042                                  NEXT.ERRNO
045042  013701  002310          232$:   MOV     COUNT,R1                ;GET BYTE COUNT
045046                                  ERRHRD  ERRNO,T175CMP,FIFEXP    ;REPORT ERROR
045046  104456                                                                  TRAP    C$ERHRD
045050  001177                                                                  .WORD   639
045052  047452                                                                  .WORD   T175CMP
045054  012170                                                                  .WORD   FIFEXP
045056                          240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
045056  104406                                                                  TRAP    C$CLP1

                                ;       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
045060  004737  047744                  JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
045064  012704  050330                  MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
045070  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
045074  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
045100                                  FORCERROR       252$                    ;@@DFORCE ERROR IF FORCER=1
045114  103407                          BCS     260$                    ;BR IF CARRY SET (GOOD RETURN)
045116  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
045120                                  NEXT.ERRNO
045120                          252$:   ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
045120  104455                                                                  TRAP    C$ERDF
045122  001200                                                                  .WORD   640
045124  046706                                                                  .WORD   T173SSR
045126  012046                                                                  .WORD   PKTSSR
```

```
3135 045130  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
3136 045134                             2604$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045134  104406                                                                     TRAP    C$CLP1
3137                                    ;       Set WORDS 0-7 of expd message buffer = to recv since not testing
3138 045136  004737  050126                    JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV
3139 045142  012701  046362                    MOV     @T17EXSTA,R1            ;GET EXPECTED READ STATUS
3140 045146  012702  050222                    MOV     @T17RFSTA,R2            ;GET RECV READ STATUS
3141 045152  012221                            MOV     (R2)+,(R1)+             ;SET EXPD WORD 08 = RECV TEMP
3142 045154  011211                            MOV     (R2),(R1)               ;SET EXPD WORD 09 = RECV TEMP
3143 045156  052711  000020                    BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
3144 045162  042711  000040                    BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
3145 045166  052711  000200                    BIS     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 1
3146 045172  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
3147 045174  012701  050202                    MOV     @T17RFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
3148 045200  012702  046342                    MOV     @T17EXP,R2              ;EXPD ADDRESS
3149 045204  012703  000024                    MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
3150 045210  004737  011500                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3151 045214                                    FORCERROR       2724$,NOTSSR    ;BRD
3152 045224  103404                            BCS     2804                    ;BR IF YES
3153 045226                                    NEXT.ERRNO
3154 045226                             2724$:  ERRHRD  ERRNO,T174CMP,MSGSTAT   ;REPORT ERROR
     045226  104456                                                                     TRAP    C$ERHRD
     045230  001201                                                                     .WORD   641
     045232  047567                                                                     .WORD   T174CMP
     045234  012350                                                                     .WORD   MSGSTAT
3155 045236                             2804$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045236  104406                                                                     TRAP    C$CLP1
3156
3157 045240                                    ENDSUB                          ;////////////// END SUBTEST ////////////
     045240                                                                     L10063:
     045240  104403                                                                     TRAP    C$ESUB
3158
3159 045242  005737  002222                            TST     FATFLG          ;ANY FATAL ERRORS ?
3160 045246  001402                                    BEQ     3004            ;BRANCH IF NOT
3161 045250  004737  017202                            JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
3162 045254                             3004$:
3163
3164
3165
3166                                            .SBTTL  TEST  6: SUBTEST 6: FIFO Verify Reset FIFO Test
3167
3168                                    ;++
3169                                    ; TEST 6: SUBTEST 6:
3170                                    ;
3171                                    ; SUBTEST DESCRIPTION:
3172                                    ;
3173                                    ;       This subtest verifies that the Reset FIFO function within
3174                                    ;       the Write Miscellaneous Control 1 function initializes
3175                                    ;       the FIFO to correct intial status. The following steps
3176                                    ;       are performed:
3177                                    ;               1. Reset an already initialized FIFO and check for
3178                                    ;                  proper status.
3179                                    ;               2. Write a varying number of bytes (1-65.) into the
3180                                    ;                  FIFO and verify that after each block of bytes is
3181                                    ;                  written the FIFO can be be reset to it's initial
3182                                    ;                  state.
3183                                    ;
```

```
                                ; TEST STEPS:
                                ;
                                ; BEGIN
                                ;       Write to TSSR to soft initialize
                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
                                ;       Do a Write Subsystem Write Misc to Reset FIFO
                                ;       Do a Write Subsystem READ STATUS
                                ;       If all Tape Status 2 (ICER,IFMK,IMER) 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,IMER) flip-flop
                                ;         signals NOT=0 Then Print Error
                                ;    END
                                ;
045254                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////
045254                                                                              T6.6:
045254  104402                                                                              TRAP    C$BSUB

                                ;       Write to TSSR register to soft initialize the controller
045256                          5$:
045256  004737  015774          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
045262  103405                  BCS     10$                     ;BR IF SOFT INIT OKAY
045264  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
045266                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
045266  104455                                                                      TRAP    C$ERDF
045270  001201                                                                      .WORD   641
045272  003652                                                                      .WORD   SFIERR
045274  012034                                                                      .WORD   SFIMSG
                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
045276  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
045302  012704  050160                  MOV     #T17PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
045306  004737  010662                  JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
045312                                  FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
045326  103407                          BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
045330  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
045332                                  NEXT.ERRNO
045332                          42$:    ERRDF   ERRNO,T17SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
045332  104455                                                                      TRAP    C$ERDF
045334  001202                                                                      .WORD   642
045336  046605                                                                      .WORD   T17SSR
045340  012046                                                                      .WORD   PKTSSR
045342  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
045346                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
045346  104406                                                                      TRAP    C$CLP1
                                ;       Do a Write Subsystem Write Misc to Reset FIFO
045350  004737  047764          JSR     PC,T17RSFIF             ;SETUP PKT FOR WRITE MISC RESET FIFO
045354  012704  050330          MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
045360  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
045364  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
045370                                  FORCERROR       62$                     ;@@FORCE ERROR IF FORCER=1
045404  103407                          BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
```

```
3230 045406  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3231 045410                            NEXT.ERRNO
3232 045410                    628:    ERRDF   ERRNO,T172SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045410  104455                                                            TRAP    C$ERDF
     045412  001203                                                            .WORD   643
     045414  046642                                                            .WORD   T172SSR
     045416  012046                                                            .WORD   PKTSSR
3233 045420  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
3234 045424                    708:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045424  104406                                                            TRAP    C$CLP1
3235
3236                           ;       Do a Write Subsystem READ STATUS
3237                           ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
3238                           ;         signals NOT=0 Then Print Error
3239 045426  004737  047744            JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
3240 045432  012704  050330            MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3241 045436  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3242 045442  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3243 045446                            FORCERROR       778             ;@@DFORCE ERROR IF FORCER=1
3244 045462  103407                    BCS     808                     ;BR IF CARRY SET (GOOD RETURN)
3245 045464  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3246 045466                            NEXT.ERRNO
3247 045466                    778:    ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045466  104455                                                            TRAP    C$ERDF
     045470  001204                                                            .WORD   644
     045472  046706                                                            .WORD   T173SSR
     045474  012046                                                            .WORD   PKTSSR
3248 045476  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
3249 045502                    808:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045502  104406                                                            TRAP    C$CLP1
3250 045504  004737  050126            JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
3251 045510  012701  046362            MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
3252 045514  012702  050222            MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
3253 045520  011211                    MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
3254 045522  042711  002000            BIC     #S1.ICER,(R1)           ;SET EXPD ICER  =0
3255 045526  042711  001000            BIC     #S1.IFMK,(R1)           ;SET EXPD IFMK  =0
3256 045532  042711  000400            BIC     #S1.IHER,(R1)           ;SET EXPD IHER  =0
3257 045536  016261  000002  000002    MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
3258 045544  005000                    CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
3259 045546  012701  050202            MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
3260 045552  012702  046342            MOV     #T17EXP,R2              ;EXPD ADDRESS
3261 045556  012703  000024            MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
3262 045562  004737  011500            JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3263 045566                            FORCERROR       928,NOTSSR      ;@@D
3264 045576  103404                    BCS     1008                    ;BR IF YES
3265 045600                            NEXT.ERRNO
3266 045600                    928:    ERRHRD  ERRNO,T177CMP,MSGSTAT   ;REPORT ERROR
     045600  104456                                                            TRAP    C$ERHRD
     045602  001205                                                            .WORD   645
     045604  047634                                                            .WORD   T177CMP
     045606  012350                                                            .WORD   MSGSTAT
3267 045610                    1008:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045610  104406                                                            TRAP    C$CLP1
3268
3269                           ;       Do a Write Subsystem WRITE NPR to set tape direction out
3270 045612  012700  000100            MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
3271 045616  004737  050006            JSR     PC,T17SNPR              ;SETUP T17PK2 FOR WRITE NPR
```

```
3272 045622  012704  050330                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3273 045626  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3274 045632  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3275 045636                                    FORCERROR       112$                    ;@@FORCE ERROR IF FORCER=1
3276 045652  103407                           BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
3277 045654  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3278 045656                                    NEXT.ERRNO
3279 045656                            112$:   ERRDF   ERRNO,T174SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045656  104455                                                                    TRAP    C$ERDF
     045660  001206                                                                    .WORD   646
     045662  046753                                                                    .WORD   T174SSR
     045664  012046                                                                    .WORD   PKTSSR
3280 045666  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
3281 045672                            120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     045672  104406                                                                    TRAP    C$CLP1
3282
3283                                   ;       Setup incrementing pattern in FIFO data buffer
3284 045674  012701  046362                    MOV     #T17EXSTA,R1            ;EXPD WRITE FIFO DATA BUFFER
3285 045700  012702  000100                    MOV     #64.,R2                 ;TEST PATTERN SIZE
3286 045704  005000                           CLR     R0                      ;INCREMENT TEST PATTERN
3287 045706  110021                    130$:   MOVB    R0,(R1)+                ;STORE INCREMENT TEST BYTE
3288 045710  005200                           INC     R0                      ;SET NEXT PATTERN
3289 045712  005302                           DEC     R2                      ;DONE?
3290 045714  003374                           BGT     130$                    ;BR IF NO
3291
3292                                   ;  REPEAT FOR BYTE COUNT 1 TO 65.
3293 045716  012737  000001  002310            MOV     #1,COUNT                ;GET FIRST BYTE COUNT
3294                                   ;       Do a Write Subsystem WRITE FIFO with the current byte count
3295 045724                            150$:                                   ;REPEAT LOOP LABEL
3296 045724  013700  002310                    MOV     COUNT,R0                ;FIFO BYTE COUNT
3297 045730  012701  046362                    MOV     #T17EXSTA,R1            ;FIFO WRITE DATA ADDRESS
3298 045734  004737  050032                    JSR     PC,T17WFIF              ;SETUP T17PK2 FOR WRITE FIFO
3299 045740  012704  050330                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3300 045744  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3301 045750  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3302 045754                                    FORCERROR       152$                    ;@@FORCE ERROR IF FORCER=1
3303 045770  103407                           BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
3304 045772  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3305 045774                                    NEXT.ERRNO
3306 045774                            152$:   ERRDF   ERRNO,T175SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     045774  104455                                                                    TRAP    C$ERDF
     045776  001207                                                                    .WORD   647
     046000  047016                                                                    .WORD   T175SSR
     046002  012046                                                                    .WORD   PKTSSR
3307 046004  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
3308 046010                            160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     046010  104406                                                                    TRAP    C$CLP1
3309
3310                                   ;       Do a Write Subsystem Write Misc to Reset FIFO
3311 046012  004737  047764                    JSR     PC,T17RSFIF             ;SETUP PKT FOR WRITE MISC RESET FIFO
3312 046016  012704  050330                    MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3313 046022  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3314 046026  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3315 046032                                    FORCERROR       162$                    ;@@FORCE ERROR IF FORCER=1
3316 046046  103407                           BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
3317 046050  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3318 046052                                    NEXT.ERRNO
```

```
3319 046052                          162$:  ERRDF   ERRNO,T172SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     046052  104455                                                          TRAP    C$ERDF
     046054  001210                                                          .WORD   648
     046056  046642                                                          .WORD   T172SSR
     046060  012046                                                          .WORD   PKTSSR
3320 046062  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
3321 046066                          170$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     046066  104406                                                          TRAP    C$CLP1
3322
3323                                 ;      Do a Write Subsystem READ STATUS
3324                                 ;      If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
3325                                 ;        signals NOT=0 Then Print Error
3326 046070  004737  047744                 JSR     PC,T17SRD               ;SETUP PACKET FOR READ STATUS
3327 046074  012704  050330                 MOV     #T17PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3328 046100  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3329 046104  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3330 046110                                 FORCERROR       177$            ;@@FORCE ERROR IF FORCER=1
3331 046124  103407                         BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
3332 046126  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3333 046130                                 NEXT.ERRNO
3334 046130                          177$:  ERRDF   ERRNO,T173SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     046130  104455                                                          TRAP    C$ERDF
     046132  001211                                                          .WORD   649
     046134  046706                                                          .WORD   T173SSR
     046136  012046                                                          .WORD   PKTSSR
3335 046140  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
3336 046144                          180$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     046144  104406                                                          TRAP    C$CLP1
3337 046146  004737  050126                 JSR     PC,T17SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
3338 046152  012701  046362                 MOV     #T17EXSTA,R1            ;GET EXPECTED READ STATUS
3339 046156  012702  050222                 MOV     #T17BFSTA,R2            ;GET RECV READ STATUS
3340 046162  011211                         MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
3341 046164  042711  002000                 BIC     #S1.ICER,(R1)           ;SET EXPD ICER  =0
3342 046170  042711  001000                 BIC     #S1.IFMK,(R1)           ;SET EXPD IFMK  =0
3343 046174  042711  000400                 BIC     #S1.IHER,(R1)           ;SET EXPD IHER  =0
3344 046200  016261  000002  000002         MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
3345 046206  005000                         CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
3346 046210  012701  050202                 MOV     #T17BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
3347 046214  012702  046342                 MOV     #T17EXP,R2              ;EXPD ADDRESS
3348 046220  012703  000024                 MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
3349 046224  004737  011500                 JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
3350 046230                                 FORCERROR       192$,NOTSSR     ;@@D
3351 046240  103404                         BCS     200$                    ;BR IF YES
3352 046242                                 NEXT.ERRNO
3353 046242                          192$:  ERRHRD  ERRNO,T177CMP,MSGSTAT   ;REPORT ERROR
     046242  104456                                                          TRAP    C$ERHRD
     046244  001212                                                          .WORD   650
     046246  047634                                                          .WORD   T177CMP
     046250  012350                                                          .WORD   MSGSTAT
3354 046252                          200$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     046252  104406                                                          TRAP    C$CLP1
3355
3356
3357 046254                          250$:
3358 046254                                 FORCEXIT        260$
3359 046264  005237  002310                 INC     COUNT                   ;GET NEXT BYTE COUNT
3360 046270  023727  002310  000101         CMP     COUNT,#65.              ;DONE ALL BYTES?
```

```
046276  101002                          BHI     260$                    ;BR IF YES
046300  000137  045724                  JMP     150$                    ;DO ANOTHER BYTE COUNT
046304                          260$:

046304                                  ENDSUB                          ;///////////// END SUBTEST //////////
046304                                                                          L10064:
046304  104403                                                                          TRAP    C$ESUB

046306  005737  002222                  TST     FATFLG                  ;ANY FATAL ERRORS ?
046312  001402                          BEQ     300$                    ;BRANCH IF NOT
046314  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
046320  004737  016456          300$:   JSR     PC,TSTLOOP              ;DO ITERATIONS?
046324  103002                          BCC     305$                    ;BR IF NO
046326  000137  040364                  JMP     T17LOOP                 ;LOOP UNTIL ITERATIONS DONE
046332                          305$:

046332                                  EXIT    TST                     ;/////////// EXIT TEST //////////
046332  104432                                                                          TRAP    C$EXIT
046334  002112                                                                          .WORD   L10056 .



                                ;+
                                ;LOCAL STORAGE FOR THIS TEST
                                ;



046336                          T17MSK:                                 ;MASK OF UNTESTED BITS IN READ STATUS BYTES
                                                                        ;UNTESTED BITS ARE SET TO 1
046336     377                          .BYTE   ^C<000>                 ;BYTE 0 MASK
046337     037                          .BYTE   ^C<340>                 ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
046340     360                          .BYTE   ^C<017>                 ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
046341     000                          .BYTE   0                       ;MAKE IT EVEN

046342                          T17EXP:                                 ;BEGIN EXPECTED DATA BUFFER
046342  000000                          .WORD   0                       ;MESSAGE TYPE
046344  000000                          .WORD   0                       ;DATA FIELD LENGTH
046346  000000                          .WORD   0                       ;RBPCR
046350  000000                          .WORD   0                       ;XST0
046352  000000                          .WORD   0                       ;XST1
046354  000000                          .WORD   0                       ;XST2
046356  000000                          .WORD   0                       ;XST3
046360  000000                          .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
046362                          T17EXSTA: .BLKB 66.                     ;EXPECTED READ STATUS AND WRITE FIFO DATA
046464                          T17EXEND:                                       ;END EXPECTED DATA BUFFER

046464                          T17WFDATA: .BLKB 66.                    ;WRITE FIFO EXPECTED DATA BUFFER

                                ;+
                                ;LOCAL TEXT MESSAGES FOR TEST
                                ;-

046566     106     111     106  TST17ID:        .ASCIZ  'FIFO Exerciser'
046605     127     122     111  T17SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
046642     127     122     111  T172SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
046706     127     122     111  T173SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
046753     127     122     111  T174SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
047016     127     122     111  T175SSR:.ASCIZ  'WRITE SUBSYSTEM (Write FIFO) Failed
```

```
3414 047062  127  122  111  T176SSR:.ASCIZ  'WRITE SUBSYSTEM (Read FIFO) Failed'
3415 047125  106  111  106  T171CMP:.ASCIZ  'FIFO Status in WORD #9 Incorrect after Initialize
3416 047207  122  145  141  T172CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
3417 047303  105  111  106  T173CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after WRITE FIFO'
3418 047367  106  111  106  T174CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after READ FIFO'
3419 047452  122  145  141  T175CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO Data'
3420 047526  106  111  106  T176CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after READ FIFO from an Empty FIFO'
3421 047634  106  111  106  T177CMP:.ASCIZ  'FIFO Status (In WORD #9) Incorrect after RESET FIFO'
3422                          .EVEN
3423
3424                          ;+
3425                          ; CLEAR MESSAGE BUFFER
3426                          ;
3427 047720                   T17CLRBUF:                                    ;
3428 047720                           SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
3429 047724  012701  050202           MOV     #T17BFR,R1                    ;GET MESSAGE BUFFER ADDRESS
3430 047730  012702  000120           MOV     #T17BEND-T17BFR,R2            ;SIZE OF MESSAGE BUFFER IN BYTES
3431 047734  105021           10$:    CLRB    (R1)+                         ;CLEAR A BYTE
3432 047736  005302                   DEC     R2                            ;DONE?
3433 047740  003375                   BGT     10$                           ;BR IF NO
3434 047742  000207                   RTS     PC                            ;RETURN
3435
3436                          ;+
3437                          ; SETUP T17PK2 PACKET FOR READ STATUS
3438                          ;
3439 047744                   T17SRD:
3440 047744  004737  047720           JSR     PC,T17CLRBUF                  ;CLEAR MESSAGE BUFFER
3441 047750  012700  050340           MOV     #T17DT2,R0                    ;WRITE SUBSYSTEM DATA BUFFER
3442 047754  112720  000005           MOVB    #PW.RDSTATUS,(R0)+            ;STORE READ STATUS COMMAND IN BSEL0
3443 047760  105010                   CLRB    (R0)                          ;CLEAR BSEL1
3444 047762  000207                   RTS     PC                            ;RETURN
3445
3446                          ;+
3447                          ; SETUP T17PK2 PACKET FOR WRITE MISC RESET FIFO
3448                          ;-
3449 047764                   T17RSFIF:
3450 047764  004737  047720           JSR     PC,T17CLRBUF                  ;CLEAR MESSAGE BUFFER
3451 047770  012700  050340           MOV     #T17DT2,R0                    ;WRITE SUBSYSTEM DATA BUFFER
3452 047774  112720  000010           MOVB    #PW.WMISC,(R0)+               ;STORE WRITE MISCELLANEOUS IN BSEL0
3453 050000  112710  000030           MOVB    #MS.RSFIF!MS.RSTAP,(R0)       ;STORE BSEL1 CLEAR FIFO CODES
3454 050004  000207                   RTS     PC                            ;RETURN
3455
3456                          ;+
3457                          ; SETUP T17PK2 PACKET FOR WRITE NPR
3458                          ;
3459                          ; INPUT:
3460                          ;       R0 CONTAINS BSEL1 NPR DATA
3461                          ;
3462                          ;       SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
3463                          ;-
3464 050006                   T17SNPR:
3465 050006  004737  047720           JSR     PC,T17CLRBUF                  ;CLEAR MESSAGE BUFFER
3466 050012  012701  050340           MOV     #T17DT2,R1                    ;WRITE SUBSYSTEM DATA BUFFER
3467 050016  112721  000011           MOVB    #PW.WNPR,(R1)+                ;STORE WRITE NPR IN BSEL0
3468 050022  052700  000020           BIS     #NP.WRP,R0                    ;DON'T WRITE WRONG PARITY
3469 050026  110011                   MOVB    R0,(R1)                       ;STORE NPR DATA IN BSEL1
3470 050030  000207                   RTS     PC                            ;RETURN
```

```
3471
3472                                    ;+
3473                                    ; SETUP T17PK2 PACKET FOR WRITE FIFO
3474                                    ;
3475                                    ; INPUT:
3476                                    ;       R0 CONTAINS BYTE COUNT
3477                                    ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
3478                                    ;-
3479 050032                            T17W IF:
3480 050032                                    SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
3481 050036  004737  047720                    JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
3482 050042  012702  050340                    MOV     #T17DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
3483 050046  112722  000004                    MOVB    #PW.WFIFO,(R2)+         ;STORE WRITE FIFO IN BSEL0
3484 050052  110022                            MOVB    R0,(R2)+                ;STORE BYTE COUNT IN BSEL1
3485 050054  005022                            CLR     (R2)+                   ;CLEAR SEL2 (UNUSED)
3486 050056  112122                    10$:    MOVB    (R1)+,(R2)+             ;STORE DATA PATTERN BYTE
3487 050060  005300                            DEC     R0                      ;DONE ALL BYTES?
3488 050062  003375                            BGT     10$                     ;BR IF NO
3489 050064  000207                            RTS     PC                      ;RETURN
3490
3491                                    ;+
3492                                    ; SETUP T17PK2 PACKET FOR READ FIFO
3493                                    ;
3494                                    ; INPUT:
3495                                    ;       R0 CONTAINS SEL2 BYTE COUNT
3496                                    ;-
3497 050066                            T17RFIF:
3498 050066  004737  047720                    JSR     PC,T17CLRBUF            ;CLEAR MESSAGE BUFFER
3499 050072  012701  050340                    MOV     #T17DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
3500 050076  112721  000003                    MOVB    #PW.RFIFO,(R1)+         ;STORE READ FIFO IN BSEL0
3501 050102  110021                            MOVB    R0,(R1)+                ;STORE BYTE COUNT IN BSEL1
3502 050104  000207                            RTS     PC                      ;RETURN
3503                                    ;+
3504                                    ; CLEAR EXPECTED DATA MESSAGE BUFFER
3505                                    ;-
3506 050106                            T17CLEXP:
3507 050106  012701  046342                    MOV     #T17EXP,R1              ;GET EXPD ADDRESS
3508 050112  012700  000122                    MOV     #T17EXEND-T17EXP,R0     ;GET EXPD SIZE
3509 050116  105021                    10$:    CLRB    (R1)+                   ;CLEAR A BYTE
3510 050120  005300                            DEC     R0                      ;DONE?
3511 050122  003375                            BGT     10$                     ;BR IF NO
3512 050124  000207                            RTS     PC                      ;RETURN
3513
3514                                    ;+
3515                                    ;Set WORDS 0-7 of expd message buffer = to recv since not testing
3516                                    ;-
3517 050126                            T17SETEXP:
3518 050126  012702  046342                    MOV     #T17EXP,R2              ;GET EXPD
3519 050132  012703  050202                    MOV     #T17BFR,R3              ;GET READ STATUS RECV BUFFER
3520 050136  012700  000010                    MOV     #8.,R0                  ;SET WORDS 0-7 EXP=RECV
3521 050142  012322                    5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
3522 050144  005300                            DEC     R0                      ;DONE WORDS 0-7 WORDS?
3523 050146  003375                            BGT     5$                      ;BR IF NO
3524 050150  000207                            RTS     PC                      ;RETURN
3525
3527         050160                            .=<.+10>&177770
3529                                    ;
```

```
3530                                    ;WRITE CHARACTERISTICS COMMAND PACKET
3531                                    ;
3532 050160                             T17PACKET:                              ;COMMAND PACKET FOR TEST
3533 050160  100004                             .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
3534 050162  050170                             .WORD   T17DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
3535 050164  000000                             .WORD   0
3536 050166  000012                             .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
3537
3538 050170                             T17DATA:                                ;CHARACTERISTICS DATA BLOCK
3539 050170  050202                             .WORD   T17BFR                  ;ADDRESS OF MESSAGE BUFFER
3540 050172  000000                             .WORD   0
3541 050174  000024                             .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
3542 050176  000000                             .WORD   0                       ;ESS,ENB,EAI,ERI
3543 050200  000000                             .WORD   0                       ;EXTENDED FEATURES UNIT NO. ETC.
3544
3545
3546                                    ;MESSAGE BUFFER FOR ALL TEST 6 COMMANDS
3547
3548 050202                             T17BFR:                                 ;BEGIN MESSAGE BUFFER
3549 050202  000000                             .WORD   0                       ;MESSAGE TYPE
3550 050204  000000                             .WORD   0                       ;DATA FIELD LENGTH
3551 050206  000000                             .WORD   0                       ;RBPCR
3552 050210  000000                             .WORD   0                       ;XST0
3553 050212  000000                             .WORD   0                       ;XST1
3554 050214  000000                             .WORD   0                       ;XST2
3555 050216  000000                             .WORD   0                       ;XST3
3556 050220  000000                             .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
3557 050222                             T17BFSTA: .BLKB 64.                     ;READ STATUS AND WRITE FIFO BUFFER
3558 050322                             T17BEND:                                ;END OF MESSAGE BUFFER
3559                                    ;
3560                                    ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
3561                                    ;
3563         050330                             .=<.+10>&177770
3565 050330                             T17PK2:
3566 050330  100006                             .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
3567 050332  050340                             .WORD   T17DT2                  ;LOW ADDRESS OF DATA BLOCK
3568 050334  000000                             .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
3569 050336  000012                             .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
3570
3571 050340                             T17DT2:                                 ;DATA BLOCK
3572 050340     000                             .BYTE   0                       ;BSEL0
3573 050341     000                             .BYTE   0                       ;BSEL1
3574 050342  000000                             .WORD   0                       ;SEL2
3575 050344                                     .BLKB   66.                     ;WRITE FIFO DATA OUTPUT BUFFER
3576
3577 050446                                    ENDTST
     050446                                                                  L10056:
     050446  104401                                                                TRAP    C$ETST
3578
3579                                            .SBTTL  TEST  7: STATIC TRANSPORT BUS INTERFACE TEST
3580                                    ;+
3581                                    ; TEST DESCRIPTION:
3582                                    ;
3583                                    ;
3584                                    ; TEST STEPS:
3585                                    ;
3586                                    ;    REPEAT FOR LOOPCNT
```

```
3587                                    ;       BEGIN
3588                                    ;       Write to TSSR register to soft initialize the controller
3589                                    ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
3590                                    ;       If Extended Features Hardware Switch Clear then:
3591                                    ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
3592                                    ;       Do WRITE CHARACTERISTICS to select reserved unit 7
3593                                    ;       Do a Write Subsystem READ STATUS
3594                                    ;       If any transport interface signals are asserted then Print Error
3595                                    ;       END
3596                                    ;
3597
3598
3599 050450                                    BGNTST
     050450                                                                          T7::
3604 050450  012700  051156                    MOV     #TST18ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
3605 050454  004737  016510                    JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
3606 050460  012737  000012  002216           MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
3607 050466                            T18LOOP:
3608                                    ;       Write to TSSR register to soft initialize the controller
3609 050466                            5$:
3610 050466  004737  015774                    JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
3611 050472  103405                            BCS     10$                     ;BR IF SOFT INIT OKAY
3612 050474  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3613 050476                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     050476  104455                                                                  TRAP    C$ERDF
     050500  001274                                                                  .WORD   700
     050502  003652                                                                  .WORD   SFIERR
     050504  012034                                                                  .WORD   SFIMSG
3614
3615                                    ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
3616 050506  005037  002222            10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
3617 050512  012704  051640                    MOV     #T18PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3618 050516  004737  010662                    JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
3619 050522                                    FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
3620 050536  103407                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
3621 050540  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3622 050542                                    NEXT.ERRNO
3623 050542                            12$:    ERRDF   ERRNO,T18SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     050542  104455                                                                  TRAP    C$ERDF
     050544  001275                                                                  .WORD   701
     050546  051215                                                                  .WORD   T18SSR
     050550  012046                                                                  .WORD   PKTSSR
3624 050552  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
3625 050556                            15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     050556  104406                                                                  TRAP    C$CLP1
3526
3627                                    ;       If Extended Features Hardware Switch Clear then:
3628                                    ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
3629 050560  012701  051662                    MOV     #T18BFR,R1              ;MESSAGE BUFFER ADDRESS
3630 050564  032761  000200  000012           BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
3631 050572  001026                            BNE     30$                     ;BR IF YES
3632 050574  004737  051506                    JSR     PC,T18SMISC             ;SETUP PACKET FOR WRITE MISCELLANEOUS
3633 050600  012704  051710                    MOV     #T18PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3634 050604  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3635 050610  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3636 050614                                    FORCERROR       22$                     ;@@DFORCE ERROR IF FORCER=1
3637 050630  103407                            BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
```

```
3638 050632  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3639 050634                            NEXT.ERRNO
3640 050634                    22$:    ERRDF   ERRNO,T182SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     050634  104455                                                            TRAP    C$ERDF
     050636  001276                                                            .WORD   702
     050640  051252                                                            .WORD   T182SSR
     050642  012046                                                            .WORD   PKTSSR
3641 050644  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
3642 050650                    30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     050650  104406                                                            TRAP    C$CLP1
3643
3644
3645                           ;       Do WRITE CHARACTERISTICS to select reserved unit 7
3646 050652  005037  002222            CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
3647 050656  012704  051640            MOV     #T18PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3648 050662  004737  010662            JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
3649 050666                            FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
3650 050702  103407                    BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
3651 050704  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3652 050706                            NEXT.ERRNO
3653 050706                    42$:    ERRDF   ERRNO,T18SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     050706  104455                                                            TRAP    C$ERDF
     050710  001277                                                            .WORD   703
     050712  051215                                                            .WORD   T18SSR
     050714  012046                                                            .WORD   PKTSSR
3654 050716  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
3655 050722                    50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     050722  104406                                                            TRAP    C$CLP1
3656
3657                           ;       Clear message buffer
3658 050724  012701  051662            MOV     #T18BFR,R1              ;GET MESSAGE BUFFER ADDRESS
3659 050730  013700  051654            MOV     T18DATA+4,R0            ;SIZE OF MESSAGE BUFFER IN BYTES
3660 050734  105021            60$:    CLRB    (R1)+                   ;CLEAR A BYTE
3661 050736  005300                    DEC     R0                      ;DONE?
3662 050740  003375                    BGT     60$                     ;BR IF NO
3663                           ;       Do a Write Subsystem READ STATUS
3664 050742  004737  051466            JSR     PC,T18SRD               ;SETUP PACKET FOR READ STATUS
3665 050746  012704  051710            MOV     #T18PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
3666 050752  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
3667 050756  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3668 050762                            FORCERROR       62$                     ;@@DFORCE ERROR IF FORCER=1
3669 050776  103407                    BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
3670 051000  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3671 051002                            NEXT.ERRNO
3672 051002                    62$:    ERRDF   ERRNO,T183SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     051002  104455                                                            TRAP    C$ERDF
     051004  001300                                                            .WORD   704
     051006  051316                                                            .WORD   T183SSR
     051010  012046                                                            .WORD   PKTSSR
3673 051012  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
3674 051016                    70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     051016  104406                                                            TRAP    C$CLP1
3675
3676
3677                           ;       Set first 8 words of expd message buffer = to recv since not testing
3678                           ;       Set unused bits in Read Status expd equal rcvd
3679 051020  004737  051530            JSR     PC,T18SETEXP            ;SET SOME EXPD TO RECV
```

```
                                        ;       If any transport interface signals are asserted then Print Error
                                                CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
                                                MOV     #T18BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
                                                MOV     #T18EXP,R2              ;EXPD ADDRESS
                                                MOV     #10.,R3                 ;NUMBER OF WORDS TO COMPARE
                                                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
                                                FORCERROR       82$,NOTSSR      ;BSD
                                                BCS     90$                     ;BR IF YES
                                                NEXT.ERRNO
                                        82$:    ERRHRD  ERRNO,T18CMP,MSGSTAT    ;REPORT ERROR
                                                                                        TRAP    C$ERHRD
                                                                                        .WORD   705
                                                                                        .WORD   T18CMP
                                                                                        .WORD   MSGSTAT
                                        90$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                        TRAP    C$CLP1

                                                TST     FATFLG                  ;ANY FATAL ERRORS ?
                                                BEQ     160$                    ;BRANCH IF NOT
                                                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
                                        160$:   JSR     PC,TSTLOOP              ;DO ITERATIONS?
                                                BCC     165$                    ;BR IF NO
                                                JMP     T18LOOP                 ;LOOP UNTIL ITERATIONS DONE
                                        165$:
                                                EXIT    TST
                                                                                        TRAP    C$EXIT
                                                                                        .WORD   L10065-.


                                        ;+
                                        ;LOCAL STORAGE FOR THIS TEST
                                        ;-

                                        T18MSK:                                 ;MASK OF UNUSED BITS IN READ STATUS BYTES
                                                .BYTE   ^C<000>                 ;BYTE 0 MASK
                                                .BYTE   ^C<340>                 ;BYTE 1
                                                .BYTE   ^C<277>                 ;BYTE 2
                                                .BYTE   0                       ;MAKE IT EVEN

                                        T18EXP:                                 ;EXPECTED DATA BUFFER
                                                .WORD   0                       ;MESSAGE TYPE
                                                .WORD   0                       ;DATA FIELD LENGTH
                                                .WORD   0                       ;RBPCR
                                                .WORD   0                       ;XST0
                                                .WORD   0                       ;XST1
                                                .WORD   0                       ;XST2
                                                .WORD   0                       ;XST3
                                                .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUB)
                                                .WORD   0                       ;READ STATUS BYTE 1/0
                                                .WORD   0                       ;READ STATUS BYTE 2

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


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

```
3730
3731 051156     123     164     141  TST18ID:        .ASCIZ  'Static Transport Bus Interface
3732 051215     127     122     111  T18SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
3733 051252     127     122     111  T182SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
3734 051316     127     122     111  T183SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
3735 051363     124     162     141  T18CMP: .ASCIZ  'Transport Bus Interface Signals NOT Negated After Unit 7 Selected
3736                                          .EVEN
3737
3738                                 ;+
3739                                 ; SETUP T18PK2 PACKET FOR READ STATUS
3740                                 ;-
3741 051466                         T18SRD:
3742 051466                                 SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
3743 051472  012700  051720                MOV     #T18DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
3744 051476  112720  000005                MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
3745 051502  105010                        CLRB    (R0)                    ;CLEAR BSEL1
3746 051504  000207                        RTS     PC                      ;RETURN
3747
3748                                 ;+
3749                                 ; SETUP T18PK2 PACKET FOR WRITE MISC.
3750                                 ;-
3751 051506                         T18SMISC:
3752 051506                                 SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
3753 051512  012700  051720                MOV     #T18DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
3754 051516  112720  000010                MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
3755 051522  112710  000200                MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
3756 051526  000207                        RTS     PC                      ;RETURN
3757
3758                                 ;+
3759                                 ;Set first 8 words of expd message buffer = to recv since not testing
3760                                 ;       Set unused bits in Read Status expd equal rcvd
3761                                 ;-
3762 051530                         T18SETEXP:
3763 051530  012702  051126                MOV     #T18EXP,R2              ;GET EXPD
3764 051534  012703  051662                MOV     #T18BFR,R3              ;GET READ STATUS RECV BUFFER
3765 051540  012700  000010                MOV     #8.,R0                  ;SET FIRST 8 WORDS EXP=RECV
3766 051544  012322                5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
3767 051546  005300                        DEC     R0                      ;DONE FIRST 8 WORDS?
3768 051550  003375                        BGT     5$                      ;BR IF NO
3769 051552  012701  051122                MOV     #T18MSK,R1              ;GET UNUSED BIT MASK
3770 051556  013712  051152                MOV     T18XS,(R2)              ;SETUP BASE EXPECTED BYTE 1/0
3771 051562  013762  051154  000002        MOV     T18XS+2,2(R2)           ;SETUP BASE EXPECTED BYTE 2
3772 051570  011300                        MOV     (R3),R0                 ;GET RECV BYTE 1 AND BYTE 0
3773 051572  041100                        BIC     (R1),R0                 ;CLEAR ALL BUT UNUSED
3774 051574  040012                        BIC     R0,(R2)                 ;CLEAR UNUSED IN EXP
3775 051576  050012                        BIS     R0,(R2)                 ;SET UNUSED EXPD=RECV FOR COMPARE
3776 051600  016300  000002                MOV     2(R3),R0                ;GET RECV BYTE 2
3777 051604  046100  000002                BIC     2(R1),R0                ;CLEAR ALL BUT UNUSED
3778 051610  040062  000002                BIC     R0,2(R2)                ;CLEAR UNUSED IN EXPD
3779 051614  050062  000002                BIS     P0,2(R2)                ;SET UNUSED EXPD=RECV FOR COMPARE
3780 051620  105062  000003                CLRB    3(R2)                   ;CLEAR EXPD BYTE 3 (UNUSED)
3781 051624  105063  000003                CLRB    3(R3)                   ;CLEAR RECV BYTE 3 (UNUSED)
3782 051630  000207                        RTS     PC                      ;RETURN
3783
3785         051640                        .=<.+10>&177770
3787
3788                                 ;WRITE CHARARTERISTICS COMMAND PACKET
```

```
                                        ;
                        T18PACKET:                      ;COMMAND PACKET FOR TEST
100004                          .WORD   100004          ;WRITE CHARACTERISTICS COMMAND, WITH ACK
051650                          .WORD   T18DATA         ;ADDRESS OF CHARACTERISTICS BLOCK
000000                          .WORD   0
000012                          .WORD   10.             ;MESSAGE PACKET MINIMUM SIZE

                        T18DATA:                        ;CHARACTERISTICS DATA BLOCK
051662                          .WORD   T18BFR          ;ADDRESS OF MESSAGE BUFFER
000000                          .WORD   0
000024                          .WORD   20.             ;LENGTH OF MESSAGE BUFFER
000000                          .WORD   0               ;ESS,ENB,EAI,ERI
000007                          .WORD   7               ;SELECT RESERVED UNIT 7

                        T18BFR:                         ;MESSAGE BUFFER
000000                          .WORD   0               ;MESSAGE TYPE
000000                          .WORD   0               ;DATA FIELD LENGTH
000000                          .WORD   0               ;RBPCR
000000                          .WORD   0               ;XST0
000000                          .WORD   0               ;XST1
000000                          .WORD   0               ;XST2
000000                          .WORD   0               ;XST3
000000                          .WORD   0               ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
000000                          .WORD   0               ;READ STATUS BYTE 1/0 RETURNED
000000                          .WORD   0               ;READ STATUS BYTE 2
                        ;
                        ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
                        ;
051710                          .=.+10>&177770
                        T18PK2:
100006                          .WORD   P.WRTSUB!P.ACK  ;WRITE SUBSYSTEM WITH ACK
051720                          .WORD   T18DT2          ;LOW ADDRESS OF DATA BLOCK
000000                          .WORD   0               ;HIGH ADDRESS OF DATA BLOCK
000010                          .WORD   8.              ;BUFFER EXTENT

051720                  T18DT2:                         ;DATA BLOCK
000                             .BYTE   0               ;BSEL0
000                             .BYTE   0               ;BSEL1
000000                          .WORD   0               ;SEL2
000000                          .WORD   0               ;DATA

                                ENDTST
051726                                                          L10065:
051726                                                                  TRAP    C$ETST
104401
                                .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
                                ;
                                ;
                                ;
                                        BGNTST
                                                                                T8::
051730  012700  060142                  MOV     @TST19ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
051734  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
051740  012737  000012  002216          MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
051746                          T19LOOP:
                                ;
                                        .SBTTL  TEST  8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST
                                ;
                                ;**
                                ; TEST 8: SUBTEST 1:
                                ;
                                ; SUBTEST DESCRIPTION:
                                ;
                                ;       This subtest verifies the Transport Control loopback
                                ;       path can transmit and receive correctly. The
                                ;       control signals are all loopback signals other
                                ;       than the read/write data (IW<7:0> and IR<7:0>).
                                ;
                                ; TEST STEPS:
                                ;
                                ;       The loopback signals IFAD,ITADO,ITAD1 are the tape unit select
                                ;       lines. Since reserved unit 7 must remain selected these signals
                                ;       are always set low. This further means the signals they drive
                                ;       (ISPEED,.TRDY,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,IMER) fl p flop
                                ;         signals NOT=0 Then Print Error
                                ;   END
                                ;
                                        BGNSUB                          ;////////////// BEGIN SUBTEST //////////
                                                                                    T8.1:
051746  104402                                                                        TRAP    C$BSUB
                                ;       Write to TSSR register to soft initialize the controller
                                5$:
051750  004737  015774          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
051754  103405                  BCS     10$                     ;BR IF SOFT INIT OKAY
051756  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
051760                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
051760  104455                                                                      TRAP    C$ERDF
051762  001440                                                                      .WORD   800
051764  003652                                                                      .WORD   SFIERR
051766  012034                                                                      .WORD   SFIMSG
                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
051770  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
051774  012704  062270          MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
052000  004737  010662          JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
052004                          FORCERROR       12$                     ;@@DFORCE ERROR IF FORCER=1
052020  103407                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
052022  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
052024                          NEXT.ERRNO
052024                  12$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
052024  104455                                                                      TRAP    C$ERDF
052026  001441                                                                      .WORD   801
052030  060203                                                                      .WORD   T19SSR
052032  012046                                                                      .WORD   PKTSSR
052034  005237  002222          15$:    INC     FAIFLG                  ;SET FATAL ERROR FLAG
052040                          CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
052040  104406                                                                      TRAP    C$CLP1
                                ;       If Extended Features Hardware Switch Clear then:
                                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
052042  012701  062312          MOV     #T19BFR,R1              ;MESSAGE BUFFER ADDRESS
052046  032761  000200  000012  BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
052054  001026                  BNE     30$                     ;BR IF YES
052056  004737  062142          JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
052062  012704  062440          MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
052066  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
052072  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
052076                          FORCERROR       22$                     ;@@DFORCE ERROR IF FORCER=1
052112  103407                  BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
052114  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
052116                          NEXT.ERRNO
052116                  22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
052116  104455                                                                      TRAP    C$ERDF
052120  001442                                                                      .WORD   802
052122  060240                                                                      .WORD   T192SSR
052124  012046                                                                      .WORD   PKTSSR
052126  005237  002222          30$:    INC     FATFLG                  ;SET FATAL ERROR FLAG
052132                          CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

```
052132  104406                                                                   TRAP    C$CLP1
                                 ;       Do WRITE CHARACTERISTICS to select reserved unit 7
052134  005037  002222                   CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
052140  012704  062270                   MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
052144  004737  010662                   JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
052150                                   FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
052164  103407                           BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
052166  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
052170                                   NEXT.ERRNO
052170                           42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
052170  104455                                                                   TRAP    C$ERDF
052172  001443                                                                   .WORD   803
052174  060203                                                                   .WORD   T19SSR
052176  012046                                                                   .WORD   PKTSSR
052200  005237  002222                   INC     FATFLG                  ;SET FATAL ERROR FLAG
052204                           50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
052204  104406                                                                   TRAP    C$CLP1
                                 ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
052206  012700  000100                   MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
052212  052700  000040                   BIS     #NP.LOOP,R0             ;SET LOOPBACK ENABLE
052216  004737  062002                   JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
052222  012704  062440                   MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
052226  010465  000000                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
052232  004737  016336                   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
052236                                   FORCERROR       62$                     ;@@FORCE ERROR IF FORCER=1
052252  103407                           BCS     70$                     ;BR IF CARRY SET (GOOD RETURN)
052254  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
052256                                   NEXT.ERRNO
052256                           62$:    ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
052256  104455                                                                   TRAP    C$ERDF
052260  001444                                                                   .WORD   804
052262  060351                                                                   .WORD   T194SSR
052264  012046                                                                   .WORD   PKTSSR
052266  005237  002222                   INC     FATFLG                  ;SET FATAL ERROR FLAG
052272                           70$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
052272  104406                                                                   TRAP    C$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))
052274  005000                           CLR     R0                      ;WRITE 0'S
052276  042700  000200                   BIC     #WC.IFAD,R0             ;IFAD MUST  ALWAYS =0
052302  042700  000100                   BIC     #WC.I0TAD,R0            ;ITAD0 MUST ALWAYS =0
052306  042700  000040                   BIC     #WC.I1TAD,R0            ;ITAD1 MUST ALWAYS =0
052312  004737  062102                   JSR     PC,T19WCTL              ;SETUP PACKET FOR WRITE CONTROL
052316  012704  062440                   MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
052322  010465  000000                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
052326  004737  016336                   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
052332                                   FORCERROR       82$                     ;@@FORCE ERROR IF FORCER=1
052346  103407                           BCS     90$                     ;BR IF CARRY SET (GOOD RETURN)
052350  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
052352                                   NEXT.ERRNO
052352                           82$:    ERRDF   ERRNO,T197SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
052352  104455                                                                   TRAP    C$ERDF
052354  001445                                                                   .WORD   805
052356  060523                                                                   .WORD   T197SSR
052360  012046                                                                   .WORD   PKTSSR
```

```
3990 052362  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
3991 052366                        90$:     CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     052366  104406                                                                TRAP    C$CLP1
3992 052370  005000                CLR     R0                              ;SET FORMAT DRIVE DATA=0
3993 052372  004737  062122        JSR     PC,T19WFMT                      ;SETUP PACKET FOR WRITE FORMAT
3994 052376  012704  062440        MOV     #T19PK2,R4                      ;GET WRITE SUBSYSTEM COMMAND PACKET
3995 052402  010465  000000        MOV     R4,TSDB(R5)                     ;SET THE PACKET ADDRESS TO EXECUTE
3996 052406  004737  016336        JSR     PC,CHKTSSR                      ;WAIT FOR SSR TO SET
3997 052412                        FORCERROR       102$                            ;@@DFORCE ERROR IF FORCER=1
3998 052426  103407                BCS     110$                            ;BR IF CARRY SET (GOOD RETURN)
3999 052430  010001                MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
4000 052432                        NEXT.ERRNO
4001 052432                        102$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     052432  104455                                                                TRAP    C$ERDF
     052434  001446                                                                .WORD   806
     052436  060572                                                                .WORD   T198SSR
     052440  012046                                                                .WORD   PKTSSR
4002 052442  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4003 052446                        110$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     052446  104406                                                                TRAP    C$CLP1
4004                               ;        Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
4005 052450  004737  061760        JSR     PC,T19RSFIF                     ;SETUP PKT FOR WRITE MISC Reset Tape Status F-FLOPS
4006 052454  012704  062440        MOV     #T19PK2,R4                      ;GET WRITE SUBSYSTEM COMMAND PACKET
4007 052460  010465  000000        MOV     R4,TSDB(R5)                     ;SET THE PACKET ADDRESS TO EXECUTE
4008 052464  004737  016336        JSR     PC,CHKTSSR                      ;WAIT FOR SSR TO SET
4009 052470                        FORCERROR       122$                            ;@@DFORCE ERROR IF FORCER=1
4010 052504  103407                BCS     130$                            ;BR IF CARRY SET (GOOD RETURN)
4011 052506  010001                MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
4012 052510                        NEXT.ERRNO
4013 052510                        122$:   ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     052510  104455                                                                TRAP    C$ERDF
     052512  001447                                                                .WORD   807
     052514  060240                                                                .WORD   T192SSR
     052516  012046                                                                .WORD   PKTSSR
4014 052520  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4015 052524                        130$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     052524  104406                                                                TRAP    C$CLP1
4016                               ;        Do a Write Subsystem READ STATUS
4017                               ;        If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
4018                               ;          signals NOT=0 Then Print Error
4019 052526  004737  061740        JSR     PC,T19SRD                       ;SETUP PACKET FOR READ STATUS
4020 052532  012704  062440        MOV     #T19PK2,R4                      ;GET WRITE SUBSYSTEM COMMAND PACKET
4021 052536  010465  000000        MOV     R4,TSDB(R5)                     ;SET THE PACKET ADDRESS TO EXECUTE
4022 052542  004737  016336        JSR     PC,CHKTSSR                      ;WAIT FOR SSR TO SET
4023 052546                        FORCERROR       132$                            ;@@DFORCE ERROR IF FORCER=1
4024 052562  103407                BCS     140$                            ;BR IF CARRY SET (GOOD RETURN)
4025 052564  010001                MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
4026 052566                        NEXT.ERRNO
4027 052566                        132$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     052566  104455                                                                TRAP    C$ERDF
     052570  001450                                                                .WORD   808
     052572  060304                                                                .WORD   T193SSR
     052574  012046                                                                .WORD   PKTSSR
4028 052576  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4029 052602                        140$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     052602  104406                                                                TRAP    C$CLP1
4030 052604  004737  062200        JSR     PC,T19SETEXP                    ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
```

```
4031 052610  012701  060042                    MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4032 052614  012702  062332                    MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
4033 052620  011211                            MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
4034 052622  042711  002000                    BIC     #S1.ICER,(R1)           ;SET EXPD ICER  =0
4035 052626  042711  001000                    BIC     #S1.IFMK,(R1)           ;SET EXPD IFMK  =0
4036 052632  042711  000400                    BIC     #S1.IHER,(R1)           ;SET EXPD IHER  =0
4037 052636  016261  000002  000002            MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
4038 052644  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4039 052646  012701  062312                    MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4040 052652  012702  060022                    MOV     #T19EXP,R2              ;EXPD ADDRESS
4041 052656  012703  000024                    MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
4042 052662  004737  011500                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4043 052666                                    FORCERROR       152$,NOTS:R     ;@@D
4044 052676  103404                            BCS     160$                    ;BR IF YES
4045 052700                                    NEXT.ERRNO
4046 052700                            152$:   ERRHRD  ERRNO,T197CMP,MSGLOOP   ;REPORT ERROR
     052700  104456                                                                    TRAP    C$ERHRD
     052702  001451                                                                    .WORD   809
     052704  061243                                                                    .WORD   T197CMP
     052706  013064                                                                    .WORD   MSGLOOP
4047 052710                            160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     052710  104406                                                                    TRAP    C$CLP1
4048                                   ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4049 052712  005037  057754                    CLR     T19PREV                 ;INIT 1-0 TRANSITION FLAG
4050 052716  012703  002752                    MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
4051 052722  012300                    200$:   MOV     (R3)+,R0                ;GET A TEST PATTERN
4052 052724  010337  002316                    MOV     R3,TSTPTR               ;SAVE POINTER INTO TSTBLK
4053 052730  042700  000200                    BIC     #WC.IFAD,R0             ;IFAD MUST  ALWAYS =0
4054 052734  042700  000100                    BIC     #WC.IOTAD,R0            ;ITAD0 MUST ALWAYS =0
4055 052740  042700  000040                    BIC     #WC.I1TAD,R0            ;ITAD1 MUST ALWAYS =0
4056 052744  010037  002312                    MOV     R0,DATA                 ;SET DATA PATTERN
4057                                   ;       Do Write Subsystem Write Control to Drive loopback signals group 1.
4058                                   ;@@D    CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
4059 052750  013700  002312                    MOV     DATA,R0                 ;GET TEST PATTERN
4060 052754  004737  062224                    JSR     PC,T19CNVT              ;CONVERT PATTERN TO CONTROL DRIVE MASK
4061                                                                           ;R0 CONTAINS WRITE CONTROL DATA HERE
4062 052760  004737  062102                    JSR     PC,T19WCTL              ;SETUP PACKET FOR WRITE CONTROL
4063 052764  012704  062440                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4064 052770  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4065 052774  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4066 053000                                    FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
4067 053014  103407                            BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
4068 053016  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4069 053020                                    NEXT.ERRNO
4070 053020                            212$:   ERRDF   ERRNO,T197SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053020  104455                                                                    TRAP    C$ERDF
     053022  001452                                                                    .WORD   810
     053024  060523                                                                    .WORD   T197SSR
     053026  012046                                                                    .WORD   PKTSSR
4071 053030  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4072 053034                            220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053034  104406                                                                    TRAP    C$CLP1
4073
4074                                   ;       Do Write Subsystem Write Format to Drive loopback signals group 2.
4075                                   ;@@D    CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
4076 053036  013700  002312                    MOV     DATA,R0                 ;GET TEST PATTERN
4077 053042  004737  062224                    JSR     PC,T19CNVT              ;CONVERT PATTERN TO FORMAT DRIVE MASK
```

```
4078 053046  000300                    SWAB    R0                      ;WRITE FORMAT DATA RETURNED IN HIGH BYTE
4079 053050  004737  062122            JSR     PC,T19WFMT              ;SETUP PACKET FOR WRITE FORMAT
4080 053054  012704  062440            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4081 053060  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4082 053064  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4083 053070                            FORCERROR       232$                    ;@@DFORCE ERROR IF FORCER=1
4084 053104  103407                    BCS     240$                    ;BR IF CARRY SET (GOOD RETURN)
4085 053106  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4086 053110                            NEXT.ERRNO
4087 053110                    232$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053110  104455                                                            TRAP    C$ERDF
     053112  001453                                                            .WORD   811
     053114  060572                                                            .WORD   T198SSR
     053116  012046                                                            .WORD   PKTSSR
4088 053120  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4089 053124                    240$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053124  104406                                                            TRAP    C$CLP1
4090                           ;       Do a Write Subsystem READ STATUS
4091 053126  004737  061740            JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
4092 053132  012704  062440            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4093 053136  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4094 053142  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4095 053146                            FORCERROR       252$                    ;@@DFORCE ERROR IF FORCER=1
4096 053162  103407                    BCS     260$                    ;BR IF CARRY SET (GOOD RETURN)
4097 053164  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4098 053166                            NEXT.ERRNO
4099 053166                    252$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053166  104455                                                            TRAP    C$ERDF
     053170  001454                                                            .WORD   812
     053172  060304                                                            .WORD   T193SSR
     053174  012046                                                            .WORD   PKTSSR
4100 053176  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4101 053202                    260$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053202  104406                                                            TRAP    C$CLP1
4102                           ;       If loopback data NOT= data sent Then Print Error
4103 053204  004737  062200            JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4104 053210  012701  060042            MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4105 053214  012702  062332            MOV     #T198FSTA,R2            ;GET RECV READ STATUS
4106 053220  013711  002312            MOV     DATA,(R1)               ;SET EXPD WORD #0 TO TEST DATA FIRST
4107 053224  013700  057754            MOV     T19PREV,R0              ;GET PREVIOUS DATA PATTERN
4108 053230  013703  002312            MOV     DATA,R3                 ;GET CURRENT PATTERN
4109 053234  012704  000400            MOV     #S1.IHER,R4             ;SETUP IHER EXPECTED
4110 053240  040411                    BIC     R4,(R1)                 ;SET EXPD IHER  =0
4111 053242  030400                    BIT     R4,R0                   ;PREVIOUS =1?
4112 053244  001403                    BEQ     275$                    ;BR IF NO
4113 053246  030403                    BIT     R4,R3                   ;CURRENT  =0?
4114 053250  001001                    BNE     275$                    ;BR IF NO
4115 053252  050411                    BIS     R4,(R1)                 ;SET EXPD IHER  =1
4116 053254  012704  001000    275$:   MOV     #S1.IFMK,R4             ;SETUP IFMK EXPECTED
4117 053260  040411                    BIC     R4,(R1)                 ;SET EXPD IFMK  =0
4118 053262  030400                    BIT     R4,R0                   ;PREVIOUS =1?
4119 053264  001403                    BEQ     280$                    ;BR IF NO
4120 053266  030403                    BIT     R4,R3                   ;CURRENT  =0?
4121 053270  001001                    BNE     280$                    ;BR IF NO
4122 053272  050411                    BIS     R4,(R1)                 ;SET EXPD IFMK  =1
4123 053274  012704  002000    280$:   MOV     #S1.ICER,R4             ;SETUP ICER EXPECTED
4124 053300  040411                    BIC     R4,(R1)                 ;SET EXPD ICER  =0
```

```
4125 053302  030400                        BIT     R4,R0                   ;PREVIOUS =1?
4126 053304  001403                        BEQ     285$                    ;BR IF NO
4127 053306  030403                        BIT     R4,R3                   ;CURRENT  =0?
4128 053310  001001                        BNE     285$                    ;BR IF NO
4129 053312  050411                        BIS     R4,(R1)                 ;SET EXPD ICER  =1
4130 053314  011100                285$:   MOV     (R1),R0                 ;GET EXPD WORD
4131                                       If previous IIDENT=1 and current is IIDENT=1 then EXPD= 0 else 1
4132 053316  012704  004000                MOV     #S1.IIDENT,R4           ;IIDENT
4133 053322  050400                        BIS     R4,R0                   ;ASSUME EXPD=1
4134 053324  030437  057754                BIT     R4,T19PREV              ;PREVIOUS IIDENT=1?
4135 053330  001403                        BEQ     288$                    ;BR IF NO
4136 053332  030403                        BIT     R4,R3                   ;IS CURRENT IIDENT=1?
4137 053334  001401                        BEQ     288$                    ;BR IF NO
4138 053336  040400                        BIC     R4,R0                   ;SET EXPD=0
4139 053340  052700  040000          288$:   BIS     #S1.I2RES,R0            ;IRESV2 EXPD ALWAYS=1
4140 053344  052700  020000                BIS     #S1.I1RES,R0            ;IRESV1 EXPD ALWAYS=1
4141 053350  042700  100000                BIC     #S1.PARERR,R0           ;IGNORE PARERR
4142 053354  032712  100000                BIT     #S1.PARERR,(R2)         ;IS PARERR SET IN RECV?
4143 053360  001402                        BEQ     290$                    ;BR IF NO
4144 053362  052700  100000                BIS     #S1.PARERR,R0           ;SET IN EXPD
4145 053366  010011                290$:   MOV     R0,(R1)                 ;SETUP FINAL EXPD IN WORD #8
4146 053370  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
4147 053376  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4148 053400  012701  062312                MOV     #T198FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4149 053404  012702  060022                MOV     #T19EXP,R2              ;EXPD ADDRESS
4150 053410  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
4151 053414  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4152 053420                                FORCERROR       302$,NOTSSR     ;###
4153 053430  103404                        BCS     310$                    ;BR IF YES
4154 053432                                NEXT.ERRNO
4155 053432                        302$:   ERRHRD  ERRNO,T198CMP,MSGLOOP   ;REPORT ERROR
     053432  104456                                                        TRAP    C$ERHRD
     053434  001455                                                        .WORD   813
     053436  061331                                                        .WORD   T198CMP
     053440  013064                                                        .WORD   MSGLOOP
4156 053442                        310$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053442  104406                                                        TRAP    C$CLP1
4157                               ;       Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
4158 053444  004737  061760                JSR     PC,T19RSFIF             ;SETUP PKT FOR WRITE MISC Reset STATUS
4159 053450  012704  062440                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4160 053454  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4161 053460  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4162 053464                                FORCERROR       322$            ;###FORCE ERROR IF FORCER=1
4163 053500  103407                        BCS     330$                    ;BR IF CARRY SET (GOOD RETURN)
4164 053502  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4165 053504                                NEXT.ERRNO
4166 053504                        322$:   ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053504  104455                                                        TRAP    C$ERDF
     053506  001456                                                        .WORD   814
     053510  060240                                                        .WORD   T192SSR
     053512  012046                                                        .WORD   PKTSSR
4167 053514  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4168 053520                        330$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053520  104406                                                        TRAP    C$CLP1
4169                               ;       Do a Write Subsystem READ STATUS
4170 053522  004737  061740                JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
4171 053526  012704  062440                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
```

```
4172 053532  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4173 053536  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4174 053542                                FORCERROR       342$                    ;@@DFORCE ERROR IF FORCER=1
4175 053556  103407                        BCS     350$                    ;BR IF CARRY SET (GOOD RETURN)
4176 053560  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4177 053562                                NEXT.ERRNO
4178 053562                        342$:   ERRDF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     053562  104455                                                                TRAP    C$ERDF
     053564  001457                                                                .WORD   815
     053566  060304                                                                .WORD   T193SSR
     053570  012046                                                                .WORD   PKTSSR
4179 053572  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4180 053576                        350$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053576  104406                                                                TRAP    C$CLP1
4181 053600  004737  062200                JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4182 053604  012701  060042                MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4183 053610  012702  062332                MOV     #T198FSTA,R2            ;GET RECV READ STATUS
4184 053614  011211                        MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
4185 053616  042711  002000                BIC     #S1.ICER,(R1)           ;SET EXPD ICER  =0
4186 053622  042711  001000                BIC     #S1.IFMK,(R1)           ;SET EXPD IFMK  =0
4187 053626  042711  000400                BIC     #S1.IHER,(R1)           ;SET EXPD IHER  =0
4188 053632  016261  000002  000002        MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
4189 053640  005000                        CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4190 053642  012701  062312                MOV     #T198FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4191 053646  012702  060022                MOV     #T19EXP,R2              ;EXPD ADDRESS
4192 053652  012703  000024                MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
4193 053656  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4194 053662                                FORCERROR       362$,NOTSSR             ;@@D
4195 053672  103404                        BCS     370$                    ;BR IF YES
4196 053674                                NEXT.ERRNO
4197 053674                        362$:   ERRHRD  ERRNO,T197CMP,MSGSTAT   ;REPORT ERROR
     053674  104456                                                                TRAP    C$ERHRD
     053676  001460                                                                .WORD   816
     053700  061243                                                                .WORD   T197CMP
     053702  012350                                                                .WORD   MSGSTAT
4198 053704                        370$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     053704  104406                                                                TRAP    C$CLP1
4199
4200 053706  013737  002312  057754        MOV     DATA,T19PREV            ;SETUP PREVIOUS DATA FOR EXPD CALC.
4201 053714  013703  002316                MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
4202 053720  020327  003062                CMP     R3,#TBLEND              ;END OF TSTBLK?
4203 053724  103002                        BHIS    400$                    ;BR IF YES
4204 053726  000137  052722                JMP     200$                    ;DO NEXT TSTBLK PATTERN
4205 053732                        400$:
4206
4207 053732                                ENDSUB                          ;////////////// END SUBTEST //////////
     053732                                                                        L10067:
     053732  104403                                                                TRAP    C$ESUB
4208
4209 053734  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
4210 053740  001402                        BEQ     460$                    ;BRANCH IF NOT
4211 053742  004737  017202                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4212 053746                        460$:
4213
4214
4215
4216
```

```
4217
4218                                    .SBTTL  TEST  8: SUBTEST 2: LOOPBACK READ/WRITE SIGNALS TEST
4219
4220                                ;**
4221                                ; TEST 8: SUBTEST 2:
4222                                ;
4223                                ; SUBTEST DESCRIPTION:
4224                                ;
4225                                ;       This subtest verifies the Read/Write data loopback path.
4226                                ;       The Read/Write data signals are IR<7:0> and IW<7:0>
4227                                ;       respectively.
4228                                ;
4229                                ; TEST STEPS:
4230                                ;
4231                                ;
4232                                ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4233                                ;    BEGIN
4234                                ;       Write to TSSR register to soft initialize the controller
4235                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
4236                                ;       If Extended Features Hardware Switch Clear then:
4237                                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
4238                                ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
4239                                ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
4240                                ;       Do a WRITE NPR to set loopback and tape direction OUT
4241                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4242                                ;       Do a READ FIFO with tape direction OUT to load tape out write latch
4243                                ;       Do a WRITE NPR to set loopback and tape direction IN
4244                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
4245                                ;         to strobe loopback data into FIFO.
4246                                ;       Do a READ FIFO with tape direction IN to read data
4247                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
4248                                ;       Do a Write Subsystem READ STATUS
4249                                ;       If Input Ready  NOT=1    Then Print Error
4250                                ;       If Output Ready NOT=0    Then Print Error
4251                                ;       If Data In Miss NOT=0    Then Print Error
4252                                ;    END
4253                                ;--
4254 053746                             BGNSUB                          ;////////////// BEGIN SUBTEST //////////
      053746                                                                    T8.2:
      053746  104402                                                           TRAP    C$BSUB
4255                                ;       Write to TSSR register to soft initialize the controller
4256 053750                         5$:
4257 053750  004737  015774                JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
4258 053754  103405                        BCS     10$                     ;BR IF SOFT INIT OKAY
4259 053756  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4260 053760                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
      053760  104455                                                           TRAP    C$ERDF
      053762  001460                                                           .WORD   816
      053764  003652                                                           .WORD   SFIERR
      053766  012034                                                           .WORD   SFIMSG
4261                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
4262 053770  005037  002222        10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
4263 053774  012704  062270                MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
4264 054000  004737  010662                JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
4265 054004                                FORCERROR       12$                     ;%%FORCE ERROR IF FORCER=1
4266 054020  103407                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
4267 054022  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

```
4268 054024                                    NEXT.ERRNO
4269 054024                            12$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     054024  104455                                                              TRAP    C$ERDF
     054026  001461                                                              .WORD   817
     054030  060203                                                              .WORD   T19SSR
     054032  012046                                                              .WORD   PKTSSR
4270 054034  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4271 054040                            15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054040  104406                                                              TRAP    C$CLP1
4272                                   ;       If Extended Features Hardware Switch Clear then:
4273                                   ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
4274 054042  012701  062312                    MOV     #T19BFR,R1              ;MESSAGE BUFFER ADDRESS
4275 054046  032761  000200  000012            BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
4276 054054  001026                            BNE     30$                     ;BR IF YES
4277 054056  004737  062142                    JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
4278 054062  012704  062440                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4279 054066  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4280 054072  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4281 054076                                    FORCERROR       22$                     ;@@FORCE ERROR IF FORCER=1
4282 054112  103407                            BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
4283 054114  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4284 054116                                    NEXT.ERRNO
4285 054116                            22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054116  104455                                                              TRAP    C$ERDF
     054120  001462                                                              .WORD   818
     054122  060240                                                              .WORD   T192SSR
     054124  012046                                                              .WORD   PKTSSR
4286 054126  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4287 054132                            30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054132  104406                                                              TRAP    C$CLP1
4288                                   ;       Do WRITE CHARACTERISTICS to select reserved unit 7
4289 054134  012704  062270                    MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
4290 054140  004737  010662                    JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
4291 054144                                    FORCERROR       42$                     ;@@FORCE ERROR IF FORCER=1
4292 054160  103407                            BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
4293 054162  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4294 054164                                    NEXT.ERRNO
4295 054164                            42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     054164  104455                                                              TRAP    C$ERDF
     054166  001463                                                              .WORD   819
     054170  060203                                                              .WORD   T19SSR
     054172  012046                                                              .WORD   PKTSSR
4296 054174  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4297 054200                            50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054200  104406                                                              TRAP    C$CLP1
4298
4299
4300                                   ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4301 054202  012703  002752                    MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
4302 054206  012337  002312            100$:   MOV     (R3)+,DATA              ;GET A TEST PATTERN
4303 054212  042737  177400  002312            BIC     #^C<377>,DATA           ;DATA IS BYTE
4304 054220  010337  002316                    MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
4305                                   ;       Do a WRITE NPR to set loopback and tape direction OUT
4306 054224  012700  000100                    MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
4307 054230  052700  000040                    BIS     #NP.LOOP,R0             ;SET LOOPBACK
4308 054234  004737  062002                    JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
4309 054240  012704  062440                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
```

```
4310 054244  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4311 054250  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4312 054254                                FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
4313 054270  103407                        BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
4314 054272  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4315 054274                                NEXT.ERRNO
4316 054274                        102$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054274  104455                                                                  TRAP    C$ERDF
     054276  001464                                                                  .WORD   820
     054300  060351                                                                  .WORD   T194SSR
     054302  012046                                                                  .WORD   PKTSSR
4317 054304  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4318 054310                        105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054310  104406                                                                  TRAP    C$CLP1
4319                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4320 054312  012700  000001                MOV     #1,R0                   ;WRITE 1 BYTE
4321 054316  012701  002312                MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
4322 054322  004737  062046                JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
4323 054326  012704  062440                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4324 054332  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4325 054336  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4326 054342                                FORCERROR       107$                    ;@@DFORCE ERROR IF FORCER=1
4327 054356  103407                        BCS     110$                    ;BR IF CARRY SET (GOOD RETURN)
4328 054360  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4329 054362                                NEXT.ERRNO
4330 054362                        107$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054362  104455                                                                  TRAP    C$ERDF
     054364  001465                                                                  .WORD   821
     054366  060414                                                                  .WORD   T195SSR
     054370  012046                                                                  .WORD   PKTSSR
4331 054372  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4332 054376                        110$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054376  104406                                                                  TRAP    C$CLP1
4333                                ;       Do a READ FIFO with tape direction OUT to load tape out write latch
4334 054400  012700  000001                MOV     #1,R0                   ;SET READ BYTE COUNT
4335 054404  004737  062026                JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
4336 054410  012704  062440                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4337 054414  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4338 054420  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4339 054424                                FORCERROR       122$                    ;@@DFORCE ERROR IF FORCER=1
4340 054440  103407                        BCS     130$                    ;BR IF CARRY SET (GOOD RETURN)
4341 054442  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4342 054444                                NEXT.ERRNO
4343 054444                        122$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054444  104455                                                                  TRAP    C$ERDF
     054446  001466                                                                  .WORD   822
     054450  060460                                                                  .WORD   T196SSR
     054452  012046                                                                  .WORD   PKTSSR
4344 054454  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
4345 054460                        130$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054460  104406                                                                  TRAP    C$CLP1
4346                                ;       Do a WRITE NPR to set loopback and tape direction IN
4347 054462  005000                        CLR     R0                      ;CLR NP.OUT TO SET TAPE DIRECTION IN
4348 054464  052700  000040                BIS     #NP.LOOP,R0             ;SET LOOPBACK
4349 054470  004737  062002                JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
4350 054474  012704  062440                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4351 054500  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
```

```
4352 054504  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4353 054510                          FORCERROR       142$                    ;@@DFORCE ERROR IF FORCER=1
4354 054524  103407                  BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
4355 054526  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4356 054530                          NEXT.ERRNO
4357 054530                  142$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054530  104455                                                          TRAP    C$ERDF
     054532  001467                                                          .WORD   823
     054534  060351                                                          .WORD   T194SSR
     054536  012046                                                          .WORD   PKTSSR
4358 054540  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
4359 054544                  150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054544  104406                                                          TRAP    C$CLP1
4360                         ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
4361 054546  012700  000001          MOV     #1,R0                   ;WRITE 1 BYTE
4362 054552  012701  002312          MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
4363 054556  004737  062046          JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
4364 054562  012704  062440          MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4365 054566  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4366 054572  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4367 054576                          FORCERROR       162$                    ;@@DFORCE ERROR IF FORCER=1
4368 054612  103407                  BCS     170$                    ;BR IF CARRY SET (GOOD RETURN)
4369 054614  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4370 054616                          NEXT.ERRNO
4371 054616                  162$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054616  104455                                                          TRAP    C$ERDF
     054620  001470                                                          .WORD   824
     054622  060414                                                          .WORD   T195SSR
     054624  012046                                                          .WORD   PKTSSR
4372 054626  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
4373 054632                  170$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054632  104406                                                          TRAP    C$CLP1
4374                         ;       Do a READ FIFO with tape direction IN to read data
4375                         ;       If Data read from FIFO NOT= to Data sent Then Print Error
4376 054634  012700  000001          MOV     #1,R0                   ;SET READ BYTE COUNT
4377 054640  004737  062026          JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
4378 054644  012704  062440          MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4379 054650  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4380 054654  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4381 054660                          FORCERROR       182$                    ;@@DFORCE ERROR IF FORCER=1
4382 054674  103407                  BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
4383 054676  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4384 054700                          NEXT.ERRNO
4385 054700                  182$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     054700  104455                                                          TRAP    C$ERDF
     054702  001471                                                          .WORD   825
     054704  060460                                                          .WORD   T196SSR
     054706  012046                                                          .WORD   PKTSSR
4386 054710  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
4387 054714                  190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     054714  104406                                                          TRAP    C$CLP1
4388 054716  004737  062200          JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4389 054722  012701  060042          MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4390 054726  012702  062332          MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
4391 054732  013711  002312          MOV     DATA,(R1)               ;SET EXPD WORD #8 = DATA
4392 054736  016261  000002  000002  MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
4393 054744  005000                  CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
```

```
4394 054746  012701  062312          MOV     #T198FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4395 054752  012702  060022          MOV     #T19EXP,R2              ;EXPD ADDRESS
4396 054756  012703  000022          MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
4397 054762  004737  011500          JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4398 054766                          FORCERROR       2021,NOTSSR     ;BSD
4399 054776  103404                  BCS     2101                    ;BR IF YES
4400 055000                          NEXT.ERRNO
4401 055000                  2021:   ERRHRD  ERRNO,T199CMP,MSGSUB    ;REPORT ERROR
     055000  104456                                                          TRAP    C$ERHRD
     055002  001472                                                          .WORD   826
     055004  061420                                                          .WORD   T199CMP
     055006  013742                                                          .WORD   MSGSUB
4402 055010                  2101:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055010  104406                                                          TRAP    C$CLP1
4403                         ;       Do a Write Subsystem READ STATUS
4404                         ;       If Input Ready  NOT=1   Then Print Error
4405                         ;       If Output Ready NOT=0   Then Print Error
4406                         ;       If Data In Miss NOT=0   Then Print Error
4407 055012  004737  061740          JSR     PC,T19SRD               ;SETUP PACKET FOR READ STATUS
4408 055016  012704  062440          MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4409 055022  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4410 055026  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4411 055032                          FORCERROR       2121            ;BSDFORCE ERROR IF FORCER=1
4412 055046  103407                  BCS     2201                    ;BR IF CARRY SET (GOOD RETURN)
4413 055050  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4414 055052                          NEXT.ERRNO
4415 055052                  2121:   ERROF   ERRNO,T193SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     055052  104455                                                          TRAP    C$EROF
     055054  001473                                                          .WORD   827
     055056  060304                                                          .WORD   T193SSR
     055060  012046                                                          .WORD   PKTSSR
4416 055062  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
4417 055066                  2201:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055066  104406                                                          TRAP    C$CLP1
4418 055070  004737  062200          JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4419 055074  012701  060042          MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4420 055100  012702  062332          MOV     #T198FSTA,R2            ;GET RECV READ STATUS
4421 055104  012221                  MOV     (R2)+,(R1)+             ;SET EXPD WORD #8 = RECV TEMP
4422 055106  011211                  MOV     (R2),(R1)               ;SET EXPD WORD #9 = RECV TEMP
4423 055110  052711  000020          BIS     #S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
4424 055114  042711  000040          BIC     #S2.OUTRDY,(R1)         ;SET EXP OUTPUT READY= 0
4425 055120  042711  000200          BIC     #S2.DIM,(R1)            ;SET EXP DATA IN MISS = 0
4426 055124  005000                  CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4427 055126  012701  062312          MOV     #T198FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4428 055132  012702  060022          MOV     #T19EXP,R2              ;EXPD ADDRESS
4429 055136  012703  000024          MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
4430 055142  004737  011500          JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4431 055146                          FORCERROR       2321,NOTSSR     ;BSD
4432 055156  103404                  BCS     2401                    ;BR IF YES
4433 055160                          NEXT.ERRNO
4434 055160                  2321:   ERRHRD  ERRNO,T196CMP,MSGSTAT   ;REPORT ERROR
     055160  104456                                                          TRAP    C$ERHRD
     055162  001474                                                          .WORD   828
     055164  061160                                                          .WORD   T196CMP
     055166  012350                                                          .WORD   MSGSTAT
4435 055170                  2401    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055170  104406                                                          TRAP    C$CLP1
```

```
                                        FORCEXIT        2558            ;BBD
                                        MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
                                        CMP     R3,@TBLEND              ;END OF TSTBLK?
                                        BHIS    2558                    ;BR IF YES
                                        JMP     1008                    ;DO ANOTHER TSTBLK PATTERN
                                2558:

                                        ENDSUB                          ;////////////// END SUBTEST //////////
                                                                                L10070:
                                                                                        TRAP    C$ESUB
                                        TST     FATFLG                  ;ANY FATAL ERRORS ?
                                        BEQ     2608                    ;BRANCH IF NOT
                                        JSR     PC,CXDROP               ;TRY TO DROP THE UNIT
                                2608:


                                        .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 wr te
                                ;       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 MPR to set tape direction out and Loopback
                                ;       Do a WRITE MPR 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 MPR 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 direct on IN to read data
                                ;       If Data read from FIFO NOT= to Data sent Then Print Error
                                ;    END
                                ;
                                        BGNSUB                          ;////////////// BEGIN SUBTEST //////////
                                                                                T8.3:
```

```
     055234  104402                                                                     TRAP    C$BSUB
                                     ;      Write to TSSR register to soft initialize the controller
     055236                          5$:
     055236  004737  015774                 JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
     055242  103405                         BCS     10$                     ;BR IF SOFT INIT OKAY
     055244  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
     055246                                 ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     055246  104455                                                                     TRAP    C$ERDF
     055250  001474                                                                     .WORD   828
     055252  003652                                                                     .WORD   SFIERR
     055254  012034                                                                     .WORD   SFIMSG
                                     ;      Do WRITE CHARACTERISTICS to check for Extended Features Switch
     055256  005037  002222          10$:   CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
     055262  012704  062270                 MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
     055266  004737  010662                 JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
     055272                                 FORCERROR       12$                     ;BBDFORCE ERROR IF FORCER=1
     055306  103407                         BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
     055310  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
     055312                                 NEXT,ERRNO
     055312                          12$:   ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     055312  104455                                                                     TRAP    C$ERDF
     055314  001475                                                                     .WORD   829
     055316  060203                                                                     .WORD   T19SSR
     055320  012046                                                                     .WORD   PKTSSR
     055322  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
     055326                          15$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055326  104406                                                                     TRAP    C$CLP1
                                     ;      If Extended Features Hardware Switch Clear then:
                                     ;        Do Write Subsystem Write Miscellaneous to Set Extended Features.
     055330  012701  062312                 MOV     #T19BFR,R1              ;MESSAGE BUFFER ADDRESS
     055334  032761  000200  000012         BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
     055342  001026                         BNE     30$                     ;BR IF YES
     055344  004737  062142                 JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
     055350  012704  062440                 MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
     055354  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
     055360  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
     055364                                 FORCERROR       22$                     ;BBDFORCE ERROR IF FORCER=1
     055400  103407                         BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
     055402  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
     055404                                 NEXT,ERRNO
     055404                          22$:   ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     055404  104455                                                                     TRAP    C$ERDF
     055406  001476                                                                     .WORD   830
     055410  060240                                                                     .WORD   T192SSR
     055412  012046                                                                     .WORD   PKTSSR
     055414  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
     055420                          30$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055420  104406                                                                     TRAP    C$CLP1
                                     ;      Do WRITE CHARACTERISTICS to select reserved unit 7
     055422  012704  062270                 MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
     055426  004737  010662                 JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
     055432                                 FORCERROR       42$                     ;BBDFORCE ERROR IF FORCER=1
     055446  103407                         BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
     055450  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
     055452                                 NEXT,ERRNO
     055452                          42$:   ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     055452  104455                                                                     TRAP    C$ERDF
```

```
        055454  001477                                                          .WORD   831
        055456  060203                                                          .WORD   T19SSR
        055460  012046                                                          .WORD   PKTSSR
4531    055462  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
4532    055466                          50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        055466  104406                                                          TRAP    C$CLP1
4533
4534                                    ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4535    055470  012703  002752                  MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
4536    055474  012337  002312          100$:   MOV     (R3)+,DATA              ;GET A TEST PATTERN
4537    055500  042737  177400  002312          BIC     #^C<377>,DATA           ;DATA IS BYTE
4538    055506  010337  002316                  MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
4539                                    ;       Do a WRITE NPR to set loopback and tape direction OUT
4540    055512  012700  000100                  MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
4541    055516  052700  000040                  BIS     #NP.LOOP,R0             ;SET LOOPBACK
4542    055522  004737  062002                  JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
4543    055526  012704  062440                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4544    055532  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4545    055536  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4546    055542                                  FORCERROR       102$            ;@@DFORCE ERROR IF FORCER=1
4547    055556  103407                          BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
4548    055560  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4549    055562                                  NEXT.ERRNO
4550    055562                          102$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        055562  104455                                                          TRAP    C$ERDF
        055564  001500                                                          .WORD   832
        055566  060351                                                          .WORD   T194SSR
        055570  012046                                                          .WORD   PKTSSR
4551    055572  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
4552    055576                          105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        055576  104406                                                          TRAP    C$CLP1
4553                                    ;       Do a WRITE FORMAT to set IRESV3-->IWSTR = 1
4554    055600  012700  000002                  MOV     #WF.I3RES,R0            ;IRESV3-->IWSTR=1
4555    055604  004737  062122                  JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
4556    055610  012704  062440                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4557    055614  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4558    055620  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4559    055624                                  FORCERROR       112$            ;@@DFORCE ERROR IF FORCER=1
4560    055640  103407                          BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
4561    055642  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4562    055644                                  NEXT.ERRNO
4563    055644                          112$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
        055644  104455                                                          TRAP    C$ERDF
        055646  001501                                                          .WORD   833
        055650  060572                                                          .WORD   T198SSR
        055652  012046                                                          .WORD   PKTSSR
4564    055654  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
4565    055660                          120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        055660  104406                                                          TRAP    C$CLP1
4566                                    ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4567    055662  012700  000001                  MOV     #1,R0                   ;WRITE 1 BYTE
4568    055666  012701  002312                  MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
4569    055672  004737  062046                  JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
4570    055676  012704  062440                  MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4571    055702  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4572    055706  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4573    055712                                  FORCERROR       132$            ;@@DFORCE ERROR IF FORCER=1
```

```
4574 055726  103407                    BCS     140$                    ;BR IF CARRY SET (GOOD RETURN)
4575 055730  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4576 055732                            NEXT.ERRNO
4577 055732                    132$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     055732  104455                                                            TRAP    C$ERDF
     055734  001502                                                            .WORD   834
     055736  060414                                                            .WORD   T195SSR
     055740  012046                                                            .WORD   PKTSSR
4578 055742  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4579 055746                    140$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     055746  104406                                                            TRAP    C$CLP1
4580                           ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 0
4581 055750  005000            CLR     R0                              ;SET IRESV3==>IWSTR=0
4582 055752  004737  062122    JSR     PC,T19WFMT                      ;SETUP T9PK2 FOR WRITE FORMAT
4583 055756  012704  062440    MOV     #T19PK2,R4                      ;GET WRITE SUBSYSTEM COMMAND PACKET
4584 055762  010465  000000    MOV     R4,TSDB(R5)                     ;SET THE PACKET ADDRESS TO EXECUTE
4585 055766  004737  016336    JSR     PC,CHKTSSR                      ;WAIT FOR SSR TO SET
4586 055772                    FORCERROR       152$                            ;@@DFORCE ERROR IF FORCER=1
4587 056006  103407            BCS     160$                            ;BR IF CARRY SET (GOOD RETURN)
4588 056010  010001            MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
4589 056012                    NEXT.ERRNO
4590 056012                    152$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     056012  104455                                                            TRAP    C$ERDF
     056014  001503                                                            .WORD   835
     056016  060572                                                            .WORD   T198SSR
     056020  012046                                                            .WORD   PKTSSR
4591 056022  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4592 056026                    160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056026  104406                                                            TRAP    C$CLP1
4593                           ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
4594 056030  012700  000002            MOV     #WF.I3RES,R0            ;IRESV3==>IWSTR=1
4595 056034  004737  062122    JSR     PC,T19WFMT                      ;SETUP T9PK2 FOR WRITE FORMAT
4596 056040  012704  062440    MOV     #T19PK2,R4                      ;GET WRITE SUBSYSTEM COMMAND PACKET
4597 056044  010465  000000    MOV     R4,TSDB(R5)                     ;SET THE PACKET ADDRESS TO EXECUTE
4598 056050  004737  016336    JSR     PC,CHKTSSR                      ;WAIT FOR SSR TO SET
4599 056054                    FORCERROR       172$                            ;@@DFORCE ERROR IF FORCER=1
4600 056070  103407            BCS     180$                            ;BR IF CARRY SET (GOOD RETURN)
4601 056072  010001            MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
4602 056074                    NEXT.ERRNO
4603 056074                    172$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     056074  104455                                                            TRAP    C$ERDF
     056076  001504                                                            .WORD   836
     056100  060572                                                            .WORD   T198SSR
     056102  012046                                                            .WORD   PKTSSR
4604 056104  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4605 056110                    180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056110  104406                                                            TRAP    C$CLP1
4606
4607                           ;       Do a WRITE NPR to set loopback and tape direction IN
4608 056112  005000            CLR     R0                              ;CLR NP.OUT TO SET TAPE DIRECTION IN
4609 056114  052700  000040    BIS     #NP.LOOP,R0                     ;SET LOOPBACK
4610 056120  004737  062002    JSR     PC,T19SNPR                      ;SETUP T19PK2 FOR WRITE NPR
4611 056124  012704  062440    MOV     #T19PK2,R4                      ;GET WRITE SUBSYSTEM COMMAND PACKET
4612 056130  010465  000000    MOV     R4,TSDB(R5)                     ;SET THE PACKET ADDRESS TO EXECUTE
4613 056134  004737  016336    JSR     PC,CHKTSSR                      ;WAIT FOR SSR TO SET
4614 056140                    FORCERROR       182$                            ;@@DFORCE ERROR IF FORCER=1
4615 056154  103407            BCS     190$                            ;BR IF CARRY SET (GOOD RETURN)
```

```
4616 056156  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4617 056160                             NEXT.ERRNO
4618 056160                     182$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     056160  104455                                                             TRAP    C$ERDF
     056162  001505                                                             .WORD   837
     056164  060351                                                             .WORD   T194SSR
     056166  012046                                                             .WORD   PKTSSR
4619 056170  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4620 056174                     190$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056174  104406                                                             TRAP    C$CLP1
4621                            ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
4622 056176  012700  000001            MOV     #1,R0                   ;WRITE 1 BYTE
4623 056202  012701  002312            MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
4624 056206  004737  062046            JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
4625 056212  012704  062440            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4626 056216  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4627 056222  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4628 056226                             FORCERROR       202$                    ;BBDFORCE ERROR IF FORCER=1
4629 056242  103407                    BCS     210$                    ;BR IF CARRY SET (GOOD RETURN)
4630 056244  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4631 056246                             NEXT.ERRNO
4632 056246                     202$:   ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     056246  104455                                                             TRAP    C$ERDF
     056250  001506                                                             .WORD   838
     056252  060414                                                             .WORD   T195SSR
     056254  012046                                                             .WORD   PKTSSR
4633 056256  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4634 056262                     210$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056262  104406                                                             TRAP    C$CLP1
4635                            ;       Do a READ FIFO with tape direction IN to read data
4636 056264  012700  000001            MOV     #1,R0                   ;SET READ BYTE COUNT
4637 056270  004737  062026            JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
4638 056274  012704  062440            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4639 056300  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4640 056304  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4641 056310                             FORCERROR       222$                    ;BBDFORCE ERROR IF FORCER=1
4642 056324  103407                    BCS     230$                    ;BR IF CARRY SET (GOOD RETURN)
4643 056326  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4644 056330                             NEXT.ERRNO
4645 056330                     222$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     056330  104455                                                             TRAP    C$ERDF
     056332  001507                                                             .WORD   839
     056334  060460                                                             .WORD   T196SSR
     056336  012046                                                             .WORD   PKTSSR
4646 056340  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4647 056344                     230$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056344  104406                                                             TRAP    C$CLP1
4648                            ;       If Data read from FIFO NOT= to Data sent Then Print Error
4649 056346  004737  062200            JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4650 056352  012701  060042            MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4651 056356  012702  062332            MOV     #T19BFSTA,R2            ;GET RECV READ STATUS
4652 056362  013711  002312            MOV     DATA,(R1)               ;SET EXPD WORD #8 = DATA
4653 056366  016261  000002  000002    MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
4654 056374  005000                    CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4655 056376  012701  062312            MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4656 056402  012702  060022            MOV     #T19EXP,R2              ;EXPD ADDRESS
4657 056406  012703  000022            MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
```

```
4658 056412  004737  011500                JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4659 056416                                FORCERROR       242$,NOTSSR     ;SSD
4660 056426  103404                        BCS     250$                    ;BR IF YES
4661 056430                                NEXT.ERRNO
4662 056430                        242$:   ERPHRD  ERRNO,T19WSTR,MSGSUB    ;REPORT ERROR
     056430  104456                                                        TRAP    C$ERHRD
     056432  001510                                                        .WORD   840
     056434  061503                                                        .WORD   T19WSTR
     056436  013742                                                        .WORD   MSGSUB
4663 056440                        250$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     056440  104406                                                        TRAP    C$CLP1
4664
4665
4666 056442                                FORCEXIT        255$            ;SSD
4667 056452  013703  002316                MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
4668 056456  020327  003062                CMP     R3,#TBLEND              ;END OF TSTBLK?
4669 056462  103002                        BHIS    255$                    ;BR IF YES
4670 056464  000137  055474                JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
4671 056470                        255$:
4672
4673 056470                                ENDSUB                          ;////////////// END SUBTEST //////////
     056470                                                                L10071:
     056470  104403                                                        TRAP    C$ESUB
4674
4675 056472  005737  002222                TST     FATFLG                  ;ANY FATAL ERRORS ?
4676 056476  001402                        BEQ     260$                    ;BRANCH IF NOT
4677 056500  004737  017202                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4678 056504                        260$:
4679                                       .SBTTL  TEST  8: SUBTEST 4 LOOPBACK READ STROBE TEST
4680
4681                                ;++
4682                                ; TEST 8: SUBTEST 4:
4683                                ;
4684                                ; SUBTEST DESCRIPTION:
4685                                ;
4686                                ;       This subtest verifies the Read Strobe loopback path
4687                                ;       can strobe the data from the Data lines to the FIFO.
4688                                ;       The signal IRESV4 drives IRSTR (read strobe) to write
4689                                ;       from the data lines to the FIFO.
4690                                ;
4691                                ; TEST STEPS:
4692                                ;
4693                                ;
4694                                ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4695                                ;    BEGIN
4696                                ;       Write to TSSR register to soft initialize the controller
4697                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
4698                                ;       If Extended Features Hardware Switch Clear then:
4699                                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
4700                                ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
4701                                ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
4702                                ;       Do a WRITE NPR to set loopback and tape direction OUT
4703                                ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
4704                                ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4705                                ;       Do a READ FIFO with tape direction OUT to load tape out write latch
4706                                ;       Do a WRITE NPR to set loopback and tape direction IN
4707                                ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 to write loop data to FIFO
```

```
                                ;       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
                                ;
056504                          ;       BGNSUB                          ;////////////// BEGIN SUBTEST //////////
056504                                                                                    T8.4:
056504  104402                                                                                  TRAP    C$BSUB
                                ;       Write to TSSR register to soft initialize the controller
056506                          5$:
056506  004737  015774          ;       JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
056512  103405                  ;       BCS     10$                     ;BR IF SOFT INIT OKAY
056514  010001                  ;       MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
056516                          ;       ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
056516  104455                                                                                  TRAP    C$ERDF
056520  001510                                                                                  .WORD   840
056522  003652                                                                                  .WORD   SFIERR
056524  012034                                                                                  .WORD   SFIMSG
                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
056526  005037  002222          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
056532  012704  062270          ;       MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
056536  004737  010662          ;       JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
056542                          ;       FORCERROR       12$                     ;@@FORCE ERROR IF FORCER=1
056556  103407                  ;       BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
056560  010001                  ;       MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
056562                          ;       NEXT.ERRNO
056562                          12$.    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
056562  104455                                                                                  TRAP    C$ERDF
056564  001511                                                                                  .WORD   841
056566  060203                                                                                  .WORD   T19SSR
056570  012046                                                                                  .WORD   PKTSSR
056572  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
056576                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
056576  104406                                                                                  TRAP    C$CLP1
                                ;       If Extended Features Hardware Switch Clear then:
                                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
056600  012701  062312          ;       MOV     #T19BFR,R1              ;MESSAGE BUFFER ADDRESS
056604  032761  000200  000012  ;       BIT     #X2.EXTF,XST2(R1)       ;EXTENDED FEATURES SWITCH SET?
056612  001026                  ;       BNE     30$                     ;BR IF YES
056614  004737  062142          ;       JSR     PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
056620  012704  062440          ;       MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
056624  010465  000000          ;       MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
056630  004737  016336          ;       JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
056634                          ;       FORCERROR       22$                     ;@@FORCE ERROR IF FORCER=1
056650  103407                  ;       BCS     30$                     ;BR IF CARRY SET (GOOD RETURN)
056652  010001                  ;       MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
056654                          ;       NEXT.ERRNO
056654                          22$:    ERRDF   ERRNO,T192SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
056654  104455                                                                                  TRAP    C$ERDF
056656  001512                                                                                  .WORD   842
056660  060240                                                                                  .WORD   T192SSR
056662  012046                                                                                  .WORD   PKTSSR
056664  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
056670                          30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
056670  104406                                                                                  TRAP    C$CLP1
                                ;       Do WRITE CHARACTERISTICS to select reserved unit 7
```

```
 4749 056672  012704  062270                MOV     #T19PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 4750 056676  004737  010662                JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
 4751 056702                                FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
 4752 056716  103407                        BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
 4753 056720  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4754 056722                                NEXT.ERRNO
 4755 056722                        42$:    ERRDF   ERRNO,T19SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      056722  104455                                                                TRAP    C$ERDF
      056724  001513                                                                .WORD   843
      056726  060203                                                                .WORD   T19SSR
      056730  012046                                                                .WORD   PKTSSR
 4756 056732  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 4757 056736                        50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      056736  104406                                                                TRAP    C$CLP1
 4758
 4759                               ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
 4760 056740  012703  002752                MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
 4761 056744  012337  002312        100$:   MOV     (R3)+,DATA              ;GET A TEST PATTERN
 4762 056750  042737  177400  002312        BIC     #^C<377>,DATA           ;DATA IS BYTE
 4763 056756  010337  002316                MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
 4764                               ;       Do a WRITE NPR to set loopback and tape direction OUT
 4765 056762  012700  000100                MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
 4766 056766  052700  000040                BIS     #NP.LOOP,R0             ;SET LOOPBACK
 4767 056772  004737  062002                JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
 4768 056776  012704  062440                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 4769 057002  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 4770 057006  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 4771 057012                                FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
 4772 057026  103407                        BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
 4773 057030  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4774 057032                                NEXT.ERRNO
 4775 057032                        102$:   ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      057032  104455                                                                TRAP    C$ERDF
      057034  001514                                                                .WORD   844
      057036  060351                                                                .WORD   T194SSR
      057040  012046                                                                .WORD   PKTSSR
 4776 057042  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 4777 057046                        105$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      057046  104406                                                                TRAP    C$CLP1
 4778                               ;       Do a WRITE FORMAT to set IRESV4-->IRSTR = 1
 4779 057050  012700  000001                MOV     #WF.I4RES,R0            ;IRESV4-->IRSTR=1
 4780 057054  004737  062122                JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
 4781 057060  012704  062440                MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
 4782 057064  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 4783 057070  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 4784 057074                                FORCERROR       112$                    ;@@DFORCE ERROR IF FORCER=1
 4785 057110  103407                        BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
 4786 057112  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4787 057114                                NEXT.ERRNO
 4788 057114                        112$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      057114  104455                                                                TRAP    C$ERDF
      057116  001515                                                                .WORD   845
      057120  060572                                                                .WORD   T198SSR
      057122  012046                                                                .WORD   PKTSSR
 4789 057124  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
 4790 057130                        120$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      057130  104406                                                                TRAP    C$CLP1
```

```
4791                                    ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4792 057132  012700  000001                    MOV     #1,R0                   ;WRITE 1 BYTE
4793 057136  012701  002312                    MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
4794 057142  004737  062046                    JSR     PC,T19WFIF              ;SETUP T19PK2 FOR WRITE FIFO
4795 057146  012704  062440                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4796 057152  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4797 057156  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4798 057162                                    FORCERROR       132$                    ;@@FORCE ERROR IF FORCER=1
4799 057176  103407                            BCS     140$                    ;BR IF CARRY SET (GOOD RETURN)
4800 057200  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4801 057202                                    NEXT.ERRNO
4802 057202                             132$:  ERRDF   ERRNO,T195SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057202  104455                                                                 TRAP    C$ERDF
     057204  001516                                                                 .WORD   846
     057206  060414                                                                 .WORD   T195SSR
     057210  012046                                                                 .WORD   PKTSSR
4803 057212  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4804 057216                             140$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057216  104406                                                                 TRAP    C$CLP1
4805                                    ;       Do a READ FIFO with tape direction OUT to load tape out write latch
4806 057220  012700  000001                    MOV     #1,R0                   ;SET READ BYTE COUNT
4807 057224  004737  062026                    JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
4808 057230  012704  062440                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4809 057234  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4810 057240  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4811 057244                                    FORCERROR       152$                    ;@@FORCE ERROR IF FORCER=1
4812 057260  103407                            BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
4813 057262  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4814 057264                                    NEXT.ERRNO
4815 057264                             152$:  ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057264  104455                                                                 TRAP    C$ERDF
     057266  001517                                                                 .WORD   847
     057270  060460                                                                 .WORD   T196SSR
     057272  012046                                                                 .WORD   PKTSSR
4816 057274  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4817 057300                             160$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057300  104406                                                                 TRAP    C$CLP1
4818                                    ;       Do a WRITE NPR to set loopback and tape direction IN
4819 057302  005000                            CLR     R0                      ;CLR NP.OUT TO SET TAPE DIRECTION IN
4820 057304  052700  000040                    BIS     #NP.LOOP,R0             ;SET LOOPBACK
4821 057310  004737  062002                    JSR     PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
4822 057314  012704  062440                    MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4823 057320  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4824 057324  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4825 057330                                    FORCERROR       182$                    ;@@FORCE ERROR IF FORCER=1
4826 057344  103407                            BCS     190$                    ;BR IF CARRY SET (GOOD RETURN)
4827 057346  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4828 057350                                    NEXT.ERRNO
4829 057350                             182$:  ERRDF   ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057350  104455                                                                 TRAP    C$ERDF
     057352  001520                                                                 .WORD   848
     057354  060351                                                                 .WORD   T194SSR
     057356  012046                                                                 .WORD   PKTSSR
4830 057360  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
4831 057364                             190$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057364  104406                                                                 TRAP    C$CLP1
4832                                    ;       Do a WRITE FORMAT to set IRESV4==>IRSTR = 0
```

```
4833 057366  005000                    CLR     R0                      ;SET IRESV4-->IRSTR=0
4834 057370  004737  062122            JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
4835 057374  012704  062440            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4836 057400  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4837 057404  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4838 057410                            FORCERROR       202$                    ;@@DFORCE ERROR IF FORCER=1
4839 057424  103407                    BCS     210$                    ;BR IF CARRY SET (GOOD RETURN)
4840 057426  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4841 057430                            NEXT.ERRNO
4842 057430                    202$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057430  104455                                                            TRAP    C$ERDF
     057432  001521                                                            .WORD   849
     057434  060572                                                            .WORD   T198SSR
     057436  012046                                                            .WORD   PKTSSR
4843 057440  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4844 057444                    210$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057444  104406                                                            TRAP    C$CLP1
4845                           ;       Do a WRITE FORMAT to set IRESV4-->IRSTR = 1
4846 057446  01270C  000001            MOV     #WF.I4RES,R0            ;IRESV4-->IRSTR=1
4847 057452  004737  062122            JSR     PC,T19WFMT              ;SETUP T9PK2 FOR WRITE FORMAT
4848 057456  012704  062440            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4849 057462  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4850 057466  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4851 057472                            FORCERROR       222$                    ;@@DFORCE ERROR IF FORCER=1
4852 057506  103407                    BCS     230$                    ;BR IF CARRY SET (GOOD RETURN)
4853 057510  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4854 057512                            NEXT.ERRNO
4855 057512                    222$:   ERRDF   ERRNO,T198SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057512  104455                                                            TRAP    C$ERDF
     057514  001522                                                            .WORD   850
     057516  060572                                                            .WORD   T198SSR
     057520  012046                                                            .WORD   PKTSSR
4856 057522  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4857 057526                    230$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057526  104406                                                            TRAP    C$CLP1
4858                           ;       Do a READ FIFO with tape direction IN to read data
4859 057530  012700  000001            MOV     #1,R0                   ;SET READ BYTE COUNT
4860 057534  004737  062026            JSR     PC,T19RFIF              ;SETUP T19PK2 FOR READ FIFO
4861 057540  012704  062440            MOV     #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4862 057544  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
4863 057550  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
4864 057554                            FORCERROR       282$                    ;@@DFORCE ERROR IF FORCER=1
4865 057570  103407                    BCS     290$                    ;BR IF CARRY SET (GOOD RETURN)
4866 057572  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4867 057574                            NEXT.ERRNO
4868 057574                    282$:   ERRDF   ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     057574  104455                                                            TRAP    C$ERDF
     057576  001523                                                            .WORD   851
     057600  060460                                                            .WORD   T196SSR
     057602  012046                                                            .WORD   PKTSSR
4869 057604  005237  002222            INC     FATFLG                  ;SET FATAL ERROR FLAG
4870 057610                    290$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     057610  104406                                                            TRAP    C$CLP1
4871                           ;       If Data read from FIFO NOT= to Data sent Then Print Error
4872 057612  004737  062200            JSR     PC,T19SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4873 057616  012701  060042            MOV     #T19EXSTA,R1            ;GET EXPECTED READ STATUS
4874 057622  012702  062332            MOV     #T198FSTA,R2            ;GET RECV READ STATUS
```

```
4875 057626  013711  002312            MOV     DATA,(R1)               ;SET EXPD WORD #8 = DATA
4876 057632  016261  000002  000002    MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTING)
4877 057640  005000                    CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
4878 057642  012701  062312            MOV     #T19BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
4879 057646  012702  060022            MOV     #T19EXP,R2              ;EXPD ADDRESS
4880 057652  012703  000022            MOV     #18.,R3                 ;NUMBER OF BYTES TO COMPARE
4881 057656  004737  011500            JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
4882 057662                            FORCERROR       302$,NOTSSR     ;@@D
4883 057672  103404                    BCS     310$                    ;BR IF YES
4884 057674                            NEXT.ERRNO
4885 057674                    302$:   ERRHRD  ERRNO,T19RSTR,MSGSUB    ;REPORT ERROR
     057674  104456                                                            TRAP    C$ERHRD
     057676  001524                                                            .WORD   852
     057700  061610                                                            .WORD   T19RSTR
     057702  013742                                                            .WORD   MSGSUB
4886 057704                    310$:   CK_OOP                          ;LOOP ON ERROR, IF FLAG SET
     057704  104406                                                            TRAP    C$CLP1
4887
4888
4889 057706                            FORCEXIT        355$            ;@@D
4890 057716  013703  002316            MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
4891 057722  020327  003062            CMP     R3,#TBLEND              ;END OF TSTBLK?
4892 057726  103002                    BHIS    355$                    ;BR IF YES
4893 057730  000137  056744            JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
4894 057734                    355$:
4895
4896 057734                            ENDSUB                          ;////////////// END SUBTEST //////////
     057734                                                                    L10072:
     057734  104403                                                            TRAP    C$ESUB
4897
4898 057736  005737  002222            TST     FATFLG                  ;ANY FATAL ERRORS ?
4899 057742  001402                    BEQ     360$                    ;BRANCH IF NOT
4900 057744  004737  017202            JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4901 057750                    360$:
4902
4903 057750                            EXIT    TST                     ;////////// EXIT TEST //////////
     057750  104432                                                            TRAP    C$EXIT
     057752  002602                                                            .WORD   L10066 .
4904
4905
4906                           ;+
4907                           ;LOCAL STORAGE FOR THIS TEST
4908                           ;-
4909
4910 057754  000000            T19PREV:        .WORD   0               ;DRIVE SIGNAL 1-0 TRANSITION FLAG
4911                           ;+
4912                           ; LOOPBACK DRIVE SIGNAL TABLE
4913                           ; THIS TABLE IS USED BY T19CNVT TO SETUP
4914                           ; A DRIVE PATTERN FROM THE TEST DATA INPUT PATTERN.
4915                           ;
4916                           ; WRITE CONTROL SIGNALS ARE OF FORM WC.XXX
4917                           ; WRITE FORMAT SIGNALS ARE OF FORM WF.XXXX
4918                           ;-
4919 057756                    T19BFCTL:                               ;WRITE CONTROL DRIVE SIGNALS
4920 057756  000001                    WC.IGO                          ;IGO==>IFPT     DATA<0>
4921 057760  000002                    WC.IFEN                         ;IFEN==>IFBY    DATA<1>
4922 057762  000004                    WC.IRWU                         ;IRWU==>IRWD    DATA<2>
```

```
4923 057764  000010                        WC.IREW          ;IREW-->IDBY    DATA<3>
4924 057766  002000                        WF.IERASE*256.   ;IFAD-->ILDP    DATA<4>
4925 057770  000040                        WC.I1TAD         ;ITAD1-->IONL   DATA<5>
4926 057772  000100                        WC.IOTAD         ;ITAD0-->IRDY   DATA<6>
4927 057774  000200                        WC.IFAD          ;IERASE-->ISPEED DATA<7>
4928 057776  004000                        WF.IEDIT*256.    ;IEDIT-->IHER   DATA<8>
4929 060000  010000                        WF.IWFM*256.     ;IWFM-->IFMK    DATA<9>
4930 060002  020000                        WF.IREV*256.     ;IREV-->ICER    DATA<10>
4931 060004  040000                        WF.IWRT*256.     ;IWRT-->IIDENT  DATA<11>
4932 060006  100000                        WF.IHISP*256.    ;IHISP-->IEOT   DATA<12>
4933 060010  000000                        .WORD   0        ;IRESV2 (UNUSED)DATA<13>
4934 060012  000000                        .WORD   0        ;IRESV1 (UNUSED)DATA<14>
4935 060014  000000                        .WORD   0        ;PARERR (UNTESTED)DATA<15>
4936
4937 060016                        T19MSK:                  ;MASK OF UNTESTED BITS IN READ STATUS BYTES
4938                                                         ;UNTESTED BITS ARE SET TO 1
4939 060016      377                       .BYTE   ^C<000>  ;BYTE 0 MASK
4940 060017      037                       .BYTE   ^C<340>  ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
4941 060020      360                       .BYTE   ^C<017>  ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
4942 060021      000                       .BYTE   0        ;MAKE IT EVEN
4943
4944 060022                        T19EXP:                  ;BEGIN EXPECTED DATA BUFFER
4945 060022  000000                        .WORD   0        ;MESSAGE TYPE
4946 060024  000000                        .WORD   0        ;DATA FIELD LENGTH
4947 060026  000000                        .WORD   0        ;RBPCR
4948 060030  000000                        .WORD   0        ;XST0
4949 060032  000000                        .WORD   0        ;XST1
4950 060034  000000                        .WORD   0        ;XST2
4951 060036  000000                        .WORD   0        ;XST3
4952 060040  000000                        .WORD   0        ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
4953 060042                        T19EXSTA: .BLKB 64.      ;EXPECTED READ STATUS AND WRITE FIFO DATA
4954 060142                        T19EXEND:                ;END EXPECTED DATA BUFFER
4955                                                         ;*
4956                                                         ;LOCAL TEXT MESSAGES FOR TEST
4957                                                         ;-
4958
4959 060142  124  162  141  TST19ID:       .ASCIZ  'Transport Bus Interface Loopback
4960 060203  127  122  111  T19SSR:  .ASCIZ  'WRITE CHARACTERISTICS Failed'
4961 060240  127  122  111  T192SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
4962 060304  127  122  111  T193SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
4963 060351  127  122  111  T194SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
4964 060414  127  122  111  T195SSR:.ASCIZ  'WRITE SUBSYSTEM (Write FIFO) Failed'
4965 060460  127  122  111  T196SSR:.ASCIZ  'WRITE SUBSYSTEM (Read FIFO) Failed'
4966 060523  127  122  111  T197SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Control) Failed'
4967 060572  127  122  111  T198SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Format) Failed'
4968 060640  106  111  106  T191CMP:.ASCIZ  'FIFO Status in WORD #9 Incorrect after Initialize'
4969 060722  122  145  141  T192CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
4970 061016  124  141  160  T193CMP:.ASCIZ  'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
4971 061104  122  145  141  T195CMP:.ASCIZ  'Read FIFO Data not equal to Write FIFO Data'
4972 061160  106  111  106  T196CMP:.ASCIZ  'FIFO Status (in WORD #9) Incorrect after READ FIFO'
4973 061243  124  141  160  T197CMP:.ASCIZ  'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
4974 061331  103  157  156  T198CMP:.ASCIZ  'Control Signal Loopback Data Error, Data is in WORD #8'
4975 061420  122  145  141  T199CMP:.ASCIZ  'Read/Write Loopback Data Error, Data is in WORD #8'
4976 061503  114  157  157  T19WSTR:.ASCIZ  'Loopback Data Error when strobed by Write strobe, Data is in WORD #8'
4977 061610  114  157  157  T19RSTR:.ASCIZ  'Loopback Data Error when strobed by Read Strobe, Data is in WORD #8'
4978
4979                                .EVEN
```

```
                                        ;*
                                        ; CLEAR MESSAGE BUFFER
                                        ;
                                        T19CLRBUF:                              ;
                                                SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
061714 012701 062312                            MOV     #T19BFR,R1              ;GET MESSAGE BUFFER ADDRESS
061720 012702 000120                            MOV     #T19BEND-T19BFR,R2      ;SIZE OF MESSAGE BUFFER IN BYTES
061724 105021                           10$:    CLRB    (R1)+                   ;CLEAR A BYTE
061730 005302                                   DEC     R2                      ;DONE?
061732 003375                                   BGT     10$                     ;BR IF NO
061734 000207                                   RTS     PC                      ;RETURN

                                        ;*
                                        ; SETUP T19PK2 PACKET FOR READ STATUS
                                        ;
                                        T19SRD:
061736 004737 061714                            JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
061742 012700 062450                            MOV     #T19DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
061746 112720 000005                            MOVB    #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
061752 105010                                   CLRB    (R0)                    ;CLEAR BSEL1
061754 000207                                   RTS     PC                      ;RETURN

                                        ;*
                                        ; SETUP T19PK2 PACKET FOR WRITE MISC Reset Tape Status F-FLOPS
                                        ;
                                        T19RSFIF:
061760 004737 061714                            JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
061764 012700 062450                            MOV     #T19DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
061770 112720 000010                            MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
061774 112710 000030                            MOVB    #MS.RSFIF!MS.RSTAP,(R0) ;STORE BSEL1 CLEAR FIFO CODES
062000 000207                                   RTS     PC                      ;RETURN

                                        ;*
                                        ; SETUP T19PK2 PACKET FOR WRITE NPR
                                        ;
                                        ; INPUT:
                                        ;       R0 CONTAINS BSEL1 NPR DATA
                                        ;
                                        ;       SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
                                        ;
                                        T19SNPR:
062002 004737 061714                            JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
062006 012701 062450                            MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
062012 112721 000011                            MOVB    #PW.WNPR,(R1)+          ;STORE WRITE NPR IN BSEL0
062016 052700 000020                            BIS     #NP.WRP,R0              ;DON'T WRITE WRONG PARITY
062022 110011                                   MOVB    R0,(R1)                 ;STORE NPR DATA IN BSEL1
062024 000207                                   RTS     PC                      ;RETURN

                                        ;*
                                        ; SETUP T19PK2 PACKET FOR READ FIFO
                                        ;
                                        ; INPUT:
                                        ;       R0 CONTAINS SEL2 BYTE COUNT
                                        ;
                                        T19RFIF:
062026 004737 061714                            JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
```

```
5037 062032  012701  062450                MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5038 062036  112721  000003                MOVB    #PW.RFIFO,(R1)+         ;STORE READ FIFO IN BSEL0
5039 062042  110021                        MOVB    R0,(R1)+                ;STORE BYTE COUNT IN BSEL1
5040 062044  000207                        RTS     PC                      ;RETURN
5041                                ;+
5042                                ; SETUP T19PK2 PACKET FOR WRITE FIFO
5043                                ;
5044                                ; INPUT:
5045                                ;       R0 CONTAINS BYTE COUNT
5046                                ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
5047                                ;
5048 062046                        T19WFIF:
5049 062046                                SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
5050 062052  004737  061714                JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
5051 062056  012702  062450                MOV     #T19DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
5052 062062  112722  000004                MOVB    #PW.WFIFO,(R2)+         ;STORE WRITE FIFO IN BSEL0
5053 062066  110022                        MOVB    R0,(R2)+                ;STORE BYTE COUNT IN BSEL1
5054 062070  005022                        CLR     (R2)+                   ;CLEAR SEL2 (UNUSED)
5055 062072  112122                10$:    MOVB    (R1)+,(R2)+             ;STORE DATA PATTERN BYTE
5056 062074  005300                        DEC     R0                      ;DONE ALL BYTES?
5057 062076  003375                        BGT     10$                     ;BR IF NO
5058 062100  000207                        RTS     PC                      ;RETURN
5059                                ;+
5060                                ; SETUP T19PK2 FOR WRITE CONTROL
5061                                ;
5062                                ; INPUT:
5063                                ;       R0 CONTAINS DRIVING DATA PATTERN
5064                                ;-
5065 062102                        T19WCTL:
5066 062102  004737  061714                JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
5067 062106  012701  062450                MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5068 062112  112721  000006                MOVB    #PW.WCTL,(R1)+          ;STORE WRITE CONTROL IN BSEL0
5069 062116  110021                        MOVB    R0,(R1)+                ;STORE DATA WORD IN BSEL1
5070 062120  000207                        RTS     PC                      ;RETURN
5071                                ;+
5072                                ; SETUP T19PK2 FOR WRITE FORMAT TRANSPORT REGISTER
5073                                ;
5074                                ; INPUT:
5075                                ;       R0 CONTAINS DRIVING DATA PATTERN
5076                                ;-
5077 062122                        T19WFMT:
5078 062122  004737  061714                JSR     PC,T19CLRBUF            ;CLEAR MESSAGE BUFFER
5079 062126  012701  062450                MOV     #T19DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5080 062132  112721  000007                MOVB    #PW.WFMT,(R1)+          ;STORE WRITE FORMAT IN BSEL0
5081 062136  110021                        MOVB    R0,(R1)+                ;STORE DATA WORD IN BSEL1
5082 062140  000207                        RTS     PC                      ;RETURN
5083                                ;+
5084                                ; SETUP T19PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
5085                                ;-
5086 062142                        T19SEXT:
5087 062142  012700  062450                MOV     #T19DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
5088 062146  112720  000010                MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
5089 062152  112710  000200                MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
5090 062156  000207                        RTS     PC                      ;RETURN
5091                                ;+
5092                                ; CLEAR EXPECTED DATA MESSAGE BUFFER
5093                                ;-
```

```
                                T19CLEXP:
                                        MOV     #T19EXP,R1              ;GET EXPD ADDRESS
                                        MOV     #T19EXEND-T19EXP,R0     ;GET EXPD SIZE
                                10$:    CLRB    (R1)+                   ;CLEAR A BYTE
                                        DEC     R0                      ;DONE?
                                        BGT     10$                     ;BR IF NO
                                        RTS     PC                      ;RETURN

                                ;+
                                ;Set WORDS 0-7 of expd message BUFFER = to recv since not testing
                                ;-
                                T19SETEXP:
                                        MOV     #T19EXP,R2              ;GET EXPD
                                        MOV     #T19BFR,R3              ;GET READ STATUS RECV BUFFER
                                        MOV     #8.,R0                  ;SET WORDS 0-7 EXP=RECV
                                5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
                                        DEC     R0                      ;DONE WORDS 0-7 WORDS?
                                        BGT     5$                      ;BR IF NO
                                        RTS     PC                      ;RETURN

                                ;+
                                ; CONVERT A TEST PATTERN DATA WORD TO LOOPBACK DRIVE SIGNALS
                                ;
                                ; INPUTS:
                                ;
                                ;       R0      TEST PATTERN
                                ;
                                ; IMPLICIT INPUTS:
                                ;
                                ;       T19BFCTL  - CONTAINS WRITE CONTROL / WRITE FORMAT CONVERSION BITS
                                ;
                                ; OUTPUTS:
                                ;
                                ;       R0      - LOW BYTE CONTAINS WRITE CONTROL DATA
                                ;               - HIGH BYTE CONTAINS WRITE FORMAT DATA
                                ;-
                                T19CNVT:
                                        SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
                                        MOV     #T19BFCTL,R1            ;CONVERSION TABLE ADDRESS
                                        CLR     R2                      ;INIT RESULT OF CONVERSION
                                        MOV     #16.,R3                 ;BIT COUNT
                                10$:    ROR     R0                      ;IS THIS BIT EQUAL TO 1?
                                        BCC     20$                     ;BR IF NO
                                        BIS     (R1),R2                 ;SET CONVERTED BIT
                                20$:    TST     (R1)+                   ;POINT TO NEXT BIT IN CONVERSION TABLE
                                        DEC     R3                      ;DONE?
                                        BGT     10$                     ;BR IF NO
                                        MOV     R2,R0                   ;COPY RESULT
                                        RTS     PC                      ;RETURN


                                .=<.+10>&177770
                                ;
                                ;WRITE CHARACTERISTICS COMMAND PACKET
                                ;
                                T19PACKET:                              ;COMMAND PACKET FOR TEST
```

```
5153 062270  100004                    .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
5154 062272  062300                    .WORD   T19DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
5155 062274  000000                    .WORD   0
5156 062276  000012                    .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
5157
5158 062300                    T19DATA:                                ;CHARACTERISTICS DATA BLOCK
5159 062300  062312                    .WORD   T19BFR                  ;ADDRESS OF MESSAGE BUFFER
5160 062302  000000                    .WORD   0
5161 062304  000024                    .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
5162 062306  000000                    .WORD   0                       ;ESS,ENB,EAI,ERI
5163 062310  000007                    .WORD   7                       ;EXTENDED FEATURES UNIT NO.
5164
5165
5166                           ;MESSAGE BUFFER FOR ALL TEST 8 COMMANDS
5167
5168 062312                    T19BFR:                                 ;BEGIN MESSAGE BUFFER
5169 062312  000000                    .WORD   0                       ;MESSAGE TYPE
5170 062314  000000                    .WORD   0                       ;DATA FIELD LENGTH
5171 062316  000000                    .WORD   0                       ;RBPCR
5172 062320  000000                    .WORD   0                       ;XST0
5173 062322  000000                    .WORD   0                       ;XST1
5174 062324  000000                    .WORD   0                       ;XST2
5175 062326  000000                    .WORD   0                       ;XST3
5176 062330  000000                    .WORD   0                       ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
5177 062332                    T19BFSTA: .BLKB 64.                     ;READ STATUS AND WRITE FIFO BUFFER
5178 062432                    T19BEND:                                ;END OF MESSAGE BUFFER
5179                           ;
5180                           ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
5181                           ;
5183         062440                    .=<.+10>&177770
5185 062440                    T19PK2:
5186 062440  100006                    .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
5187 062442  062450                    .WORD   T19DT2                  ;LOW ADDRESS OF DATA BLOCK
5188 062444  000000                    .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
5189 062446  000012                    .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
5190
5191 062450                    T19DT2:                                 ;DATA BLOCK
5192 062450     000                    .BYTE   0                       ;BSEL0
5193 062451     000                    .BYTE   0                       ;BSEL1
5194 062452  000000                    .WORD   0                       ;SEL2
5195 062454                            .BLKB   64.                     ;WRITE FIFO DATA OUTPUT BUFFER
5196
5197
5198 062554                            ENDTST
     062554                                                                    L10066:
     062554  104401                                                                      TRAP    C$ETST
5199                                   .SBTTL  TEST  9: READ/WRITE DATA PARITY TEST
5200                           ;**
5201                           ; TEST DESCRIPTION:
5202                           ;
5203                           ;       This test verifies that the Write Data Parity generator
5204                           ;       and the Read Data Parity checker operate properly. The
5205                           ;       Transport Bus signal loopback mode is enabled and a
5206                           ;       Set Wrong parity function is executed. Then various
5207                           ;       Write Subsystem Memory functions are performed to
5208                           ;       write data to and from the FIFO in loopback mode.
5209                           ;       The program then checks to insure a Read Data parity
```

```
                        ;       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
                        ;--
                        
062556                          BGNTST
062556                                                                  T9::
062556  012700  065142          MOV     #TST20ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
062562  004737  016510          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
062566  012737  000012  002216  MOV     #10.,LOOPCNT            ;PERFORM 10 ITERATIONS
062574                  T20LOOP:
```

```
                                        BGNSUB                          ;////////////// BEGIN SUBTEST //////////
                                                                                  T9.1:
                                                                                        TRAP    C$BSUB
                                ;       Write to TSSR register to soft initialize the controller
                                5$:
                                        JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
                                        BCS     10$                     ;BR IF SOFT INIT OKAY
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
                                                                                        TRAP    C$ERDF
                                                                                        .WORD   900
                                                                                        .WORD   SFIERR
                                                                                        .WORD   SFIMSG
                                ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
                                10$:    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$:    ERRDF   ERRNO,T20SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                                                                                        TRAP    C$ERDF
                                                                                        .WORD   901
                                                                                        .WORD   T20SSR
                                                                                        .WORD   PKTSSR
                                        INC     FATFLG                  ;SET FATAL ERROR FLAG
                                15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                        TRAP    C$CLP1
                                ;       If Extended Features Hardware Switch Clear then:
                                ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
                                        MOV     #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$:    ERRDF   ERRNO,T202SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                                                                                        TRAP    C$ERDF
                                                                                        .WORD   902
                                                                                        .WORD   T202SSR
                                                                                        .WORD   PKTSSR
                                        INC     FATFLG                  ;SET FATAL ERROR FLAG
                                30$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                        TRAP    C$CLP1
                                ;       Do WRITE CHARACTERISTICS to select reserved unit 7
                                        MOV     #T20PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
                                        JSR     PC,WRTCHR               ;DO WRITE CHARACTERISTICS COMMAND
                                        FORCERROR       42$                     ;@@DFORCE ERROR IF FORCER=1
                                        BCS     50$                     ;BR IF CARRY SET (GOOD RETURN)
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        NEXT.ERRNO
```

```
5311 063012                          42$:   ERRDF   ERRNO,T20SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     063012  104455                                                                TRAP    C$ERDF
     063014  001607                                                                .WORD   903
     063016  065171                                                                .WORD   T20SSR
     063020  012046                                                                .WORD   PKTSSR
5312 063022  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
5313 063026                          50$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063026  104406                                                                TRAP    C$CLP1
5314
5315
5316                                 ;  REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
5317 063030  012703  002752                 MOV     #TSTBLK,R3              ;GET FIRST PATTERN ADDRESS
5318 063034  012337  002312          100$:  MOV     (R3)+,DATA              ;GET A TEST PATTERN
5319 063040  042737  177400  002312        BIC     #^C<377>,DATA           ;DATA IS BYTE
5320 063046  010337  002316                 MOV     R3,TSTPTR               ;SETUP CURRENT TSTBLK POINTER
5321                                 ;       Do a WRITE NPR to set loopback and tape direction OUT and
5322                                 ;                 and SET Write Wrong Parity.
5323 063052  012700  000100                 MOV     #NP.OUT,R0              ;SET TAPE DIRECTION OUT
5324 063056  052700  000040                 BIS     #NP.LOOP,R0             ;SET LOOPBACK
5325 063062  042700  000020                 BIC     #NP.WRP,R0              ;SET WRITE WRONG PARITY (INVERTED)
5326 063066  004737  066126                 JSR     PC,T20WNPR              ;SETUP T20PK2 FOR WRITE NPR
5327 063072  012704  066510                 MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5328 063076  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5329 063102  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5330 063106                                 FORCERROR       102$                    ;@@DFORCE ERROR IF FORCER=1
5331 063122  103407                         BCS     105$                    ;BR IF CARRY SET (GOOD RETURN)
5332 063124  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5333 063126                                 NEXT.ERRNO
5334 063126                          102$:  ERRDF   ERRNO,T204SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063126  104455                                                                TRAP    C$ERDF
     063130  001610                                                                .WORD   904
     063132  065337                                                                .WORD   T204SSR
     063134  012046                                                                .WORD   PKTSSR
5335 063136  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
5336 063142                          105$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063142  104406                                                                TRAP    C$CLP1
5337                                 ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
5338 063144  012700  000001                 MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR = 1
5339 063150  004737  066222                 JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
5340 063154  012704  066510                 MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5341 063160  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5342 063164  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5343 063170                                 FORCERROR       112$                    ;@@DFORCE ERROR IF FORCER=1
5344 063204  103407                         BCS     120$                    ;BR IF CARRY SET (GOOD RETURN)
5345 063206  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5346 063210                                 NEXT.ERRNO
5347 063210                          112$:  ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063210  104455                                                                TRAP    C$ERDF
     063212  001611                                                                .WORD   905
     063214  065511                                                                .WORD   T208SSR
     063216  012046                                                                .WORD   PKTSSR
5348 063220  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
5349 063224                          120$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063224  104406                                                                TRAP    C$CLP1
5350                                 ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
5351 063226  012700  000001                 MOV     #1,R0                   ;WRITE 1 BYTE
5352 063232  012701  002312                 MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
```

```
5353 063236  004737  066166                  JSR     PC,T20WFIF              ;SETUP T20PK2 FOR WRITE FIFO
5354 063242  012704  066510                  MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5355 063246  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5356 063252  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5357 063256                                  FORCERROR       152$                    ;@@DFORCE ERROR IF FORCER=1
5358 063272  103407                          BCS     160$                    ;BR IF CARRY SET (GOOD RETURN)
5359 063274  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5360 063276                                  NEXT.ERRNO
5361 063276                          152$:   ERRDF   ERRNO,T205SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063276  104455                                                                      TRAP    C$ERDF
     063300  001612                                                                      .WORD   906
     063302  065402                                                                      .WORD   T205SSR
     063304  012046                                                                      .WORD   PKTSSR
5362 063306  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
5363 063312                          160$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063312  104406                                                                      TRAP    C$CLP1
5364                                 ;       Do a READ FIFO with tape direction OUT to load tape out write latch
5365                                 ;         (this is when wrong parity (IWP) is set)
5366 063314  012700  000001                  MOV     #1,R0                   ;SET READ BYTE COUNT
5367 063320  004737  066146                  JSR     PC,T20RFIF              ;SETUP T20PK2 FOR READ FIFO
5368 063324  012704  066510                  MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5369 063330  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5370 063334  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5371 063340                                  FORCERROR       172$                    ;@@DFORCE ERROR IF FORCER=1
5372 063354  103407                          BCS     180$                    ;BR IF CARRY SET (GOOD RETURN)
5373 063356  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5374 063360                                  NEXT.ERRNO
5375 063360                          172$:   ERRDF   ERRNO,T206SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063360  104455                                                                      TRAP    C$ERDF
     063362  001613                                                                      .WORD   907
     063364  065446                                                                      .WORD   T206SSR
     063366  012046                                                                      .WORD   PKTSSR
5376 063370  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
5377 063374                          180$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063374  104406                                                                      TRAP    C$CLP1
5378                                 ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 0 (sets read strobe low)
5379                                 ;               (Read Strobe sets PAR IN H [Parity Error])
5380 063376  005000                          CLR     R0                      ;IRESV4==>IRSTR = 0
5381 063400  004737  066222                  JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
5382 063404  012704  066510                  MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5383 063410  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5384 063414  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5385 063420                                  FORCERROR       192$                    ;@@DFORCE ERROR IF FORCER=1
5386 063434  103407                          BCS     200$                    ;BR IF CARRY SET (GOOD RETURN)
5387 063436  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5388 063440                                  NEXT.ERRNO
5389 063440                          192$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063440  104455                                                                      TRAP    C$ERDF
     063442  001614                                                                      .WORD   908
     063444  065511                                                                      .WORD   T208SSR
     063446  012046                                                                      .WORD   PKTSSR
5390 063450  005237  002222                  INC     FATFLG                  ;SET FATAL ERROR FLAG
5391 063454                          200$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063454  104406                                                                      TRAP    C$CLP1
5392                                 ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
5393 063456  012700  000001                  MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR = 1
5394 063462  004737  066222                  JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
```

```
5395 063466  012704  066510          MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5396 063472  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5397 063476  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5398 063502                          FORCERROR       212$                    ;@@DFORCE ERROR IF FORCER=1
5399 063516  103407                  BCS     220$                    ;BR IF CARRY SET (GOOD RETURN)
5400 063520  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5401 063522                          NEXT.ERRNO
5402 063522                  212$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063522  104455                                                                  TRAP    C$ERDF
     063524  001615                                                                  .WORD   909
     063526  065511                                                                  .WORD   T208SSR
     063530  012046                                                                  .WORD   PKTSSR
5403 063532  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
5404 063536                  220$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063536  104406                                                                  TRAP    C$CLP1
5405                         ;       Do a Write Subsystem READ STATUS
5406 063540  004737  066106          JSR     PC,T20SRD               ;SETUP PACKET FOR READ STATUS
5407 063544  012704  066510          MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5408 063550  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5409 063554  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5410 063560                          FORCERROR       232$                    ;@@DFORCE ERROR IF FORCER=1
5411 063574  103407                  BCS     240$                    ;BR IF CARRY SET (GOOD RETURN)
5412 063576  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5413 063600                          NEXT.ERRNO
5414 063600                  232$:   ERRDF   ERRNO,T203SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063600  104455                                                                  TRAP    C$ERDF
     063602  001616                                                                  .WORD   910
     063604  065272                                                                  .WORD   T203SSR
     063606  012046                                                                  .WORD   PKTSSR
5415 063610  005237  002222          INC     FATFLG                  ;SET FATAL ERROR FLAG
5416 063614                  240$:   CKLOC                           ;LOOP ON ERROR, IF FLAG SET
     063614  104406                                                                  TRAP    C$CLP1
5417                         ;       If Read Data parity error NOT=1  Then Print Error
5418 063616  004737  066314          JSR     PC,T20SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
5419 063622  012701  065042          MOV     #T20EXSTA,R1            ;GET EXPECTED READ STATUS
5420 063626  012702  066402          MOV     #T20BFSTA,R2            ;GET RECV READ STATUS
5421 063632  011211                  MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
5422 063634  016261  000002  000002  MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTED)
5423 063642  052711  100000          BIS     #S1.PARERR,(R1)         ;SET EXP PAR ERR  =1
5424 063646  005000                  CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
5425 063650  012701  066362          MOV     #T20BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
5426 063654  012702  065022          MOV     #T20EXP,R2              ;EXPD ADDRESS
5427 063660  012703  000024          MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
5428 063664  004737  011500          JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
5429 063670                          FORCERROR       252$,NOTSSR             ;@@D
5430 063700  103404                  BCS     260$                    ;BR IF YES
5431 063702                          NEXT.ERRNO
5432 063702                  252$:   ERRHRD  ERRNO,T20SWP,MSGSTAT    ;REPORT ERROR
     063702  104456                                                                  TRAP    C$ERHRD
     063704  001617                                                                  .WORD   911
     063706  065557                                                                  .WORD   T20SWP
     063710  012350                                                                  .WORD   MSGSTAT
5433 063712                  260$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063712  104406                                                                  TRAP    C$CLP1
5434                         ;       Do a Write Misc to RESET FIFO
5435 063714  012700  000020          MOV     #MS.RSFIF,R0            ;SET RESET FIFO COMMAND
5436 063720  004737  066242          JSR     PC,T20WMISC             ;SETUP T20PK2 FOR WRITE MISC
```

```
5437 063724  012704  066510                 MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5438 063730  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5439 063734  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5440 063740                                 FORCERROR       282$                    ;@@DFORCE ERROR IF FORCER=1
5441 063754  103407                        BCS     290$                    ;BR IF CARRY SET (GOOD RETURN)
5442 063756  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5443 063760                                 NEXT.ERRNO
5444 063760                         282$:   ERRDF   ERRNO,T202SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     063760  104455                                                                TRAP    C$ERDF
     063762  001620                                                                .WORD   912
     063764  065226                                                                .WORD   T202SSR
     063766  012046                                                                .WORD   PKTSSR
5445 063770  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
5446 063774                         290$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     063774  104406                                                                TRAP    C$CLP1
5447                                ;       Do a Write Subsystem READ STATUS
5448                                ;       If Read Data parity error NOT=0  Then Print Error
5449 063776  004737  066314                 JSR     PC,T20SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
5450 064002  012701  065042                 MOV     #T20EXSTA,R1            ;GET EXPECTED READ STATUS
5451 064006  012702  066402                 MOV     #T208FSTA,R2            ;GET RECV READ STATUS
5452 064012  011211                         MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
5453 064014  016261  000002  000002         MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTED)
5454 064022  042711  100000                 BIC     #S1.PARERR,(R1)         ;SET EXP PAR ERR  =0
5455 064026  005000                         CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
5456 064030  012701  066362                 MOV     #T208FR,R1              ;LOW RECV ADDRESS FOR CKMSG2
5457 064034  012702  065022                 MOV     #T20EXP,R2              ;EXPD ADDRESS
5458 064040  012703  000024                 MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
5459 064044  004737  011500                 JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
5460 064050                                 FORCERROR       302$,NOTSSR     ;@@D
5461 064060  103404                        BCS     320$                    ;BR IF YES
5462 064062                                 NEXT.ERRNO
5463 064062                         302$:   ERRHRD  ERRNO,T20RSF,MSGSTAT    ;REPORT ERROR
     064062  104456                                                                TRAP    C$ERHRD
     064064  001621                                                                .WORD   913
     064066  065666                                                                .WORD   T20RSF
     064070  012350                                                                .WORD   MSGSTAT
5464 064072                         320$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064072  104406                                                                TRAP    C$CLP1
5465                                ;       (* Verify Data can be transferred without a Parity Error *)
5466                                ;       Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
5467 064074  012700  000001                 MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR = 1
5468 064100  004737  066222                 JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
5469 064104  012704  066510                 MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5470 064110  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5471 064114  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5472 064120                                 FORCERROR       332$                    ;@@DFORCE ERROR IF FORCER=1
5473 064134  103407                        BCS     340$                    ;BR IF CARRY SET (GOOD RETURN)
5474 064136  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5475 064140                                 NEXT.ERRNO
5476 064140                         332$:   ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064140  104455                                                                TRAP    C$ERDF
     064142  001622                                                                .WORD   914
     064144  065511                                                                .WORD   T208SSR
     064146  012046                                                                .WORD   PKTSSR
5477 064150  005237  002222                 INC     FATFLG                  ;SET FATAL ERROR FLAG
5478 064154                         340$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064154  104406                                                                TRAP    C$CLP1
```

```
5479                                    ;       Do a WRITE MPR to set loopback and tape direction OUT and
5480                                    ;               and CLEAR Write Wrong Parity.
5481 064156  012700  000100                MOV     @#P.OUT,R0              ;SET TAPE DIRECTION OUT
5482 064162  052700  000040                BIS     @#P.LOOP,R0             ;SET LOOPBACK
5483 064166  052700  000020                BIS     @#P.WRP,R0              ;CLEAR WRITE WRONG PARITY (INVERTED)
5484 064172  004737  066126                JSR     PC,T20WMPR              ;SETUP T20PK2 FOR WRITE MPR
5485 064176  012704  066510                MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5486 064202  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5487 064206  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5488 064212                                FORCERROR       352$                    ;@@@FORCE ERROR IF FORCER=1
5489 064226  103407                        BCS     360$                    ;BR IF CARRY SET (GOOD RETURN)
5490 064230  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5491 064232                                NEXT.ERRNO
5492 064232                        352$:   ERRDF   ERRNO,T204SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064232  104455                                                                    TRAP    C$ERDF
     064234  001623                                                                    .WORD   915
     064236  065337                                                                    .WORD   T204SSR
     064240  012046                                                                    .WORD   PKTSSR
5493 064242  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
5494 064246                        360$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064246  104406                                                                    TRAP    C$CLP1
5495                                    ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
5496 064250  012700  000001                MOV     #1,R0                   ;WRITE 1 BYTE
5497 064254  012701  002312                MOV     #DATA,R1                ;FIFO WRITE DATA ADDRESS
5498 064260  004737  066166                JSR     PC,T20WFIF              ;SETUP T20PK2 FOR WRITE FIFO
5499 064264  012704  066510                MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5500 064270  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5501 064274  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5502 064300                                FORCERROR       372$                    ;@@@FORCE ERROR IF FORCER=1
5503 064314  103407                        BCS     380$                    ;BR IF CARRY SET (GOOD RETURN)
5504 064316  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5505 064320                                NEXT.ERRNO
5506 064320                        372$:   ERRDF   ERRNO,T205SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064320  104455                                                                    TRAP    C$ERDF
     064322  001624                                                                    .WORD   916
     064324  065402                                                                    .WORD   T205SSR
     064326  012046                                                                    .WORD   PKTSSR
5507 064330  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
5508 064334                        380$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064334  104406                                                                    TRAP    C$CLP1
5509                                    ;       Do a READ FIFO with tape direction OUT to load tape out write latch
5510 064336  012700  000001                MOV     #1,R0                   ;SET READ BYTE COUNT
5511 064342  004737  066146                JSR     PC,T20RFIF              ;SETUP T20PK2 FOR READ FIFO
5512 064346  012704  066510                MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
5513 064352  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
5514 064356  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5515 064362                                FORCERROR       392$                    ;@@@FORCE ERROR IF FORCER=1
5516 064376  103407                        BCS     400$                    ;BR IF CARRY SET (GOOD RETURN)
5517 064400  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5518 064402                                NEXT.ERRNO
5519 064402                        392$:   ERRDF   ERRNO,T206SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064402  104455                                                                    TRAP    C$ERDF
     064404  001625                                                                    .WORD   917
     064406  065446                                                                    .WORD   T206SSR
     064410  012046                                                                    .WORD   PKTSSR
5520 064412  005237  002222                INC     FATFLG                  ;SET FATAL ERROR FLAG
5521 064416                        400$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

```
     064416  104406                                                                     TRAP    C$CLP1
                                      ;        Do a WRITE FORMAT  to set IRESV4==>IRSTR = 0 (sets read strobe low)
                                      ;               (Read Strobe sets PAR IN H [Parity Error])
     064420  005000                            CLR     R0                      ;IRESV4==>IRSTR = 0
     064422  004737  066222                    JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
     064426  012704  066510                    MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
     064432  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
     064436  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
     064442                                    FORCERROR       412$              ;@@DFORCE ERROR IF FORCER=1
     064456  103407                            BCS     420$                    ;BR IF CARRY SET (GOOD RETURN)
     064460  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
     064462                                    NEXT.ERRNO
     064462                           412$:    ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064462  104455                                                                     TRAP    C$ERDF
     064464  001626                                                                     .WORD   918
     064466  065511                                                                     .WORD   T208SSR
     064470  012046                                                                     .WORD   PKTSSR
     064472  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
     064476                           420$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064476  104406                                                                     TRAP    C$CLP1
                                      ;        Do a WRITE FORMAT  to set IRESV4==>IRSTR = 1 (sets read strobe high)
     064500  012700  000001                    MOV     #WF.I4RES,R0            ;IRESV4==>IRSTR = 1
     064504  004737  066222                    JSR     PC,T20WFMT              ;SETUP T20PK2 FOR WRITE FORMAT
     064510  012704  066510                    MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
     064514  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
     064520  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
     064524                                    FORCERROR       432$              ;@@DFORCE ERROR IF FORCER=1
     064540  103407                            BCS     440$                    ;BR IF CARRY SET (GOOD RETURN)
     064542  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
     064544                                    NEXT.ERRNO
     064544                           432$:    ERRDF   ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064544  104455                                                                     TRAP    C$ERDF
     064546  001627                                                                     .WORD   919
     064550  065511                                                                     .WORD   T208SSR
     064552  012046                                                                     .WORD   PKTSSR
     064554  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
     064560                           440$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064560  104406                                                                     TRAP    C$CLP1

                                      ;        Do a Write Subsystem READ STATUS
     064562  004737  066106                    JSR     PC,T20SRD               ;SETUP PACKET FOR READ STATUS
     064566  012704  066510                    MOV     #T20PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
     064572  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
     064576  004737  016336                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
     064602                                    FORCERROR       452$              ;@@DFORCE ERROR IF FORCER=1
     064616  103407                            BCS     460$                    ;BR IF CARRY SET (GOOD RETURN)
     064620  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
     064622                                    NEXT.ERRNO
     064622                           452$:    ERRDF   ERRNO,T203SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
     064622  104455                                                                     TRAP    C$ERDF
     064624  001630                                                                     .WORD   920
     064626  065272                                                                     .WORD   T203SSR
     064630  012046                                                                     .WORD   PKTSSR
     064632  005237  002222                    INC     FATFLG                  ;SET FATAL ERROR FLAG
     064636                           460$     CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064636  104406                                                                     TRAP    C$CLP1
                                      ;        If Read Data parity error NOT=0  Then Print Error
```

```
5563 064640  004737  066314                    JSR     PC,T20SETEXP            ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
5564 064644  012701  065042                    MOV     #T20EXSTA,R1            ;GET EXPECTED READ STATUS
5565 064650  012702  066402                    MOV     #T20BFSTA,R2            ;GET RECV READ STATUS
5566 064654  011211                            MOV     (R2),(R1)               ;SET EXPD WORD #8 = RECV TEMP
5567 064656  016261  000002  000002            MOV     2(R2),2(R1)             ;SET EXPD WORD #9 = RECV (NOT TESTED)
5568 064664  042711  100000                    BIC     #S1.PARERR,(R1)         ;SET EXP PAR ERR  =0
5569 064670  005000                            CLR     R0                      ;HIGH RECV ADDRESS FOR CKMSG2
5570 064672  012701  066362                    MOV     #T20BFR,R1              ;LOW RECV ADDRESS FOR CKMSG2
5571 064676  012702  065022                    MOV     #T20EXP,R2              ;EXPD ADDRESS
5572 064702  012703  000024                    MOV     #20.,R3                 ;NUMBER OF BYTES TO COMPARE
5573 064706  004737  011500                    JSR     PC,CKMSG2               ;EXPD EQUAL RECV?
5574 064712                                    FORCERROR       472$,NOTSSR     ;BAD
5575 064722  103404                            BCS     480$                    ;BR IF YES
5576 064724                                    NEXT.ERRNO
5577 064724                            472$:   ERRHRD  ERRNO,T20CMP,MSGSTAT    ;REPORT ERROR
     064724  104456                                                                    TRAP    C$ERHRD
     064726  001631                                                                    .WORD   921
     064730  065767                                                                    .WORD   T20CMP
     064732  012350                                                                    .WORD   MSGSTAT
5578 064734                            480$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     064734  104406                                                                    TRAP    C$CLP1
5579
5580 064736                                    FORCEXIT        555$            ;BAD
5581 064746  013703  002316                    MOV     TSTPTR,R3               ;RESTORE CURRENT TSTBLK POINTER
5582 064752  020327  003062                    CMP     R3,#TBLEND              ;END OF TSTBLK?
5583 064756  103002                            BHIS    555$                    ;BR IF YES
5584 064760  000137  063034                    JMP     100$                    ;DO ANOTHER TSTBLK PATTERN
5585 064764                            555$:
5586
5587 064764                                    ENDSUB                          ;////////////// END SUBTEST //////////
     064764                                                                            L10074:
     064764  104403                                                                    TRAP    C$ESUB
5588
5589 064766  005737  002222                    TST     FATFLG                  ;ANY FATAL ERRORS ?
5590 064772  001402                            BEQ     560$                    ;BRANCH IF NOT
5591 064774  004737  017202                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
5592 065000                            560$:
5593 065000  004737  016456                    JSR     PC,TSTLOOP              ;DO ITERATIONS?
5594 065004  103002                            BCC     565$                    ;BR IF NO
5595 065006  000137  050466                    JMP     T18LOOP                 ;LOOP UNTIL ITERATIONS DONE
5596 065012                            565$:
5597 065012                                    EXIT    TST                     ;////////// EXIT TEST //////////
     065012  104432                                                                    TRAP    C$EXIT
     065014  001610                                                                    .WORD   L10073-.
5598
5599
5600                                   ;+
5601                                   ;LOCAL STORAGE FOR THIS TEST
5602                                   ;-
5603
5604
5605 065016                            T20MSK:                                 ;MASK OF UNTESTED BITS IN READ STATUS
5606                                                                            ;UNTESTED BITS ARE SET TO 1
5607 065016      377                           .BYTE   ^C<000>                 ;BYTE 0 MASK
5608 065017      037                           .BYTE   ^C<340>                 ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
5609 065020      360                           .BYTE   ^C<017>                 ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
5610 065021      000                           .BYTE   0                       ;MAKE IT EVEN
```

```
5611
5612 065022                                T20EXP:                         ;BEGIN EXPECTED DATA BUFFER
5613 065022  000000                               .WORD   0                ;MESSAGE TYPE
5614 065024  000000                               .WORD   0                ;DATA FIELD LENGTH
5615 065026  000000                               .WORD   0                ;RBPCR
5616 065030  000000                               .WORD   0                ;XST0
5617 065032  000000                               .WORD   0                ;XST1
5618 065034  000000                               .WORD   0                ;XST2
5619 065036  000000                               .WORD   0                ;XST3
5620 065040  000000                               .WORD   0                ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
5621 065042                                T20EXSTA: .BLKB 64.              ;EXPECTED READ STATUS AND WRITE FIFO DATA
5622 065142                                T20EXEND:                               ;END EXPECTED DATA BUFFER
5623                                        ;*
5624                                        ;LOCAL TEXT MESSAGES FOR TEST
5625                                        ;
5626
5627 065142      122      145      141    TST20ID:        .ASCIZ  'Read/Write Data Parity'
5628 065171      127      122      111    T20SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
5629 065226      127      122      111    T202SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
5630 065272      127      122      111    T203SSR:.ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
5631 065337      127      122      111    T204SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
5632 065402      127      122      111    T205SSR:.ASCIZ  'WRITE SUBSYSTEM (Write FIFO) Failed'
5633 065446      127      122      111    T206SSR:.ASCIZ  'WRITE SUBSYSTEM (Read FIFO) Failed'
5634 065511      127      122      111    T208SSR:.ASCIZ  'WRITE SUBSYSTEM (Write Format) Failed'
5635 065557      122      145      141    T20SWP: .ASCIZ  'Read Data Parity Error (PARERR) Failed to Set after Write Wrong Parity
5636 065666      122      145      141    T20RSF: .ASCIZ  'Read Data Parity Error (PARERR) Failed to Clear after RESET FIFO'
5637 065767      122      145      141    T20CWP: .ASCIZ  'Read Data Parity Error (PARERR) occurred in Data Loopback'
5638                                               .EVEN
5639
5640                                        ;*
5641                                        ; CLEAR MESSAGE BUFFER
5642                                        ;-
5643 066062                                T20CLRBUF:                      ;
5644 066062                                        SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
5645 066066  012701  066362                       MOV     #T20BFR,R1      ;GET MESSAGE BUFFER ADDRESS
5646 066072  012702  000120                       MOV     #T20BEND-T20BFR,R2 ;SIZE OF MESSAGE BUFFER IN BYTES
5647 066076  105021                        10$:    CLRB    (R1)+           ;CLEAR A BYTE
5648 066100  005302                               DEC     R2              ;DONE?
5649 066102  003375                               BGT     10$             ;BR IF NO
5650 066104  000207                               RTS     PC              ;RETURN
5651
5652                                        ;*
5653                                        ; SETUP T20PK2 PACKET FOR READ STATUS
5654                                        ;-
5655 066106                                T20SRD:
5656 066106  004737  066062                       JSR     PC,T20CLRBUF    ;CLEAR MESSAGE BUFFER
5657 066112  012700  066520                       MOV     #T20DT2,R0      ;WRITE SUBSYSTEM DATA BUFFER
5658 066116  112720  000005                       MOVB    #PW.RDSTATUS,(R0)+ ;STORE READ STATUS COMMAND IN BSEL0
5659 066122  105010                               CLRB    (R0)            ;CLEAR BSEL1
5660 066124  000207                               RTS     PC              ;RETURN
5661
5662
5663                                        ;*
5664                                        ; SETUP T20PK2 PACKET FOR WRITE NPR
5665                                        ;
5666                                        ; INPUT:
5667                                        ;       R0 CONTAINS BSEL1 NPR DATA
```

```
5668                                    ;
5669                                    ;
5670 066126                             T20WNPR:
5671 066126  004737  066062                    JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
5672 066132  012701  066520                    MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5673 066136  112721  000011                    MOVB    #PW.WNPR,(R1)+          ;STORE WRITE NPR IN BSEL0
5674 066142  110011                            MOVB    R0,(R1)                 ;STORE NPR DATA IN BSEL1
5675 066144  000207                            RTS     PC                      ;RETURN
5676
5677                                    ;+
5678                                    ; SETUP T20PK2 PACKET FOR READ FIFO
5679                                    ;
5680                                    ; INPUT:
5681                                    ;       R0 CONTAINS SEL2 BYTE COUNT
5682                                    ;-
5683 066146                             T20RFIF:
5684 066146  004737  066062                    JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
5685 066152  012701  066520                    MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5686 066156  112721  000003                    MOVB    #PW.RFIFO,(R1)+         ;STORE READ FIFO IN BSEL0
5687 066162  110021                            MOVB    R0,(R1)+                ;STORE BYTE COUNT IN BSEL1
5688 066164  000207                            RTS     PC                      ;RETURN
5689
5690                                    ;+
5691                                    ; SETUP T20PK2 PACKET FOR WRITE FIFO
5692                                    ;
5693                                    ; INPUT:
5694                                    ;       R0 CONTAINS BYTE COUNT
5695                                    ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
5696 066166                             ;-
5697 066166                             T20WFIF:
5698 066172  004737  066062                    SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
5699 066176  012702  066520                    JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
5700 066202  112722  000004                    MOV     #T20DT2,R2              ;WRITE SUBSYSTEM DATA BUFFER
5701 066206  110022                            MOVB    #PW.WFIFO,(R2)+         ;STORE WRITE FIFO IN BSEL0
5702 066210  005022                            MOVB    R0,(R2)+                ;STORE BYTE COUNT IN BSEL1
5703 066212  112122                            CLR     (R2)+                   ;CLEAR SEL2 (UNUSED)
5704 066214  005300                    10$:    MOVB    (R1)+,(R2)+             ;STORE DATA PATTERN BYTE
5705 066216  003375                            DEC     R0                      ;DONE ALL BYTES?
5706 066220  000207                            BGT     10$                     ;BR IF NO
5707                                           RTS     PC                      ;RETURN
5708
5709                                    ;+
5710                                    ; SETUP T20PK2 FOR WRITE FORMAT TRANSPORT REGISTER
5711                                    ;
5712                                    ; INPUT:
5713                                    ;       R0 CONTAINS DRIVING DATA PATTERN
5714 066222                             ;-
5715 066222  004737  066062             T20WFMT:
5716 066226  012701  066520                    JSR     PC,T20CLRBUF            ;CLEAR MESSAGE BUFFER
5717 066232  112721  000007                    MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5718 066236  110021                            MOVB    #PW.WFMT,(R1)+          ;STORE WRITE FORMAT IN BSEL0
5719 066240  000207                            MOVB    R0,(R1)+                ;STORE DATA WORD IN BSEL1
5720                                           RTS     PC                      ;RETURN
5721                                    ;+
5722                                    ; SETUP T20PK2 PACKET FOR WRITE MISC.
5723                                    ;
5724                                    ;       R0 CONTAINS WRITE MISC DATA
                                        ;-
```

```
5725 066242                              T20WMISC:
5726 066242  012701  066520                      MOV     #T20DT2,R1              ;WRITE SUBSYSTEM DATA BUFFER
5727 066246  112721  000010                      MOVB    #PW.WMISC,(R1)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
5728 066252  110011                              MOVB    R0,(R1)                 ;STORE INVERT EXTENDED FEATURES IN BSEL1
5729 066254  000207                              RTS     PC                      ;RETURN
5730                                     ;+
5731                                     ; SETUP T20PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
5732                                     ;-
5733 066256                              T20SEXT:
5734 066256  012700  066520                      MOV     #T20DT2,R0              ;WRITE SUBSYSTEM DATA BUFFER
5735 066262  112720  000010                      MOVB    #PW.WMISC,(R0)+         ;STORE WRITE MISCELLANEOUS IN BSEL0
5736 066266  112710  000200                      MOVB    #MS.EXT,(R0)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
5737 066272  000207                              RTS     PC                      ;RETURN
5738                                     ;+
5739                                     ; CLEAR EXPECTED DATA MESSAGE BUFFER
5740                                     ;-
5741 066274                              T20CLEXP:
5742 066274  012701  065022                      MOV     #T20EXP,R1              ;GET EXPD ADDRESS
5743 066300  012700  000120                      MOV     #T20EXEND-T20EXP,R0     ;GET EXPD SIZE
5744 066304  105021                      10$:    CLRB    (R1)+                   ;CLEAR A BYTE
5745 066306  005300                              DEC     R0                      ;DONE?
5746 066310  003375                              BGT     10$                     ;BR IF NO
5747 066312  000207                              RTS     PC                      ;RETURN
5748
5749                                     ;+
5750                                     ;Set WORDS 0-7 of expd message BUFFER = to recv since not testing
5751                                     ;-
5752 066314                              T20SETEXP:
5753 066314  012702  065022                      MOV     #T20EXP,R2              ;GET EXPD
5754 066320  012703  066362                      MOV     #T20BFR,R3              ;GET READ STATUS RECV BUFFER
5755 066324  012700  000010                      MOV     #8.,R0                  ;SET WORDS 0-7 EXP=RECV
5756 066330  012322                      5$:     MOV     (R3)+,(R2)+             ;SET EXPD=RECV
5757 066332  005300                              DEC     R0                      ;DONE WORDS 0-7 WORDS?
5758 066334  003375                              BGT     5$                      ;BR IF NO
5759 066336  000207                              RTS     PC                      ;RETURN
5760
5761
5762
5766                                     ;
5767                                     ;WRITE CHARACTERISTICS COMMAND PACKET
5768                                     ;
5769 066340                              T20PACKET:                              ;COMMAND PACKET FOR TEST
5770 066340  100004                              .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
5771 066342  066350                              .WORD   T20DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
5772 066344  000000                              .WORD   0
5773 066346  000012                              .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
5774
5775 066350                              T20DATA:                                ;CHARACTERISTICS DATA BLOCK
5776 066350  066362                              .WORD   T20BFR                  ;ADDRESS OF MESSAGE BUFFER
5777 066352  000000                              .WORD   0
5778 066354  000024                              .WORD   20.                     ;LENGTH OF MESSAGE BUFFER
5779 066356  000000                              .WORD   0                       ;ESS,ENB,EAI,ERI
5780 066360  000007                              .WORD   7                       ;EXTENDED FEATURES UNIT NO.
5781
5782
5783                                     ;MESSAGE BUFFER FOR ALL TEST 17 COMMANDS
5784
```

```
5785 066362                            T20BFR:                         ;BEGIN MESSAGE BUFFER
5786 066362  000000                            .WORD   0               ;MESSAGE TYPE
5787 066364  000000                            .WORD   0               ;DATA FIELD LENGTH
5788 066366  000000                            .WORD   0               ;RBPCR
5789 066370  000000                            .WORD   0               ;XST0
5790 066372  000000                            .WORD   0               ;XST1
5791 066374  000000                            .WORD   0               ;XST2
5792 066376  000000                            .WORD   0               ;XST3
5793 066400  000000                            .WORD   0               ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
5794 066402                            T20BFSTA: .BLKB 64.               ;READ STATUS AND WRITE FIFO BUFFER
5795 066502                            T20BEND:                        ;END OF MESSAGE BUFFER
5796                                   ;
5797                                   ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
5798                                   ;
5800         066510                            .=<.+10>&177770
5802 066510                            T20PK2:
5803 066510  100006                            .WORD   P.WRTSUB!P.ACK          ;WRITE SUBSYSTEM WITH ACK
5804 066512  066520                            .WORD   T20DT2                  ;LOW ADDRESS OF DATA BLOCK
5805 066514  000000                            .WORD   0                       ;HIGH ADDRESS OF DATA BLOCK
5806 066516  000012                            .WORD   10.                     ;MINIMUM MESSAGE PACKET SIZE
5807
5808 066520                            T20DT2:                         ;DATA BLOCK
5809 066520     000                            .BYTE   0               ;BSEL0
5810 066521     000                            .BYTE   0               ;BSEL1
5811 066522  000000                            .WORD   0               ;SEL2
5812 066524                                    .BLKB   64.             ;WRITE FIFO DATA OUTPUT BUFFER
5813
5814
5815 066624                                    ENDTST
     066624                                                                    L10073:
     066624  104401                                                            TRAP    C$ETST
5816                                           .SBTTL  TEST 10: MANUAL INTERVENTION
5817
5818                                   ;THE MANUAL INTERVENTION TEST IS A STANDALONE ROUTINE (NOT REALLY A "TEST")
5819                                   ;THAT ALLOWS THE OPERATOR TO CHECK OUT VARIOUS ELEMENTS AND FUNCTIONS OF
5820                                   ;THE SUBSYSTEM THAT CANNOT BE MANIPULATED BY THE PROGRAM ALONE. WHEN
5821                                   ;THIS ROUTINE IS STARTED, IT FIRST PRINTS OUT A MENU OF SELECTABLE
5822                                   ;SUBTESTS AND THEN WAITS FOR THE OPERATOR TO TYPE IN A SELECTION CODE.
5823                                   ;THE ONLY WAYS TO EXIT THIS ROUTINE AND RETURN TO THE DIAGNOSTIC SUPERVISOR
5824                                   ;ARE BY TYPING <CTRL-C> OR SELECTING CODE 7.
5825                                   ;SELECTION CODES AND SUBROUTINES ARE:
5826                                   ;
5827                                   ;
5828                                   ;       CODE    ROUTINE
5829                                   ;
5830                                   ;         0     HELP. PRINTS THIS MENU.
5831                                   ;         1     TURN ON ALL M7196 LED INDICATORS
5832                                   ;         2     TURN OFF ALL M7196 LED INDICATORS
5833                                   ;         3     OFFLINE/ONLINE ATTENTION TEST
5834                                   ;         4     WRITE-PROTECT TEST
5835                                   ;         5     INITIATE TRANSPORT SERVO EXERCISER
5836                                   ;         6     PRINT EXTENDED TRANSPORT STATUS
5837                                   ;         7     EXIT (RETURN TO SUPERVISOR)
5838                                   ;
5839                                   ;
5840                                   ;EACH MENU ITEM CORRESPONDS TO A SUBTEST, AS FOLLOWS:
5841                                   ;
```

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

```
                                        ;
                                        ;
                                        ;
                                        ;
                                BGNTST
                                                                        T10::
                                RFLAGS  R0              ;GET OPERATOR FLAGS
                                                                        TRAP    C$RFLA
                                BEQ     21$             ;BR, IF OK TO RUN
                                MOV     #T38NE,R0       ;"TEST NOT EXECUTED"
                                BR      3$              ;JUMP IF NOT FIRST TEST
                            21$:
                                MOV     #T38ID,R0       ;TEST ID MESSAGE
                            3$: JSR     PC,TSTSETUP     ;DO THE COMMON SETUP
                                JSR     PC,CHKMAN       ;IS MANUAL INTERVENTION ALLOWED?
                                BCS     22$             ;BR, IF MANUAL INTER ALLOWED
                                JMP     64$             ;JUMP IF NOT ALLOWED
                            22$:
                            2$: CLR     FATFLG          ;CLEAR THE FATAL ERROR FLAG
                                MOV     #65000.,T38DLY  ;SET UP DELAY COUNTER
                            5$: JSR     PC,SOFINIT      ;DO A SOFT INIT
                                BCS     23$             ;BRANCH IF OK
                                MOV     R0,R1           ;CONTENTS OF TSSR REGISTER
                                BIT     #SSR,R1         ;CHECK FOR TSSR SET
                                BNE     23$             ;KEEP GOING IF NOT SET
                                DELAY   250             ;CALL DELAY ROUTINE
                                                                        MOV     #250.,(PC)+
                                                                        .WORD   0
                                                                        MOV     L$DLY,(PC)+
                                                                        .WORD   0
                                                                        DEC     -6(PC)
                                                                        BNE     .-4
                                                                        DEC     -22(PC)
                                                                        BNE     .-20
                                DEC     T38DLY          ;BUMP COUNTER DOWN
                                BNE     5$              ;BR, IF MORE TIME LEFT
                                ERRDF   ERRNO,SFIERR,SFIMSG     ;REPORT FATAL ERROR
                                                                        TRAP    C$ERDF
                                                                        .WORD   1001
                                                                        .WORD   SFIERR
                                                                        .WORD   SFIMSG
                            23$:MOV     #MIMENU,R0      ;MENU OF MANUAL INTERVENTIONS
                                MOV     #6,R1           ;MAXIMUM ALLOWED SELECTION
                                JSR     PC,GETSEL       ;GO GET THE OPERATORS SELECTION
                                MOV     R0,R4           ;GET NUMBER FROM ROUTINE
                                ASL     R4              ;CONVERT TO WORD OFFSET
                                JMP     @6$(R4)         ;JUMP TO PROPER LOOP
                            6$: .WORD   2$              ;RETYPE THE MENU
                                .WORD   10$             ; 1 TURN ON LED'S
                                .WORD   15$             ; 2 TURN OFF LED'S
                                .WORD   20$             ; 3 ONLINE ATTENTION
                                .WORD   25$             ; 4 WRITE PROTECT
                                .WORD   35$             ; 5 EXTENDED TRANSPORT STATUS
                                .WORD   63$             ; 6 LEAVE THE TEST
                            10$:PRINTF  #T38MS2         ;TELL OPERATOR TO CNTRL-C FOR EXIT
                                                                        MOV     #T38MS2,-(SP)
                                                                        MOV     #1,-(SP)
```

```
        067032  010600                                                          MOV     SP,R0
        067034  104417                                                          TRAP    C$PNTF
        067036  062706  000004                                                  ADD     #4,SP
  5947  067042  004737  073756                  JSR     PC,T38REST              ;SET PACKET TO INITIAL VALUES
  5948  067046  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
  5949  067052  103405                          BCS     100$                    ;BR IF SOFT INIT = OK
  5953  067054  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  5954  067056                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
        067056  104455                                                          TRAP    C$ERDF
        067060  001752                                                          .WORD   1002
        067062  003652                                                          .WORD   SFIERR
        067064  012034                                                          .WORD   SFIMSG
  5955  067066  013737  002202  072150  100$:   MOV     UNITN,T38DSW            ;SET UNIT NUMBER
  5956
  5957  067074  012704  072130                  MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
  5958  067100  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
  5959  067104  103405                          BCS     110$                    ;BR. IF COMMAND ISSUED OK
  5963  067106  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  5964  067110                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
        067110  104456                                                          TRAP    C$ERHRD
        067112  001753                                                          .WORD   1003
        067114  005056                                                          .WORD   WRTMSG
        067116  012034                                                          .WORD   SFIMSG
  5965  067120                          110$:
  5966  067120  112737  000000  071441          MOVB    #0,T38BS1               ;CLEAR BIT #4
  5967  067126  112737  000011  071440          MOVB    #11,T38BSO              ;WRITE MISC COMMAND
  5968  067134  012704  071430                  MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
  5969                                  ;
  5970                                  ;NOTE: THIS COMMAND TURNS ON THE PROCESSOR FAIL LED
  5971                                  ;
  5972  067140  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
  5973  067144  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
  5974  067150  103405                          BCS     150$                    ;BR IF CARRY SET (GOOD RETURN)
  5975  067152  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  5979  067154                                  ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        067154  104455                                                          TRAP    C$ERDF
        067156  001754                                                          .WORD   1004
        067160  072626                                                          .WORD   T38SSR
        067162  012046                                                          .WORD   PKTSSR
  5980  067164                          150$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        067164  104406                                                          TRAP    C$CLP1
  5981  067166                                  SETPRI  #PRI07                  ;RAISE THE PRIORITY
        067166  012700  000340                                                  MOV     #PRI07,R0
        067172  104441                                                          TRAP    C$SPRI
  5982  067174  005037  071414                  CLR     TTION2                  ;ASSUME INTERRUPTS ARE ENABLED
  5983  067200  032737  000100  177560          BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
  5984  067206  001005                          BNE     701$                    ;BRANCH IF YES
  5985  067210  005237  071414                  INC     TTION2                  ;FLAG SET IF INTERRUPTS OFF
  5986  067214  052737  000100  177560          BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
  5987  067222  012701  000060          701$:   MOV     #TTIVEC,R1              ;START OF TTI VECTORS
  5988  067226  011137  071416                  MOV     (R1),TVSAV2             ;SAVE THE CURRENT TTI VECTOR
  5989  067232  012721  070710                  MOV     #590$,(R1)+             ;SET NEW INTERRUPT ROUTINE
  5990  067236  011137  071420                  MOV     (R1),TPSAV2             ;SAVE THE VECTOR PRIORITY
  5991  067242  012711  000340                  MOV     #PRI07,(R1)             ;USE PRIORITY SEVEN
  5992  067246                                  SETPRI  #PRI00                  ;LOWER INTERRUPT BR LEVEL
        067246  012700  000000                                                  MOV     #PRI00,R0
        067252  104441                                                          TRAP    C$SPRI
```

```
 5993 067254  012701  177777            MOV     # 1,R1                  ;DATA TO WRITE TO TSDB
 5994 067260  000240              12$:   NOP                             ;ALLOW OPERATOR TO TYPE ^C
 5995 067262  012702  001750            MOV     #1000.,R2               ;SET-UP INNER LOOP
 5996 067266  110165  000000      14$:   MOVB    R1,TSDB(R5)             ;WRITE DATA TO TSDB
 5997 067272  016500  000002            MOV     TSSR(R5),R0             ;READ TSSR
 5998 067276  005302                    DEC     R2                      ;REDUCE INNER COUNT
 5999 067300  001372                    BNE     14$                     ;LOOP TILL EXPIRES
 6000 067302  000766                    BR      12$                     ;LOOP UNTIL HALTED
 6001
 6002 067304                      15$:   PRINTF  #T38MS2                 ;TYPE CNTL C TO EXIT
      067304  012746  073221                                                    MOV     #T38MS2,-(SP)
      067310  012746  000001                                                    MOV     #1,-(SP)
      067314  010600                                                            MOV     SP,R0
      067316  104417                                                            TRAP    C$PNTF
      067320  062706  000004                                                    ADD     #4,SP
 6003 067324  004737  015774            JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 6004 067330  103405                    BCS     200$                    ;BR IF SOFT INIT = OK
 6008 067332  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6009 067334                            ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      067334  104455                                                            TRAP    C$ERDF
      067336  001755                                                            .WORD   1005
      067340  003652                                                            .WORD   SFIERR
      067342  012034                                                            .WORD   SFIMSG
 6010 067344                      200$:
 6011 067344  013737  002202  072150        MOV     UNITN,T38DSW            ;SET UNIT NUMBER
 6012 067352  012704  072130            MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
 6013 067356  004737  010662            JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 6014 067362  103405                    BCS     210$                    ;BR, IF COMMAND ISSUED OK
 6018 067364  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6019 067366                            ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      067366  104456                                                            TRAP    C$ERHRD
      067370  001756                                                            .WORD   1006
      067372  005056                                                            .WORD   WRTMSG
      067374  012034                                                            .WORD   SFIMSG
 6020                             ;*****************************************************************
 6021                             ;
 6022                             ;       THIS WRITE SUB-SYSTEM MEMORY COMMAND JUST HOLDS THE LEDS OFF
 6023                             ;
 6024                             ;*****************************************************************
 6025 067376                      210$:
 6026 067376  112737  000000  071441        MOVB    #0,T38S1                ;CLEAR BIT #4
 6027 067404  112737  000025  071440        MOVB    #25,T38S0               ;STOP DRIVE TEST 22
 6028 067412  012704  071430            MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
 6029 067416  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 6030 067422  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 6031 067426  103405                    BCS     250$                    ;BR IF CARRY SET (GOOD RETURN)
 6032 067430  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6036 067432                            ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      067432  104455                                                            TRAP    C$ERDF
      067434  001757                                                            .WORD   1007
      067436  072626                                                            .WORD   T38SSR
      067440  012046                                                            .WORD   PKTSSR
 6037 067442                      250$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      067442  104406                                                            TRAP    C$CLP1
 6038 067444                            SETPRI  #PRI07                  ;RAISE THE PRIORITY
      067444  012700  000340                                                    MOV     #PRI07,R0
      067450  104441                                                            TRAP    C$SPRI
```

```
6039 067452  005037  071414                    CLR     TTION2          ;ASSUME INTERRUPTS ARE ENABLED
6040 067456  032737  000100  177560            BIT     #100,@#TTICSR   ;ARE TTI INTERRUPTS ON ?
6041 067464  001005                            BNE     710$            ;BRANCH IF YES
6042 067466  005237  071414                    INC     TTION2          ;FLAG SET IF INTERRUPTS OFF
6043 067472  052737  000100  177560            BIS     #100,@#TTICSR   ;ENABLE INTERRUPTS
6044 067500  012701  000060          710$:     MOV     #TTIVEC,R1      ;START OF TTI VECTORS
6045 067504  011137  071416                    MOV     (R1),TVSAV2     ;SAVE THE CURRENT TTI VECTOR
6046 067510  012721  070710                    MOV     #590$,(R1)+     ;SET NEW INTERRUPT ROUTINE
6047 067514  011137  071420                    MOV     (R1),TPSAV2     ;SAVE THE VECTOR PRIORITY
6048 067520  012711  000340                    MOV     #PRI07,(R1)     ;USE PRIORITY SEVEN
6049 067524                                    SETPRI  #PRI00          ;LOWER INTERRUPT BR LEVEL
     067524  012700  000000                                                    MOV     #PRI00,R0
     067530  104441                                                           TRAP    C$SPRI
6050 067532  000240          260$:     NOP                             ;ALLOW CNTL C
6051 067534  000776                    BR      260$                    ;LOOP UNTIL STOPPED
6052
6053
6054 067536                    20$:     PRINTF  #T38MS2                 ;TELL'EM WHAT TO TYPE
     067536  012746  073221                                                    MOV     #T38MS2,-(SP)
     067542  012746  000001                                                    MOV     #1,-(SP)
     067546  010600                                                           MOV     SP,R0
     067550  104417                                                           TRAP    C$PNTF
     067552  062706  000004                                                    ADD     #4,SP
6055 067556                                    SETPRI  #PRI00          ;LOWER PRIORITY TO ALLOW INTERRUPTS
     067556  012700  000000                                                    MOV     #PRI00,R0
     067562  104441                                                           TRAP    C$SPRI
6056 067564  005037  002224                    CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
6057 067570  004737  015774                    JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
6058 067574  103405                            BCS     300$            ;BR IF SOFT INIT = OK
6062 067576  010001                            MOV     R0,R1           ;SAVE CONTENTS OF TSSR
6063 067600                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     067600  104455                                                           TRAP    C$ERDF
     067602  001760                                                           .WORD   1008
     067604  003652                                                           .WORD   SFIERR
     067606  012034                                                           .WORD   SFIMSG
6064 067610                    300$:
6065 067610  013737  002202  072150            MOV     UNITN,T38DSW    ;SET UNIT NUMBER IN PACKET
6066 067616  012737  000040  072146            MOV     #BIT5,T38EAI    ;ENABLE ATTENTION INTERRUPTS
6067 067624  012704  072130                    MOV     #T38PK2,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
6068 067630  004737  010662                    JSR     PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
6069 067634  103405                            BCS     310$            ;BR. IF COMMAND ISSUED OK
6073 067636  010001                            MOV     R0,R1           ;SAVE CONTENTS OF TSSR
6074 067640                                    ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     067640  104456                                                           TRAP    C$ERHRD
     067642  001761                                                           .WORD   1009
     067644  005056                                                           .WORD   WRTMSG
     067646  012034                                                           .WORD   SFIMSG
6075 067650                    310$:
6076 067650  012704  072160                    MOV     #T38PK3,R4      ;SET UP NEW PACKET FOR MESS BUF REL
6077 067654  010465  000000                    MOV     R4,TSDB(R5)     ;MESSAGE BUFFER RELEASE,ACK,CVC=1 CMD
6078 067660  004737  016250                    JSR     PC,WAITF        ;WAIT FOR SSR TO SET
6079 067664  005002                            CLR     R2              ;MAKE SURE ALL IS CLEAR
6080 067666  016501  000002                    MOV     TSSR(R5),R1     ;GET TSSR STATUS
6081 067672  032701  000100                    BIT     #OFL,R1         ;IS OFL SET
6082 067676  001402                            BEQ     320$            ;BR. IF OFL IS NOT SET
6083 067700  052702  000100                    BIS     #OFL,R2         ;SET OFL IN EXPECTED
6084 067704  052702  000200          320$:     BIS     #SSR,R2         ;SET UP EXPECTED
```

```
6085 067710  020201                    CMP     R2,R1                   ;IS EVERYTHING OK
6086 067712  001404                    BEQ     350$                    ;BR, IF ALL IS WELL
6090 067714                            ERRHRD  ERRNO,T38SST,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     067714  104456                                                            TRAP    C$ERHRD
     067716  001762                                                            .WORD   1010
     067720  073036                                                            .WORD   T38SST
     067722  012046                                                            .WORD   PKTSSR
6091 067724                    350$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     067724  104406                                                            TRAP    C$CLP1
6092 067726                            PRINTF  @T38MS1                 ;TELL OPERATOR TO TOGGLE SWITCH
     067726  012746  073126                                                    MOV     @T38MS1,-(SP)
     067732  012746  000001                                                    MOV     #1,-(SP)
     067736  010600                                                            MOV     SP,R0
     067740  104417                                                            TRAP    C$PNTF
     067742  062706  000004                                                    ADD     #4,SP
6093 067746                            PRINTF  @T38MS2                 ;TELL OPERATOR TO DO ^C TO EXIT
     067746  012746  073221                                                    MOV     @T38MS2,-(SP)
     067752  012746  000001                                                    MOV     #1,-(SP)
     067756  010600                                                            MOV     SP,R0
     067760  104417                                                            TRAP    C$PNTF
     067762  062706  000004                                                    ADD     #4,SP
6094 067766                            SETPRI  #PRI07                  ;RAISE THE PRIORITY
     067766  012700  000340                                                    MOV     #PRI07,R0
     067772  104441                                                            TRAP    C$SPRI
6095 067774  005037  071414            CLR     TTION2                  ;ASSUME INTERRUPTS ARE ENABLED
6096 070000  032737  000100  177560    BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
6097 070006  001005                    BNE     720$                    ;BRANCH IF YES
6098 070010  005237  071414            INC     TTION2                  ;FLAG SET IF INTERRUPTS OFF
6099 070014  052737  000100  177560    BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
6100 070022  012701  000060    720$:   MOV     #TTIVEC,R1              ;START OF TTI VECTORS
6101 070026  011137  071416            MOV     (R1),TVSAV2             ;SAVE THE CURRENT TTI VECTOR
6102 070032  012721  070710            MOV     #590$,(R1)+             ;SET NEW INTERRUPT ROUTINE
6103 070036  011137  071420            MOV     (R1),TPSAV2             ;SAVE THE VECTOR PRIORITY
6104 070042  012711  000340            MOV     #PRI07,(R1)             ;USE PRIORITY SEVEN
6105 070046                            SETPRI  #PRI00                  ;LOWER INTERRUPT BR LEVEL
     070046  012700  000000                                                    MOV     #PRI00,R0
     070052  104441                                                            TRAP    C$SPRI
6106 070054  000240            360$:   NOP                             ;ALLOW CONTROL C
6107 070056  005737  002224            TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
6108 070062  001001                    BNE     370$                    ;BRANCH IF YES
6109 070064  000773                    BR      360$                    ;WAIT SOME MORE FOR INTERRUPT
6110 070066                    370$:   PRINTF  @T38INT                 ;"INTERRUPT RECEIVED"
     070066  012746  072716                                                    MOV     @T38INT,-(SP)
     070072  012746  000001                                                    MOV     #1,-(SP)
     070076  010600                                                            MOV     SP,R0
     070100  104417                                                            TRAP    C$PNTF
     070102  062706  000004                                                    ADD     #4,SP
6111 070106  016501  000002            MOV     TSSR(R5),R1             ;READ TSSR STATUS
6112 070112  032701  000100            BIT     #OFL,R1                 ;CHECK THE OFF-LINE BIT
6113 070116  001011                    BNE     380$                    ;BR, IF DRIVE IS OFF-LINE
6114 070120                            PRINTF  @T38ONL                 ;"DRIVE IS NOW ON-LINE"
     070120  012746  072746                                                    MOV     @T38ONL,-(SP)
     070124  012746  000001                                                    MOV     #1,-(SP)
     070130  010600                                                            MOV     SP,R0
     070132  104417                                                            TRAP    C$PNTF
     070134  062706  000004                                                    ADD     #4,SP
6115 070140  000410                    BR      390$                    ;ALMOST DONE
```

```
6116 070142                         380$:   PRINTF  #T380FL              ;"DRIVE IS NOW OFF-LINE'
     070142  012746  073002                                                                    MOV     #T380FL,-(SP)
     070146  012746  000001                                                                    MOV     #1,-(SP)
     070152  010600                                                                            MOV     SP,R0
     070154  104417                                                                            TRAP    C$PNTF
     070156  062706  000004                                                                    ADC     #4,SP
6117 070162  005037  002224         390$:   CLR     INTRECV              ;CLEAR INTERRUPT FLAG
6118 070166  000137  067610                 JMP     300$                 ;TRY AGAIN
6119 070172                         25$:    GMANIL  T38MSG,T38DAT, 1,NO  ;WAIT FOR OPERATOR TO MOUNT TAPE
     070172  104443                                                                            TRAP    C$GMAN
     070174  000404                                                                            BR      10000$
     070176  073754                                                                            .WORD   T38DAT
     070200  000120                                                                            .WORD   T$CODE
     070202  073265                                                                            .WORD   T38MSG
     070204  177777                                                                            .WORD   -1
     070206                                                                          10000$:
6120 070206                                 BNCOMPLETE 25$               ;RETRY IF ERROR
     070206  103371                                                                            BCC     25$
6121 070210  005737  073754                 TST     T38DAT               ;DID OPERATOR SAY 'YES' ?
6122 070214  001002                         BNE     27$                  ;BRANCH IF YES
6123 070216  000137  066662                 JMP     2$                   ;RETURN TO MAIN MENU
6124 070222                         27$:
6125 070222  004737  015774                 JSR     PC,SOFINIT           ;DO SOFT INIT OF CONTROLLER
6126 070226  103405                         BCS     400$                 ;BR IF SOFT INIT = OK
6130 070230  010001                         MOV     R0,R1                ;SAVE CONTENTS OF TSSR
6131 070232                                 ERRDF   ERRNO,SFIERR,SFIMSG  ;DEVICE FATAL ERROR DURING INIT
     070232  104455                                                                            TRAP    C$ERDF
     070234  001763                                                                            .WORD   1011
     070236  003652                                                                            .WORD   SFIERR
     070240  012034                                                                            .WORD   SFIMSG
6132 070242                         400$:   CKLOOP                       ;LOOP IF SELECTED
     070242  104406                                                                            TRAP    C$CLP1
6133 070244  013737  002202  072150         MOV     UNITN,T38DSW         ;SET UNIT NUMBER
6134 070252  012704  072130                 MOV     #T38PK2,R4           ;SUBROUTINE NEEDS PACKET ADDRESS
6135 070256  004737  010662                 JSR     PC,WRTCHR            ;ISSUE WRITE CHARACTERISTICS
6136 070262  103405                         BCS     410$                 ;BR, IF COMMAND ISSUED OK
6140 070264  010001                         MOV     R0,R1                ;SAVE CONTENTS OF TSSR
6141 070266                                 ERRHRD  ERRNO,WRTMSG,SFIMSG  ;WRITE CHARACTERISTISC FAILED
     070266  104456                                                                            TRAP    C$ERHRD
     070270  001764                                                                            .WORD   1012
     070272  005056                                                                            .WORD   WRTMSG
     070274  012034                                                                            .WORD   SFIMSG
6142 070276                         410$:   CKLOOP                       ;LOOP IF SELECTED
     070276  104406                                                                            TRAP    C$CLP1
6143 070300  013701  071454                 MOV     T38BFR+6,R1          ;PICK UP XSTO CONTENTS
6144 070304  010102                         MOV     R1,R2                ;SET UP EXPECTED
6145 070306  052702  000004                 BIS     #BIT2,R2             ;SET UP THE WRITE LOCKED BIT
6146 070312  020102                         CMP     R1,R2                ;ARE THEY CORRECT
6147 070314  001406                         BEQ     430$                 ;BR, IF ALL IS WELL (OK)
6151 070316                                 ERRHRD  ERRNO,T38WRL,EXPREC  ;"WRITE LOCKED BIT IS NOT SET ETC."
     070316  104456                                                                            TRAP    C$ERHRD
     070320  001765                                                                            .WORD   1013
     070322  072444                                                                            .WORD   T38WRL
     070324  015474                                                                            .WORD   EXPREC
6152 070326  005237  002222                 INC     FATFLG               ;SET FATAL FLAG
6153 070332                         430$:   CKLOOP                       ;LOOP IF SELECTED
     070332  104406                                                                            TRAP    C$CLP1
```

```
6154 070334  005737  002222                TST     FATFLG                  ;WAS THE DRIVE NOT WRITE LOCKED
6155 070340  001402                        BEQ     435$                    ;BR, IF FLAG NOT SET
6156 070342  000137  066662                JMP     2$                      ;RE-WRITE MENU
6157 070346  017737  112552  072202  435$: MOV     @FREE,T38WR             ;SET UP WRITE BUFFER ADDRESS
6158 070354  012704  072200                MOV     #T38PK4,R4              ;GET PACKET ADDRESS
6159 070360  010465  000000                MOV     R4,TSD8(R5)             ;SET THE PACKET ADDRESS
6160 070364  004737  016250                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
6161 070370  016501  000002                MOV     TSSR(R5),R1             ;GET TSSR
6162 070374  012702  100206                MOV     #SC!SSR!BIT1!BIT2,R2    ;SET UP EXPECTED
6163 070400  020102                        CMP     R1,R2                   ;ARE THEY EQUAL (CORRECT)
6164 070402  001404                        BEQ     440$                    ;BR, IF CORRECT STATUS
6168 070404                                ERRHRD  ERRNO,T38WRT,PKTSSR     ;"TSSR INCORRECT AFTER WRITE COMMAND
     070404  104456                                                        TRAP    C$ERHRD
     070406  001766                                                        .WORD   1014
     070410  072360                                                        .WORD   T38WRT
     070412  012046                                                        .WORD   PKTSSR
6169 070414                        440$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     070414  104406                                                        TRAP    C$CLP1
6170 070416  013701  071454                MOV     T38BFR+6,R1             ;READ XSTO CONTENTS
6171 070422  010102                        MOV     R1,R2                   ;SET UPR EXPECTED
6172 070424  052702  004000                BIS     #BIT11,R2               ;SET THE WRITE LOCK ERROR BIT (XSTO)
6173 070430  020102                        CMP     R1,R2                   ;WAS THE BIT SET
6174 070432  001404                        BEQ     450$                    ;BR, IF IT WAS (GOOD)
6178 070434                                ERRHRD  ERRNO,T38WLE,EXPREC     ;"WRITE LOCK ERROR BIT NOT SET"
     070434  104456                                                        TRAP    C$ERHRD
     070436  001767                                                        .WORD   1015
     070440  072505                                                        .WORD   T38WLE
     070442  015474                                                        .WORD   EXPREC
6179 070444                        450$:   CKLOOP                          ;LOOP IF SELECTED
     070444  104406                                                        TRAP    C$CLP1
6180 070446  000137  066662                JMP     2$                      ;GO BACK TO MENU
6181
6182                               ;*************************************************************************
6183                               ;       SERVO EXERCISER NO LONGER USED
6184                               ;*************************************************************************
6185 070452                        30$:
6186 070452                                PRINTB  #T38MS3                 ;"EXE ANY OTHER MENU SELECTION TO STOP
     070452  012746  072265                                                 MOV     #T38MS3,-(SP)
     070456  012746  000001                                                 MOV     #1,-(SP)
     070462  010600                                                         MOV     SP,R0
     070464  104414                                                         TRAP    C$PNTB
     070466  062706  000004                                                 ADD     #4,SP
6187 070472  004737  073756                JSR     PC,T38REST              ;SET PACKET TO INITIAL VALUES
6188 070476  004737  015774                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
6189 070502  103405                        BCS     500$                    ;BR IF SOFT INIT = OK
6193 070504  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6194 070506                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     070506  104455                                                        TRAP    C$ERDF
     070510  001770                                                        .WORD   1016
     070512  003652                                                        .WORD   SFIERR
     070514  012034                                                        .WORD   SFIMSG
6195 070516  013737  002202  072150  500$: MOV     UNITN,T38DSW            ;SET UNIT NUMBER
6196 070524  012704  072130                MOV     #T38PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
6197 070530  004737  010662                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
6198 070534  103405                        BCS     510$                    ;BR, IF COMMAND ISSUED OK
6202 070536  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6203 070540                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
```

```
        070540  104456                                                          TRAP    C$ERHRD
        070542  001771                                                          .WORD   1017
        070544  005056                                                          .WORD   WRTMSG
        070546  012034                                                          .WORD   SFIMSG
  6204  070550                          510$:
  6205  070550  112737  000000  071441        MOVB    #0,T38S1                ;CLEAR BIT #4
  6206  070556  112737  000020  071440        MOVB    #20,T38S0               ;EXECUTE DRIVE TEST 22
  6207  070564  012704  071430              MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
  6208  070570  010465  000000              MOV     R4,TSD8(R5)             ;SET THE PACKET ADDRESS
  6209  070574  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
  6210  070600  103405                      BCS     550$                    ;BR IF CARRY SET (GOOD RETURN)
  6211  070602  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  6215  070604                              ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        070604  104455                                                          TRAP    C$ERDF
        070606  001772                                                          .WORD   1018
        070610  072626                                                          .WORD   T38SSR
        070612  012046                                                          .WORD   PKTSSR
  6216  070614                          550$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        070614  104406                                                          TRAP    C$CLP1
  6217  070616                              SETPRI  #PRI07                  ;RAISE THE PRIORITY
        070616  012700  000340                                                  MOV     #PRI07,R0
        070622  104441                                                          TRAP    C$SPRI
  6218  070624  005037  071414              CLR     TTION2                  ;ASSUME INTERRUPTS ARE ENABLED
  6219  070630  032737  000100  177560      BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
  6220  070636  001005                      BNE     555$                    ;BRANCH IF YES
  6221  070640  005237  071414              INC     TTION2                  ;FLAG SET IF INTERRUPTS OFF
  6222  070644  052737  000100  177560      BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
  6223  070652  012701  000060          555$:   MOV     #TTIVEC,R1              ;START OF TTI VECTORS
  6224  070656  011137  071416              MOV     (R1),TVSAV2             ;SAVE THE CURRENT TTI VECTOR
  6225  070662  012721  070710              MOV     #590$,(R1)+             ;SET NEW INTERRUPT ROUTINE
  6226  070666  011137  071420              MOV     (R1),TPSAV2             ;SAVE THE VECTOR PRIORITY
  6227  070672  012711  000340              MOV     #PRI07,(R1)             ;USE PRIORITY SEVEN
  6228  070676                              SETPRI  #PRI00                  ;LOWER INTERRUPT BR LEVEL
        070676  012700  000000                                                  MOV     #PRI00,R0
        070702  104441                                                          TRAP    C$SPRI
  6229  070704  000240                  560$:   NOP                             ;LOOP AWHILE
  6230  070706  000776                      BR      560$                    ;STAY IN "TIGHT" LOOP
  6231                                  ;+
  6232                                  ;PROCESS CONSOLE INTERRUPTS
  6233                                  ;-
  6234
  6235  070710  010046                  590$:   MOV     R0,-(SP)                ;SAVE WORK REGISTER
  6236  070712  113700  177562              MOVB    @#TTIBFR,R0             ;GET THE OPERATOR INPUT
  6237  070716  042700  000200              BIC     #200,R0                 ;STRIP OFF PARITY BIT
  6238  070722  122700  000015              CMPB    #15,R0                  ;IS IT A CARRIAGE RETURN ?
  6239  070726  001075                      BNE     591$                    ;JUST EXIT IF NOT
  6240  070730  012766  066662  000002      MOV     #2$,2(SP)               ;RETURN TO MASTER MENU
  6241  070736  005066  000004              CLR     4(SP)                   ;FORCE PRIORTY 0
  6242  070742  013737  071416  000060      MOV     TVSAV2,@#TTIVEC ;RESTORE VECTOR
  6243  070750  013737  071420  000062      MOV     TPSAV2,@#TTIVEC+2       ;RESTORE SUPER PRIORTY
  6244  070756  112737  000025  071440      MOVB    #25,T38S0               ;STOP DRIVE TEST 22
  6245  070764  112737  000000  071441      MOVB    #0,T38S1                ;CLEAR BS1
  6246  070772  012704  071430              MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
  6247  070776  010465  000000              MOV     R4,TSD8(R5)             ;SET THE PACKET ADDRESS
  6248  071002  012737  176750  071422      MOV     #65000.,T38DLY          ;SET UP DELAY COUNTER
  6249  071010  004737  016250          592$:   JSR     PC,WAITF                ;DO A WAIT FOR SSR
  6250  071014  016501  000002              MOV     TSSR(R5),R1             ;CONTENTS OF TSSR REGISTER
```

```
6251 071020  032701  000200            BIT     #SSR,R1          ;CHECK FOR TSSR SET
6252 071024  00101?                    BNE     595$             ;KEEP GOING IF NOT SET
6253 071026                            DELAY   250              ;CALL DELAY ROUTINE
     071026  012727  000250                                                MOV     #250,(PC)+
     071032  000000                                                        .WORD   0
     071034  013727  002116                                                MOV     L$DLY,(PC)+
     071040  000000                                                        .WORD   0
     071042  005367  177772                                                DEC     -6(PC)
     071046  001375                                                        BNE     .-4
     071050  005367  177756                                                DEC     -22(PC)
     071054  001367                                                        BNE     .-20
6254 071056  005337  071422            DEC     T38DLY           ;BUMP COUNTER DOWN
6255 071062  001352                    BNE     592$             ;BR, IF MORE TIME LEFT
6256 071064  004737  016336  595$:     JSR     PC,CHKTSSR       ;WAIT FOR SSR TO SET
6257 071070  103405                    BCS     580$             ;BR IF CARRY SET (GOOD RETURN)
6258 071072  010001                    MOV     R0,R1            ;SAVE CONTENTS OF TSSR
6262 071074                            ERRDF   ERRNO,T38SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
     071074  104455                                                        TRAP    C$ERDF
     071076  001773                                                        .WORD   1019
     071100  072626                                                        .WORD   T38SSR
     071102  012046                                                        .WORD   PKTSSR
6263 071104                    580$:   CKLOOP                   ;LOOP ON ERROR, IF FLAG SET
     071104  104406                                                        TRAP    C$CLP1
6264 071106  005737  071414            TST     TTION2           ;ARE SUPER INTERRUPTS ENABLED
6265 071112  001403                    BEQ     591$             ;BR, IF YES
6266 071114  042737  000100  177560    BIC     #100,@#TTICSR    ;RESTORE REGISTER
6267 071122  012600            591$:   MOV     (SP)+,R0         ;RESTORE REGISTER
6268 071124  000002                    RTI                      ;RETURN
6269 071126                    35$:
6270 071126  004737  073756            JSR     PC,T38REST       ;SET PACKET TO INITIAL VALUES
6271 071132  004737  015774            JSR     PC,SOFINIT       ;DO SOFT INIT OF CONTROLLER
6272 071136  103405                    BCS     600$             ;BR IF SOFT INIT = OK
6276 071140  010001                    MOV     R0,R1            ;SAVE CONTENTS OF TSSR
6277 071142                            ERRDF   ERRNO,SFIERR,SFIMSG  ;DEVICE FATAL ERROR DURING INIT
     071142  104455                                                        TRAP    C$ERDF
     071144  001774                                                        .WORD   1020
     071146  003652                                                        .WORD   SFIERR
     071150  012034                                                        .WORD   SFIMSG
6278 071152                    600$:   CKLOOP                   ;LOOP IF SELECTED
     071152  104406                                                        TRAP    C$CLP1
6279 071154  012701  071446            MOV     #T38BFR,R1       ;ADDRESS OF MESSAGE BUFFER
6280 071160  012702  125252            MOV     #125252,R2       ;ALTERNATING 1'S AND 0'S
6281
6282 071164  010221                    MOV     R2,(R1)+         ;CLEAR OUT THE MESSAGE BUFFER
6283 071166  022701  072122            CMP     #T38EB,R1        ;END OF BUFFER YET
6284 071172  001401                    BEQ     605$             ;BR, IF AT END OF BUFFER
6285 071174  000773                    BR      601$             ;NOT AT END KEEP GOING
6286 071176  013737  002202  072150  605$:   MOV     UNITN,T38DSW     ;SET UNIT NUMBER
6287 071204  012704  072130            MOV     #T38PK2,R4       ;SUBROUTINE NEEDS PACKET ADDRESS
6288 071210  004737  010662            JSR     PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
6289 071214  103405                    BCS     610$             ;BR, IF COMMAND ISSUED OK
6293 071216  010001                    MOV     R0,R1            ;SAVE CONTENTS OF TSSR
6294 071220                            ERRHRD  ERRNO,WRTMSG,SFIMSG  ;WRITE CHARACTERISTISC FAILED
     071220  104456                                                        TRAP    C$ERHRD
     071222  001775                                                        .WORD   1021
     071224  005056                                                        .WORD   WRTMSG
     071226  012034                                                        .WORD   SFIMSG
```

```
6295 071230                          610$:  CKLOOP                          ;LOOP IF SELECTED
     071230  104406                                                                 TRAP    C$CLP1
6296 071232  112737  000000  071441         MOVB    #0,T38BS1               ;CLEAR BIT #4
6297 071240  112737  000024  071440         MOVB    #24,T38BSO              ;READ EXTENDED DRIVE STATUS
6298 071246  012704  071430                 MOV     #T38PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
6299 071252  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
6300 071256  012737  000144  071422         MOV     #100.,T38DLY            ;SET UP DELAY ROUTINE
6301 071264  004737  016250         620$:   JSR     PC,WAITF                ;WAIT AWHILE FOR SSR TO SET
6302 071270  016501  000002                 MOV     TSSR(R5),R1             ;SEE IF IT REALLY DID
6303 071274  032701  000200                 BIT     #SSR,R1                 ;JUST CHECK THAT BIT
6304 071300  001017                         BNE     630$                    ;BR, IF SSR IS SET
6305 071302                                 DELAY   250                     ;DELAY ABOUT .25 SEC
     071302  012727  000250                                                         MOV     #250.,(PC)+
     071306  000000                                                                 .WORD   0
     071310  013727  002116                                                         MOV     L$DLY,(PC)+
     071314  000000                                                                 .WORD   0
     071316  005367  177772                                                         DEC     -6(PC)
     071322  001375                                                                 BNE     .-4
     071324  005367  177756                                                         DEC     -22(PC)
     071330  001367                                                                 BNE     .-20
6306 071332  005337  071422                 DEC     T38DLY                  ;START DELAY COUNT DOWN
6307 071336  001352                         BNE     620$                    ;BR, IF COUNTER IS NOT AT DONE
6308 071340  004737  016336         630$:   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6309 071344  103405                         BCS     650$                    ;BR IF CARRY SET (GOOD RETURN)
6310 071346  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6314 071350                                 ERRDF   ERRNO,T38SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     071350  104455                                                                 TRAP    C$ERDF
     071352  001776                                                                 .WORD   1022
     071354  072626                                                                 .WORD   T38SSR
     071356  012046                                                                 .WORD   PKTSSR
6315 071360                          650$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     071360  104406                                                                 TRAP    C$CLP1
6316 071362  012700  071466                 MOV     #T38BFR+20,R0           ;MESSAGE BUFFER ADDRESS
6317 071366  005001                         CLR     R1                      ;NO HIGH ORDER ADDRESS BITS
6318 071370  005037  003134                 CLR     KTENABLE                ;NO KT11 STUFF EITHER
6319 071374  004737  074014                 JSR     PC,T38MBP               ;GO PRINT MESSAGE BUFFER CONTENTS
6320 071400  000137  066662                 JMP     2$                      ;GO BACK TO MENU
6321
6322
6323 071404  000137  000200         63$:    JMP     200                     ;REALLY RETURN TO THE SUPERVISOR
6324 071410                          64$:    EXIT    TST                     ;LEAVE TEST
     071410  104432                                                                 TRAP    C$EXIT
     071412  003054                                                                 .WORD   L10075
6325
6326                                 ;+
6327                                 ;LOCAL TEXT MESSAGES FOR TEST
6328                                 ;-
6329
6330                                 ;LOCAL STORAGE FOR THIS TEST
6331                                 ;
6332                                 ;+
6333                                 ;LOCAL STORAGE FOR THIS TEST
6334                                 ;-
6335
6336 071414  000000                 TTION2:         .WORD   0               ;WORD SET IF SUPERVISOR TTI INTER OFF
6337 071416  000000                 TVSAV2: .WORD   0               ;SAVE TTI VECTOR
6338 071420  000000                 TPSAV2: .WORD   0               ;SAVE TTI PRIORITY
```

```
                                
071422  000000          T38DLY: .WORD   0                       ;DELAY COUNTER FOR TEST
        071430                  .=<.+10>&177770
071430                  T38PACKET:                              ;COMMAND PACKET FOR TEST
071430  140006                  .WORD   140006                  ;WRITE SUBSYSTEM MEM. CMD.  ACK,CVC=1
071432  071440                  .WORD   T38TAD                  ;ADDRESS OF CHARACTERISTICS BLOCK
071434  000000                  .WORD   0
071436  000012                  .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE
071440                  T38TAD:                         ;CHARACTERISTICS DATA BLOCK
071440     000          T38BS0: .BYTE   0                       ;BSEL0 BYTE
071441     000          T38BS1: .BYTE   0                       ;BSEL1 BYTE
071442  000000          T38BS2: .WORD   0                       ;BSEL1 WORD
071444  000000                  .WORD   0                       ;DATA
071446          T38BFR: .BLKW   150.                    ;MESSAGE BUFFER
072122  000000          T38EB:  .WORD                           ;END OF BUFFER ADDRESS


        072130                  .=<.+10>&177770
072130                  T38PK2:                                 ;COMMAND PACKET FOR TEST
072130  140004                  .WORD   140004                  ;WRITE CHARA. MEM. CMND.,  ACK,CVC=1
072132  072140                  .WORD   T38DTA                  ;ADDRESS OF SELECT DATA BLOCK
072134  000000                  .WORD   0
072136  000012                  .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE


072140                  T38DTA:                                 ;SELECT DATA BLOCK
072140  071446                  .WORD   T38BFR                  ;ADDRESS OF MESSAGE BUFFER
072142  000000                  .WORD   0
072144  000400                  .WORD   256.                    ;LENGTH OF MESSAGE BUFFER
072146  000000          T38EAI: .WORD   0                       ;EAI BIT WORD
072150  000000          T38DSW: .WORD   0                       ;DRIVE SELECT WORD ETC
        072160                  .=<.+10>&177770
072160  140212          T38PK3: .WORD   140212                  ;MESSAGE BUFFER RELEASE COMMAND
072162  000000                  .WORD   0                       ;NOT USED
072164  000000                  .WORD   0                       ;NOT USED
072166  000000                  .WORD   0                       ;NOT USED
072170  000000                  .WORD   0                       ;NOT USED
                        ;
                        ;WRITE TAPE PACKET
                        ;
        072200                  .=<.+10>&177770
072200  140005          T38PK4: .WORD   140005                  ;WRITE, ACK, CVC=1 COMMAND
072202  000000          T38WR:  .WORD   0                       ;ADDRESS OF WRITE BUFFER
072204  000000                  .WORD   0                       ;MORE ADDRESS OF WRITE BUFFER
072206  000400          T38SIZ: .WORD   256.                    ;SIZE OF RECORD


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

```
6404 072210  123  164  141  T38NE:  .ASCIZ  'Stand-alone Manual Intervention Not Executed'
6405 072265  045  116  045  T38MS3: .ASCIZ  'ⓂⓃⒶ Type <RETURN> To Stop Servo Exerciser, Return To Menu
6406 072360  124  123  123  T38WRT: .ASCIZ  'TSSR Not Correct After WRITE, With WRITE PROTECT On'
6407 072444  127  122  111  T38WRL: .ASCIZ  'WRITE LOCKED Bit Not Set In XSTO'
6408 072505  127  122  111  T38WLE: .ASCIZ  'WRITE LOCK ERROR Bit Not Set In XSTO'
6409 072552  127  122  111  T38NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
6410 072626  103  157  156  T38SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY
6411 072716  045  116  045  T38INT: .ASCIZ  'ⓂⓃⒶ Interrupt Received'
6412 072746  045  116  045  T38ONL: .ASCIZ  'ⓂⓃⒶ Drive Is Now   ON-LINE'
6413 073002  045  116  045  T38OFL: .ASCIZ  'ⓂⓃⒶ Drive Is Now  OFF-LINE'
6414 073036  103  157  156  T38SST: .ASCIZ  'Contents Of TSSR Incorrect After MESSAGE BUFFER RELEASE'
6415 073126  045  116  045  T38MS1: .ASCIZ  'ⓂⓃⒶToggle ON-LINE Switch to Generate ATTENTION Interrupts'
6416 073221  045  116  045  T38MS2: .ASCIZ  'ⓂⓃⒶType RETURN To Return To MenuⓂN'
6417 073265  111  163  040  T38MSG: .ASCIZ  'Is Write-Protected Tape Mounted'
6418 073325  115  141  156  T38ID:  .ASCIZ  'Manual Intervention'
6419                         .EVEN
6420 073352 073376 073450 073476 MIMENU: .WORD  1$,2$,3$,4$,5$,6$
6421 073366 073645 073710 073753         .WORD  8$,9$,10$,0
6422
6423 073376  012  123  105  1$:   .ASCIZ  <12>'SELECT OPERATION FROM FOLLOWING OPTIONS:'
6424 073450  012  011  060  2$:   .ASCIZ  <12>'    0       Display This Menu'
6425 073476  011  061  011  3$:   .ASCIZ  '       1       Turn On All M7196 LED''s'
6426 073530  011  062  011  4$:   .ASCIZ  '       2       Turn Off All M7196 LED''s'
6427 073563  011  063  011  5$:   .ASCIZ  '       3       Offline/Online Attention'
6428 073617  011  064  011  6$:   .ASCIZ  '       4       Write Protect Test'
6429 073645  011  065  011  8$:   .ASCIZ  '       5       Print Extended Transport Status'
6430 073710  011  066  011  9$:   .ASCIZ  '       6       Return to Diagnostic Supervisor'
6431 073753  000            10$:  .ASCIZ  ''
6432                         .EVEN
6433
6434                         ;*
6435                         ;LOCAL STORAGE FOR THIS TEST
6436                         ;-
6437
6438 073754 000000           T38DAT: .WORD   0               ;LOGICAL RESPONSE TO QUESTION
6439 073756                  T38REST:
6440 073756                          SAVREG                  ;SAVE THE REGISTERS
6441 073762 012701 071430            MOV     #T38PACKET,R1   ;START OF THE PACKET
6442 073766 012721 140206            MOV     #140206,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
6443 073772 012721 071440            MOV     #T38TAD,(R1)+   ;ADDRESS OF DATA BLOCK
6444 073776 005021                   CLR     (R1)+           ;EXTENDED ADDRESS
6445 074000 012721 000006            MOV     #6.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
6446 074004 005021                   CLR     (R1)+           ;CLEAR BSEL0 AND BSEL1
6447 074006 005021                   CLR     (R1)+           ;CLEAR SEL2
6448 074010 005011                   CLR     (R1)            ;CLEAR DATA AREA
6449 074012 000207                   RTS     PC              ;RETURN
6450
6451
6452                         ;*
6453                         ;
6454                         ;THIS ROUTINE PRINTS THE CONTENTS OF
6455                         ;THE 256 BYTE MESSAGE BUFFER RETURNED BY THE
6456                         ;TSV-05.
6457                         ;
6458                         ;INPUT:
6459                         ;
6460                         ;       R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
```

```
6461                                    ;       R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
6462                                    ;       NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
6463                                    ;
6464                                    ;
6465                                    ;-
6466
6467 074014                             T38MBP:
6468 074014                                     SAVREG                  ;SAVE THE REGISTERS
6469 074020  010005                             MOV     R0,R5           ;SAVE LOW ORDER ADDRESS
6470 074022  005737  003134                     TST     KTENABLE        ;ADDRESS ABOVE 28K?
6471 074026  001001                             BNE     910$            ;BR IF YES
6472 074030  005000                             CLR     R1              ;SET HIGH ORDER ADDRESS TO 0
6473 074032  010103                     910$:   MOV     R1,R3           ;SAVE HIGH ORDER ADDRESS
6474 074034  006100                             ROL     R0              ;SHIFT BIT15 TO C BIT
6475 074036  006101                             ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
6476 074040                                     PRINTX  #T38AS0,R1,R5   ;PRINT MESSAGE BUFFER ADDRESS
     074040  010546                                                                 MOV     R5,-(SP)
     074042  010146                                                                 MOV     R1,-(SP)
     074044  012746  074316                                                         MOV     #T38AS0,-(SP)
     074050  012746  000003                                                         MOV     #3,-(SP)
     074054  010600                                                                 MOV     SP,R0
     074056  104415                                                                 TRAP    C$PNTX
     074060  062706  000010                                                         ADD     #10,SP
6477 074064                                     PRINTX  #T38AS1         ;PRINT HEADER FOR CONTENTS
     074064  012746  074363                                                         MOV     #T38AS1,-(SP)
     074070  012746  000001                                                         MOV     #1,-(SP)
     074074  010600                                                                 MOV     SP,R0
     074076  104415                                                                 TRAP    C$PNTX
     074100  062706  000004                                                         ADD     #4,SP
6478 074104  010501                             MOV     R5,R1           ;COPY LOW ORDER ADDRESS
6479 074106  010300                             MOV     R3,R0           ;COPY HIGH ORDER ADDRESS
6480 074110  001403                             BEQ     913$            ;BR IF NOT ABOVE 28K
6481 074112  004737  017316                     JSR     PC,SETMAP       ;SETUP PAR ADDRESS IN R0
6482 074116  010005                             MOV     R0,R5           ;GET PAR FORMAT ADDRESS ABOVE 28K
6483 074120  010537  074464             913$:   MOV     R5,T38CNT       ;HOLD ADDRESS
6484 074124  011504                     911$:   MOV     (R5),R4         ;GET BUFFER ENTRY
6485 074126  022704  125252                     CMP     #125252,R4      ;CHECK FOR NO LOAD CONDITION
6486 074132  001417                             BEQ     912$            ;BR, IF BUFFER WASN'T LOADED
6487 074134  010403                             MOV     R4,R3           ;MAKE COPY
6488 074136  042704  170377                     BIC     #170377,R4      ;ONLY BITS 11,10,9 AND 8 ARE SAVED
6489 074142  000241                             CLC                     ;CLEAR CARRY
6490 074144  006004                             ROR     R4              ;11 TO 10 BIT POSITION
6491 074146  006004                             ROR     R4              ;10 TO 9 BIT POSITION
6492 074150  006004                             ROR     R4              ;9 TO 8 BIT POSITON
6493 074152  006004                             ROR     R4              ;8 TO 7 BIT POSITION
6494 074154  042703  177760                     BIC     #177760,R3      ;ONLY BITS 3,2,1 AND 0 ARE SAVED
6495 074160  060403                             ADD     R4,R3           ;"OR'EM TOGETHER
6496 074162  010325                             MOV     R3,(R5)+        ;PUT BACK IN BUFFER
6497 074164  020527  072122                     CMP     R5,#T38EB       ;END OF BUFFER YET
6498 074170  001355                             BNE     911$            ;BR, IF NOT AT END YET
6499 074172  013705  074464             912$:   MOV     T38CNT,R5       ;PUT ADDRESS BACK
6500 074176  012704  000001                     MOV     #1,R4           ;START BYTE NUMBER AT ONE
6501 074202                             915$:   PRINTX  #T38ASN,R4,(R5)+        ;PRT MEM BUFFER W/NEWLINE
     074202  012546                                                                 MOV     (R5)+,(SP)
     074204  010446                                                                 MOV     R4,-(SP)
     074206  012746  074440                                                         MOV     #T38ASN,-(SP)
     074212  012746  000003                                                         MOV     #3,-(SP)
```

```
        074216  010600                                                          MOV     SP,R0
        074220  104415                                                          TRAP    C$PNTX
        074222  062706  000010                                                  ADD     #10,SP
  6502  074226  005037  074464          CLR     T38CNT                  ;CLEAR COUNTER
  6503  074232  000412                  BR      921$                    ;SKIP OTHER PRINT
  6504  074234                920$:     PRINTX  #T38ASC,R4,(R5)+        ;PRINT THE CONTENTS OF MEMORY BUFFER
        074234  012546                                                          MOV     (R5)+,-(SP)
        074236  010446                                                          MOV     R4,-(SP)
        074240  012746  074421                                                  MOV     #T38ASC,-(SP)
        074244  012746  000003                                                  MOV     #3,-(SP)
        074250  010600                                                          MOV     SP,R0
        074252  104415                                                          TRAP    C$PNTX
        074254  062706  000010                                                  ADD     #10,SP
  6505  074260  005237  074464          921$:   INC     T38CNT                  ;BUMP COUNTER
  6506  074264  005204                  INC     R4              ;NUMBER OF THE NEXT
  6507  074266  020427  000200          CMP     R4,#128.                ;DONE ALL YET ?
  6508  074272  003010                  BGT     50$             ;BRANCH IF ALL DONE
  6509  074274  023727  074464  000004 CMP     T38CNT,#4               ;DONE FOUR YET
  6510  074302  001401                  BEQ     925$                    ;BR, IF THREE DONE
  6511  074304  000753                  BR      920$            ;KEEP GOING
  6512  074306  005037  074464          925$:   CLR     T38CNT                  ;CLEAR COUNTER
  6513  074312  000733                  BR      915$                    ;PRINT WITH NEW LINE
  6514  074314  000207                50$:    RTS     PC              ;RETURN
  6515
  6516  074316     045     116     045 T38AS0: .ASCIZ  '<N><A Message Buffer Address = <01><05>'
  6517  074363     045     116     045 T38AS1: .ASCIZ  '<N><A Message Buffer Contents:'
  6518  074421     045     101     040 T38ASC: .ASCIZ  '<A  <D4><A: <03>'
  6519  074440     045     116     045 T38ASN: .ASCIZ  '<N><A Byte<D4><A: <03>'
  6520                                    .EVEN
  6521  074464  000000            T38CNT: .WORD                   ;COUNTER FOR PRINT
  6522  074466                          ENDTST
        074466                                                          L10075:
        074466  104401                                                         TRAP    C$ETST
  6523                                    .SBTTL  TEST 11: CONFIGURATION TYPEOUT
  6524
  6525                            ;THIS IS A STANDALONE ROUTINE THAT PRINTS OUT ON THE CONSOLE TERMINAL
  6526                            ;THE CONFIGURATION OF THE M7196 MODULE AND TSV05 SUBSYSTEM. SPECIFICALLY,
  6527                            ;THE FOLLOWING INFORMATION IS PRESENTED:
  6528                            ;
  6529                            ;
  6530                            ; 1.0 STATE OF THE EXTENDED FEATURES SWITCH ON THE M7196: ON (EXTENDED
  6531                            ;     FEATURES ENABLED) OR OFF (EXTENDED FEATURES DISABLED).
  6532                            ;
  6533                            ; 2.0 STATE OF THE BUFFERING ENABLE SWITCH ON THE M7196: ON
  6534                            ;     (BUFFERING ENABLED) OR OFF (BUFFERING DISABLED).
  6535                            ;
  6536                            ; 3.0 MICROCODE REVSION LEVEL OF THE M7196.
  6537                            ;
  6538                            ; 4.0 NUMBER OF TAPE TRANSPORTS CONNECTED TO THE CONTROLLER.
  6539                            ;
  6540                            ; 5.0 UNIT SELECT CODE AND STATE (ONLINE/OFFLINE, WRITE ENABLED/PROTECTED)
  6541                            ;     OF EACH CONNECTED TRANSPORT. IN ADDITION, THE PROGRAM WILL INDICATE,
  6542                            ;     FOR EACH ON-LINE TRANSPORT, WHETHER OR NOT IT IS EQUIPPED WITH THE
  6543                            ;     EXTENDED TAPE STATUS READOUT FEATURE.
  6544                            ;
  6545                            ;
  6546                            ;THE OPERATOR IS EXPECTED TO READ THE PRINTOUT AND VERIFY THAT IT MATCHES
```

```
6547                                    ;THE ACTUAL CONFIGURATION AT HAND.  IF, FOR EXAMPLE, THE PROGRAM INDICATES
6548                                    ;THAT IT "SEES" TWO TRANSPORTS CONNECTED WHEN IN FACT ONLY ONE IS PRESENT,
6549                                    ;THE OPERATOR MUST INTERPRET THIS AS AN ERROR AND ATTEMPT TO FIND THE
6550                                    ;CAUSE (BAD CABLE, FAULTY UNIT-SELECT DECODING IN THE TRANSPORT, ETC.).
6551                                    ;[SINCE THE CONTROLLER CAN ONLY ACCESS UNIT 0 IF IT IS IN "STANDARD"
6552                                    ;MODE, THE PROGRAM WILL FORCE THE MODULE INTO EXTENDED MODE VIA THE
6553                                    ;WRITE SUBSYSTEM MEMORY COMMAND IN ORDER TO SCAN FOR CONNECTED TRANSPORTS.]
6554                                    ;
6555                                    ;
6556                                    ;THIS ROUTINE, WHEN ITS ACTIONS ARE COMPLETED, WILL EXIT BACK TO THE
6557                                    ;DIAGNOSTIC SUPERVISOR SO THAT IF ADDITIONAL UNITS (CONTROLLERS) ARE
6558                                    ;SELECTED (E.G., FROM THE INITIAL STARTUP DIALOG), THE ROUTINE WILL BE
6559                                    ;REENTERED SO THAT THEIR CONFIGURATIONS CAN BE PRINTED.
6560                                    ;
6561 074470                                    BGNTST
      074470                                                                    T11::
6566 074470                                    RFLAGS  R0                       ;GET OPERATOR FLAGS
      074470  104421                                                                   TRAP    C$RFLA
6567 074472  001403                           BEQ     10$                       ;BR, IF OK TO RUN
6568 074474  012700  076463                   MOV     #T39NE,R0                 ;'TEST NOT EXECUTED"
6569 074500  000402                           BR      11$                       ;JUMP OUT OF TEST IF NOT
6570 074502  012700  077612          10$:     MOV     #TST39ID,R0               ;TEST ID MESSAGE
6571 074506  004737  016510          11$:     JSR     PC,TSTSETUP               ;DO THE COMMON SETUP
6572 074512  004737  020500                   JSR     PC,CHKMAN                 ;IS MANUAL INTERVENTION ALLOWED?
6573 074516  103402                           BCS     20$                       ;BR, IF MANUAL INTERVENTION ALLOWED
6574 074520  000137  075700                   JMP     64$                       ;JUMP TO OUT IF NOT
6575 074524                           20$:
6576 074524  004737  015774                   JSR     PC,SOFINIT                ;DO SOFT INIT OF CONTROLLER
6577 074530  103405                           BCS     25$                       ;BR IF SOFT INIT = OK
6581 074532  010001                           MOV     R0,R1                     ;SAVE CONTENTS OF TSSR
6582 074534                                    ERRDF   ERRNO,SFIERR,SFIMSG       ;DEVICE FATAL ERROR DURING INIT
      074534  104455                                                                   TRAP    C$ERDF
      074536  002115                                                                   .WORD   1101
      074540  003652                                                                   .WORD   SFIERR
      074542  012034                                                                   .WORD   SFIMSG
6583 074544                           25$:     CKLOOP                                  ;LOOP IF SELECTED
      074544  104406                                                                   TRAP    C$CLP1
6584 074546  013737  002202  076430           MOV     UNITN,T39DSW              ;SET UNIT NUMBER
6585 074554  012704  076410                   MOV     #T39PK2,R4                ;SUBROUTINE NEEDS PACKET ADDRESS
6586 074560  004737  010662                   JSR     PC,WRTCHR                 ;ISSUE WRITE CHARACTERISTICS
6587 074564  103405                           BCS     50$                       ;BR, IF COMMAND ISSUED OK
6591 074566  010001                           MOV     R0,R1                     ;SAVE CONTENTS OF TSSR
6592 074570                                    ERRHRD  ERRNO,WRTMSG,SFIMSG       ;WRITE CHARACTERISTISC FAILED
      074570  104456                                                                   TRAP    C$ERHRD
      074572  002116                                                                   .WORD   1102
      074574  005056                                                                   .WORD   WRTMSG
      074576  012034                                                                   .WORD   SFIMSG
6593 074600                           50$:     CKLOOP                          ;LOOP IF SELECTED
      074600  104406                                                                   TRAP    C$CLP1
6594 074602  013701  075740                   MOV     T39BFR+12,R1              ;GET XST2 STATUS FROM MESSAGE BUFFER
6595 074606                                    PRINTX  #T39SFS                   ;"STATE OF EXTENDED FEATURES SW ="
      074606  012746  077331                                                           MOV     #T39SFS,-(SP)
      074612  012746  000001                                                           MOV     #1,-(SP)
      074616  010600                                                                   MOV     SP,R0
      074620  104415                                                                   TRAP    C$PNTX
      074622  062706  000004                                                           ADD     #4,SP
6596 074626  032701  000200                   BIT     #BIT7,R1                  ;CHECK STATE OF E.F.S.
```

```
6597 074632  001011                    BNE     100$                ;BR, IF EXT. FEA. SW. IS ON
6598 074634                            PRINTX  #T390FF             ;" OFF"
     074634  012746  077455                                                MOV     #T390FF,-(SP)
     074640  012746  000001                                                MOV     #1,-(SP)
     074644  010600                                                        MOV     SP,R0
     074646  104415                                                        TRAP    C$PNTX
     074650  062706  000004                                                ADD     #4,SP
6599 074654  000410                    BR      110$                ;SKIP OTHER PRINT STATEMENT
6600 074656                    100$:   PRINTX  #T390N              ;" ON "
     074656  012746  077464                                                MOV     #T390N,-(SP)
     074662  012746  000001                                                MOV     #1,-(SP)
     074666  010600                                                        MOV     SP,R0
     074670  104415                                                        TRAP    C$PNTX
     074672  062706  000004                                                ADD     #4,SP
6601 074676                    110$:   PRINTX  #T39SBS             ;"STATE OF BUFFERING SWITCH ="
     074676  012746  077403                                                MOV     #T39SBS,-(SP)
     074702  012746  000001                                                MOV     #1,-(SP)
     074706  010600                                                        MOV     SP,R0
     074710  104415                                                        TRAP    C$PNTX
     074712  062706  000004                                                ADD     #4,SP
6602 074716  032701  000100            BIT     #BIT6,R1            ;CHECK STATE OF BUFFERING SW
6603 074722  001011                    BNE     120$                ;BR, IF BUFFERING IS ON
6604 074724                            PRINTX  #T390FF             ;" OFF"
     074724  012746  077455                                                MOV     #T390FF,-(SP)
     074730  012746  000001                                                MOV     #1,-(SP)
     074734  010600                                                        MOV     SP,R0
     074736  104415                                                        TRAP    C$PNTX
     074740  062706  000004                                                ADD     #4,SP
6605 074744  000410                    BR      130$                ;SKIP OTHER PRINT STATEMENT
6606 074746                    120$:   PRINTX  #T390N              ;" ON "
     074746  012746  077464                                                MOV     #T390N,-(SP)
     074752  012746  000001                                                MOV     #1,-(SP)
     074756  010600                                                        MOV     SP,R0
     074760  104415                                                        TRAP    C$PNTX
     074762  062706  000004                                                ADD     #4,SP
6607 074766  042701  177700            BIC     #177700,R1          ;ONLY LEAVE MICROCODE REV LEVEL
6608 074772  010137  077550            MOV     R1,T39RL            ;LOAD UP REV LEVEL
6609 074776                            PRINTX  #T39MCL,T39RL       ;"MICROCODE REVISION LEVEL =000XXX"
     074776  013746  077550                                                MOV     T39RL,-(SP)
     075002  012746  077473                                                MOV     #T39MCL,-(SP)
     075006  012746  000002                                                MOV     #2,-(SP)
     075012  010600                                                        MOV     SP,R0
     075014  104415                                                        TRAP    C$PNTX
     075016  062706  000006                                                ADD     #6,SP
6610 075022  004737  015774            JSR     PC,SOFINIT          ;DO SOFT INIT OF CONTROLLER
6611 075026  103405                    BCS     140$                ;BR IF SOFT INIT = OK
6615 075030  010001                    MOV     R0,R1               ;SAVE CONTENTS OF TSSR
6616 075032                            ERRDF   ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
     075032  104455                                                        TRAP    C$ERDF
     075034  002117                                                        .WORD   1103
     075036  003652                                                        .WORD   SFIERR
     075040  012034                                                        .WORD   SFIMSG
6617 075042                    140$:   CKLOOP                              ;LOOP IF SELECTED
     075042  104406                                                        TRAP    C$CLP1
6618 075044  013737  002202  076430    MOV     UNITN,T39DSW        ;SET UNIT NUMBER
6619 075052  012704  076410            MOV     #T39PK2,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
6620 075056  004737  010662            JSR     PC,WRTCHR           ;ISSUE WRITE CHARACTERISTICS
```

```
6621 075062  103405                    BCS     150$                    ;BR, IF COMMAND ISSUED OK
6625 075064  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6626 075066                            ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     075066  104456                                                            TRAP    C$ERHRD
     075070  002120                                                            .WORD   1104
     075072  005056                                                            .WORD   WRTMSG
     075074  012034                                                            .WORD   SFIMSG
6627 075076                    150$:   CKLOOP                          ;LOOP IF SELECTED
     075076  104406                                                            TRAP    C$CLP1
6628 075100  005737  002226            TST     EXTFEA                  ;CHECK FOR EXTENDED FEATURES SW SWITCH
6629 075104  001036                    BNE     174$                    ;BR IF SWITCH IS ON
6630 075106  112737  000200  075721    MOVB    #200,T39BS1             ;WRITE MISCELLANEOUS CONT/READ STATUS
6631 075114  112737  000010  075720    MOVB    #10,T39BSO              ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
6632 075122  012704  075710            MOV     #T39PACKET,R4           ;WRITE SUBSYS MEM PACKET
6633 075126  010465  000000            MOV     R4,TSDB(R5)             ;ISSUE COMMAND
6634 075132  004737  016336            JSR     PC,CHKTSSR              ;WAIT FOR SSR
6635 075136  103405                    BCS     160$                    ;BR, IF NO ERROR
6636 075140  010001                    MOV     R0,R1                   ;ERROR, SAVE TSSR
6640 075142                            ERRHRD  ERRNO,T39NBA,PKTSSR     ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
     075142  104456                                                            TRAP    C$ERHRD
     075144  002121                                                            .WORD   1105
     075146  077165                                                            .WORD   T39NBA
     075150  012046                                                            .WORD   PKTSSR
6641 075152                    160$:   CKLOOP                          ;LOOP IF SELECTED
     075152  104406                                                            TRAP    C$CLP1
6642 075154  012704  076410            MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
6643                                   ;*************************************************************
6644                                   ;
6645                                   ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
6646                                   ;
6647                                   ;*************************************************************
6648
6649 075160  004737  010662            JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
6650 075164  103405                    BCS     170$                    ;BR, IF COMMAND ISSUED OK
6654 075166  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6655 075170                            ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     075170  104456                                                            TRAP    C$ERHRD
     075172  002122                                                            .WORD   1106
     075174  005056                                                            .WORD   WRTMSG
     075176  012034                                                            .WORD   SFIMSG
6656 075200                    170$:   CKLOOP                          ;SCOPE LOOP
     075200  104406                                                            TRAP    C$CLP1
6657 075202  005037  002202    174$:   CLR     UNITN                   ;SET TO DRIVE 0
6658 075206  013737  002202  076430 175$:   MOV     UNITN,T39DSW            ;SET UNIT NUMBER
6659 075214  012704  076410            MOV     #T39PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
6660 075220  004737  010662            JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
6661 075224  103405                    BCS     180$                    ;BR, IF COMMAND ISSUED OK
6665 075226  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6666 075230                            ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     075230  104456                                                            TRAP    C$ERHRD
     075232  002123                                                            .WORD   1107
     075234  005056                                                            .WORD   WRTMSG
     075236  012034                                                            .WORD   SFIMSG
6667 075240                    180$:   CKLOOP                          ;LOOP IF SELECTED
     075240  104406                                                            TRAP    C$CLP1
6668
6669 075242  016501  000002    190$:   MOV     TSSR(R5),R1             ;GET TSSR STATUS
```

```
6670 075246  032701  000100              BIT     #OFL,R1             ;CHECK FOR OFF-LINE
6671 075252  001414                      BEQ     200$                ;BR, IF DRIVE IS ON-LINE
6672 075254                              PRINTX  #T390F2,UNITN       ;"DRIVE NUMBER XX IS OFF-LINE"
     075254  013746  002202                                                  MOV     UNITN,-(SP)
     075260  012746  076724                                                  MOV     #T390F2,-(SP)
     075264  012746  000002                                                  MOV     #2,-(SP)
     075270  010600                                                          MOV     SP,R0
     075272  104415                                                          TRAP    C$PNTX
     075274  062706  000006                                                  ADD     #6,SP
6673 075300  000137  075634              JMP     250$                ;DO NOT TRY TO GET ANYMORE INFO.
6674 075304                      200$:   PRINTX  #T390N2,UNITN       ;"DRIVE NUMBER XX IS ON-LINE"
     075304  013746  002202                                                  MOV     UNITN,-(SP)
     075310  012746  076770                                                  MOV     #T390N2,-(SP)
     075314  012746  000002                                                  MOV     #2,-(SP)
     075320  010600                                                          MOV     SP,R0
     075322  104415                                                          TRAP    C$PNTX
     075324  062706  000006                                                  ADD     #6,SP
6675 075330  013701  075734              MOV     T39BFR+6,R1         ;READ EXTENDED STATUS (XST0)
6676 075334  032701  000004              BIT     #BIT2,R1            ;IS DRIVE WTITE PROTECTED
6677 075340  001013                      BNE     210$                ;BR, IF WRITE PROTECTED
6678 075342                              PRINTX  #T39WPN,UNITN       ;"DRIVE NUMBER IS NOT WRT PRO"
     075342  013746  002202                                                  MOV     UNITN,-(SP)
     075346  012746  077106                                                  MOV     #T39WPN,-(SP)
     075352  012746  000002                                                  MOV     #2,-(SP)
     075356  010600                                                          MOV     SP,R0
     075360  104415                                                          TRAP    C$PNTX
     075362  062706  000006                                                  ADD     #6,SP
6679 075366  000412                      BR      220$                ;SKIP OVER
6680 075370                      210$:   PRINTX  #T39WRT,UNITN       ;"DRIVE NUMBER XX IS WRT PRO"
     075370  013746  002202                                                  MOV     UNITN,-(SP)
     075374  012746  077033                                                  MOV     #T39WRT,-(SP)
     075400  012746  000002                                                  MOV     #2,-(SP)
     075404  010600                                                          MOV     SP,R0
     075406  104415                                                          TRAP    C$PNTX
     075410  062706  000006                                                  ADD     #6,SP
6681 075414  012737  125252  076026  220$:   MOV     #125252,T39BFR+100  ;SET 1 LOC TO KNOWN VALUE
6682 075422  112737  000000  075721          MOVB    #0,T39BS1           ;EXTENDED TAPE STATUS
6683 075430  112737  000024  075720          MOVB    #24,T39BS0          ;EXTENDED TAPE STATUS
6684 075436  012704  075710                  MOV     #T39PACKET,R4       ;WRITE SUBSYS MEM PACKET
6685 075442  010465  000000                  MOV     R4,TSDB(R5)         ;ISSUE COMMAND
6686 075446  012737  000144  075704          MOV     #100.,T39DLY        ;SET UP DELAY ROUTINE
6687 075454  004737  016250          222$:   JSR     PC,WAITF            ;WAIT AWHILE FOR SSR TO SET
6688 075460  016501  000002                  MOV     TSSR(R5),R1         ;SEE IF IT REALLY DID
6689 075464  032701  000200                  BIT     #SSR,R1             ;JUST CHECK THAT BIT
6690 075470  001017                          BNE     225$                ;BR, IF SSR IS SET
6691 075472                                  DELAY   250                 ;DELAY ABOUT .25 SEC
     075472  012727  000250                                                  MOV     #250,(PC)+
     075476  000000                                                          .WORD   0
     075500  013727  002116                                                  MOV     L$DLY,(PC)+
     075504  000000                                                          .WORD   0
     075506  005367  177772                                                  DEC     -6(PC)
     075512  001375                                                          BNE     .-4
     075514  005367  177756                                                  DEC     -22(PC)
     075520  001367                                                          BNE     .-20
6692 075522  005337  075704                  DEC     T39DLY              ;START DELAY COUNT DOWN
6693 075526  001352                          BNE     222$                ;BR, IF COUNTER IS NOT AT DONE
6694 075530  004737  016336          225$:   JSR     PC,CHKTSSR          ;WAIT FOR SSR
```

```
6695 075534  103405                      BCS     230$                    ;BR, IF NO ERROR
6696 075536  010001                      MOV     R0,R1                   ;ERROR, SAVE TSSR
6700 075540                              ERRHRD  ERRNO,T39NBA,PKTSSR     ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
     075540  104456                                                                              TRAP    C$ERHRD
     075542  002124                                                                              .WORD   1108
     075544  077165                                                                              .WORD   T39NBA
     075546  012046                                                                              .WORD   PKTSSR
6701 075550                       230$:  CKLOOP                          ;LOOP IF SELECTED
     075550  104406                                                                              TRAP    C$CLP1
6702 075552  023727  076026  125252      CMP     T39BFR+100,#125252      ;DID LOC GET OVER WRITTEN
6703 075560  001013                      BNE     240$                    ;BR, IF IT DIDN'T GET ETC.
6704 075562                              PRINTX  #T39ETN,UNITN           ;"DRIVE DOESN'T HAVE EXT TAPE STATUS
     075562  013746  002202                                                                      MOV     UNITN,-(SP)
     075566  012746  076631                                                                      MOV     #T39ETN,-(SP)
     075572  012746  000002                                                                      MOV     #2,-(SP)
     075576  010600                                                                              MOV     SP,R0
     075600  104415                                                                              TRAP    C$PNTX
     075602  062706  000006                                                                      ADD     #6,SP
6705 075606  000412                      BR      250$                    ;SKIP OVER
6706 075610                       240$:  PRINTX  #T39ETS,UNITN           ;"DRIVE HAS EXT TAPE STATUS"
     075610  013746  002202                                                                      MOV     UNITN,-(SP)
     075614  012746  076542                                                                      MOV     #T39ETS,-(SP)
     075620  012746  000002                                                                      MOV     #2,-(SP)
     075624  010600                                                                              MOV     SP,R0
     075626  104415                                                                              TRAP    C$PNTX
     075630  062706  000006                                                                      ADD     #6,SP
6707 075634  005237  002202       250$:  INC     UNITN                   ;BUMP DRIVE NUMBER
6708 075640  023727  002202  000003      CMP     UNITN,#3                ;AT END OF DRIVES YET
6709 075646  001402                      BEQ     63$                     ;BR, IF NO MORE DRIVES
6710 075650  000137  075206              JMP     175$                    ;DO NEXT DRIVE
6711 075654                       63$:   PRINTX  #T39NFL                         ;NEW LINE
     075654  012746  076460                                                                      MOV     #T39NFL,-(SP)
     075660  012746  000001                                                                      MOV     #1,-(SP)
     075664  010600                                                                              MOV     SP,R0
     075666  104415                                                                              TRAP    C$PNTX
     075670  062706  000004                                                                      ADD     #4,SP
6712 075674  000137  000200              JMP     200                     ;RETURN TO SUPERVISOR
6713 075700                       64$:   EXIT    TST                     ;EXIT THIS SECTION
     075700  104432                                                                              TRAP    C$EXIT
     075702  001736                                                                              .WORD   L10076-.
6714                              ;+
6715                              ;LOCAL TEXT MESSAGES FOR TEST
6716                              ;-
6717
6718                              ;LOCAL STORAGE FOR THIS TEST
6719                              ;-
6720
6721 075704  000000               T39DLY: .WORD  0                                ;DELAY COUNTER FOR TEST
6723         075710                      .=<.+10>&177770
6725 075710                       T39PACKET:                             ;COMMAND PACKET FOR TEST
6726 075710  140006                      .WORD   140006                  ;WRITE SUBSYSTEM MEM. CMD.  ACK,CVC=1
6727 075712  075720                      .WORD   T39TAD                  ;ADDRESS OF CHARACTERISTICS BLOCK
6728 075714  000000                      .WORD   0
6729 075716  000012                      .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE
6730 075720                       T39TAD:                        ;CHARACTERISTICS DATA BLOCK
6731 075720     000               T39BS0: .BYTE   0                       ;BSEL0 BYTE
6732 075721     000               T39BS1: .BYTE   0                       ;BSEL1 BYTE
```

```
6733 075722  000000                T39BS2: .WORD   0                       ;BSEL1 WORD
6734 075724  000000                        .WORD   0                       ;DATA
6735 075726                        T39BFR: .BLKW   150.                    ;MESSAGE BUFFER
6736
6737
6739         076410                        .=<.+10>&177770
6741 076410                        T39PK2:                                 ;COMMAND PACKET FOR TEST
6742 076410  140004                        .WORD   140004                  ;WRITE CHARA. MEM. CMND.,  ACK,CVC=1
6743 076412  076420                        .WORD   T39DTA                  ;ADDRESS OF SELECT DATA BLOCK
6744 076414  000000                        .WORD   0
6745 076416  000012                        .WORD   10.                     ;STARTING VALUE OF BLOCK SIZE
6746
6747
6748 076420                        T39DTA:                                 ;SELECT DATA BLOCK
6749 076420  075726                        .WORD   T39BFR                  ;ADDRESS OF MESSAGE BUFFER
6750 076422  000000                        .WORD   0
6751 076424  000400                        .WORD   256.                    ;LENGTH OF MESSAGE BUFFER
6752 076426  000000                T39EAI: .WORD   0                       ;EAI BIT WORD
6753 076430  000000                T39DSW: .WORD   0                       ;DRIVE SELECT WORD ETC
6755         076440                        .=<.+10>&177770
6757 076440  140012                T39PK3: .WORD   140012                  ;MESSAGE BUFFER RELEASE COMMAND
6758 076442  000000                        .WORD   0                       ;NOT USED
6759                                ;
6760                                ;WRITE TAPE PACKET
6761                                ;
6763         076450                        .=<.+10>&177770
6765 076450  140005                T39PK4: .WORD   140005                  ;WRITE, ACK, CVC=1 COMMAND
6766 076452  000000                T39WR:  .WORD   0                       ;ADDRESS OF WRITE BUFFER
6767 076454  000000                        .WORD   0                       ;MORE ADDRESS OF WRITE BUFFER
6768 076456  000400                T39SIZ: .WORD   256.                    ;SIZE OF RECORD
6769
6770
6771
6772
6773                                ;+
6774                                ;LOCAL TEXT MESSAGES FOR TEST
6775                                ;-
6776
6777
6778
6779
6780 076460      045     116     000  T39NFL: .ASCIZ  '%N'
6781 076463      123     164     141  T39NE:  .ASCIZ  'Stand-alone Configuration Typeout Not Executed'
6782 076542      045     116     045  T39ETS: .ASCIZ  '%N%A Extended Tape Status Availiable, Drive Number %D2'
6783 076631      045     116     045  T39ETN: .ASCIZ  '%N%A Extended Tape Status NOT Availiable, Drive Number %D2'
6784 076724      045     116     045  T39OF2: .ASCIZ  '%N%A Drive Number %D2%A Is Off-Line'
6785 076770      045     116     045  T39ON2: .ASCIZ  '%N%A Drive Number %D2%A Is On-Line'
6786 077033      045     116     045  T39WRT: .ASCIZ  '%N%A Drive Number %D2%A Is Write Protected'
6787 077106      045     116     045  T39WPN: .ASCIZ  '%N%A Drive Number %D2%A Is NOT Write Protected'
6788 077165      127     122     111  T39NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
6789 077241      103     157     156  T39SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
6790
6791 077331      045     116     045  T39SFS: .ASCIZ  '%N%A State Of Extended Features Switch  ='
6792 077403      045     116     045  T39SBS: .ASCIZ  '%N%A State Of Buffering Switch         ='
6793 077455      045     101     040  T39OFF: .ASCIZ  '%A OFF'
6794 077464      045     101     040  T39ON:  .ASCIZ  '%A ON '
6795 077473      045     116     045  T39MCL: .ASCIZ  '%N%A M7196 Microcode Revision Level     =%02'
```

```
6796                                            .EVEN
6797 077550  000000                    T39RL:  .WORD   0
6798                                            .EVEN
6799                                            .EVEN
6800
6801                                    ;*
6802                                    ;LOCAL STORAGE FOR THIS TEST
6803                                    ;-
6804
6805 077552  000000                    T39DAT: .WORD   0               ;LOGICAL RESPONSE TO QUESTION
6806 077554                            T39REST:
6807 077554                                    SAVREG                  ;SAVE THE REGISTERS
6808 077560  012701  075710                     MOV     #T39PACKET,R1   ;START OF THE PACKET
6809 077564  012721  140006                     MOV     #140006,(R1)+   ;WRITE SUBSYSTEM MEM. WITH ACK,CYC=1
6810 077570  012721  075720                     MOV     #T39TAD,(R1)+   ;ADDRESS OF DATA BLOCK
6811 077574  005021                             CLR     (R1)+           ;EXTENDED ADDRESS
6812 077576  012721  000006                     MOV     #6.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
6813 077602  005021                             CLR     (R1)+           ;CLEAR BSEL0 AND BSEL1
6814 077604  005021                             CLR     (R1)+           ;CLEAR SEL2
6815 077606  005011                             CLR     (R1)            ;CLEAR DATA AREA
6816 077610  000207                             RTS     PC              ;RETURN
6817
6818                                    ;*
6819                                    ;LOCAL TEXT MESSAGES FOR TEST
6820                                    ;-
6821
6822 077612      103     157     156    TST39ID:        .ASCIZ  'Configuration Typeout'
6823                                            .EVEN
6824 077640                                     ENDTST
     077640                                                         L10076:
     077640  104401                                                     TRAP    C$ETST
6825
6826                                    .SBTTL  TEST 12: SCOPE LOOPS
6827
6828
6829                            ;*
6830
6831                            ;
6832                            ;
6833                            ;       THIS IS A STANDALONE ROUTINE PROVIDING A NUMBER OF TIGHT "SCOPE
6834                            ;       LOOPS" USEFUL FOR DEBUGGING BASIC REGISTER ACCESS PROBLEMS WITH
6835                            ;       THE M7196 MODULE.  THESE SCOPE LOOPS CAN BE USED WHEN THE NORMAL
6836                            ;       "LOOP ON ERROR" OR "LOOP ON TEST (SUBTEST)" FACILITIES DON'T
6837                            ;       SEEM TO ALLOW THE OPERATOR TO ZERO IN A PROBLEM IN THE EARLY
6838                            ;       TESTS (I.E. THE HARDWARE MAY NOT BE RESPONDING TO A REGISTER
6839                            ;       ACCESS, CAUSING A BUS ERROR TRAP, EVEN THOUGH THE DEVICE ADDRESS
6840                            ;       SELECTED BY THE PROGRAM MATCHES THE CONFIGURATION SET UP IN THE
6841                            ;       HARDWARE DIP SWITCHES).  THE FOLLOWING MENU OF SCOPE LOOPS ARE
6842                            ;       AVAILABLE:
6843                            ;
6844                            ;               CODE    SCOPE LOOP
6845                            ;
6846                            ;                 0     HELP. PRINT THIS MENU.
6847                            ;                 1     TSBA READ ACCESS
6848                            ;                 2     TSSR READ ACCESS
6849                            ;                 3     INITIALIZE (TSSR WRITE ACCESS)
6850                            ;                 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.
                                        ;
                                        ;


077642                                          BGNTST
077642                                                                          T12::
077642                                          RFLAGS  R0                      ;GET OPERATOR FLAGS
077642  104421                                                                          TRAP    C$RFLA
077644  001403                                  BEQ     1$                      ;BR, IF OK TO RUN
077646  012700  101235                          MOV     #T4ONE,R0               ;"TEST NOT EXECUTED"
077652  000402                                  BR      100$                    ;JUST EXIT IF NOT
077654  012700  101302                  1$:     MOV     #TST40ID,R0             ;TEST ID MESSAGE
077660  004737  016510                  100$:   JSR     PC,TSTSETUP             ;DO THE COMMON SETUP
077664  004737  020500                          JSR     PC,CHKMAN               ;SEE IF MANUAL INTERVENTION ALLOWED
077670  103402                                  BCS     2$                      ;CARRY SET IF INTERVENTION ALLOWED
077672  000137  100356                          JMP     64$                     ;EXIT IF NO MANUAL INTERVENTION
077676  004737  015774                  2$:     JSR     PC,SOFINIT              ;DO A SOFT INIT
077702  103405                                  BCS     5$                      ;BRANCH IF OK
077704  010001                                  MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
077706                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;REPORT FATAL ERROR
077706  104455                                                                          TRAP    C$ERDF
077710  002261                                                                          .WORD   1201
077712  003652                                                                          .WORD   SFIERR
077714  012034                                                                          .WORD   SFIMSG
077716  012700  100374                  5$:     MOV     #SCMENU,R0              ;MENU OF SCOPE LOOP SELECTIONS
077722  012701  000010                          MOV     #8.,R1                  ;MAXIMUM ALLOWED SELECTION
077726  004737  020256                          JSR     PC,GETSEL               ;GO GET THE OPERATORS SELECTION
077732  005700                                  TST     R0                      ;WAS ZERO SPECIFIED ?
077734  001760                                  BEQ     2$                      ;REPEAT MENU IF YES.
077736  020027  000007                          CMP     R0,#7                   ;EXTENDED TSSR ?
077742  001015                                  BNE     3$                      ;BRANCH IF NOT
077744  005737  002226                          TST     EXTFEA                          ;CHECK FOR EXTENDED FEATURES SET
077750  001012                                  BNE     3$                      ;BR, IF IT IS ON
077752                                          PRINTF  #EXFMSG                 ;WARN OPERATOR EXTENDED FEATURES CLEAR
077752  012746  101157                                                                  MOV     #EXFMSG,-(SP)
077756  012746  000001                                                                  MOV     #1,-(SP)
077762  010600                                                                          MOV     SP,R0
077764  104417                                                                          TRAP    C$PNTF
077766  062706  000004                                                                  ADD     #4,SP
077772  000137  077676                          JMP     2$                      ;GO BACK TO BASIC MENU
077776  010004                          3$:     MOV     R0,R4                   ;SAVE THE MENU SELECTION
100000                                          SETPRI  #PRI07                  ;RAISE THE PRIORITY
100000  012700  000340                                                                  MOV     #PRI07,R0
100004  104441                                                                          TRAP    C$SPRI
100006  005037  100366                          CLR     TTION                   ;ASSUME INTERRUPTS ARE ENABLED
100012  032737  000100  177560                  BIT     #100,@#TTICSR           ;ARE TTI INTERRUPTS ON ?
100020  001005                                  BNE     4$                      ;BRANCH IF YES
```

```
6902 100022  005237  100366            INC     TTION                   ;FLAG SET IF INTERRUPTS OFF
6903 100026  052737  000100  177560    BIS     #100,@#TTICSR           ;ENABLE INTERRUPTS
6904 100034  012701  000060        4$: MOV     #TTIVEC,R1              ;START OF TTI VECTORS
6905 100040  011137  100370            MOV     (R1),TVECSAV            ;SAVE THE CURRENT TTI VECTOR
6906 100044  012721  100270            MOV     #60$,(R1)+              ;SET NEW INTERRUPT ROUTINE
6907 100050  011137  100372            MOV     (R1),TPRISAV            ;SAVE THE VECTOR PRIORITY
6908 100054  012711  000340            MOV     #PRI07,(R1)             ;USE PRIORITY SEVEN
6909 100060                            SETPRI  #PRI00                  ;LOWER INTERRUPT BR LEVEL
     100060  012700  000000                                                    MOV     #PRI00,R0
     100064  104441                                                            TRAP    C$SPRI
6910 100066  006304                    ASL     R4                      ;CONVERT TO WORD OFFSET
6911 100070  000174  100074            JMP     @6$(R4)                 ;JUMP TO PROPER LOOP
6912 100074  077676                6$: .WORD   2$                      ;RETYPE THE MENU
6913 100076  100116                    .WORD   10$                     ;TSBA READ ACCESS
6914 100100  100126                    .WORD   15$                     ;TSSR READ ACCESS
6915 100102  100140                    .WORD   20$                     ;TSSR WRITE ACCESS
6916 100104  100160                    .WORD   25$                     ;TSDB HIGH BYTE WRITE ACCESS
6917 100106  100204                    .WORD   30$                     ;TSDB LOW BYTE WRITE ACCESS
6918 100110  100230                    .WORD   35$                     ;TSDB MAINTENANCE MODE
6919 100112  100250                    .WORD   40$                     ;TSDBX WRITE ACCESS
6920 100114  100362                    .WORD   65$                     ;LEAVE THE TEST
6921
6922
6923 100116  105065  000000       10$: CLRB    TSDB(R5)                ;ENTER MAINTENANCE MODE
6924 100122  011500            12$: MOV     (R5),R0                 ;READ TSBA REGISTER
6925 100124  000776                    BR      12$                     ;LOOP UNTIL HALTED
6926
6927
6928 100126  012703  000002       15$: MOV     #TSSR,R3                ;ADDRESS OF TSSR REGISTER
6929 100132  060503                    ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
6930 100134  011300            18$: MOV     (R3),R0                 ;READ TSSR REGISTER
6931 100136  000776                    BR      18$                     ;LOOP UNTIL STOPPED
6932
6933 100140  004737  020174       20$: JSR     PC,GETPAT               ;READ THE DATA PATTERN
6934 100144  010001                    MOV     R0,R1                   ;DATA PATTERN FOR LOOP
6935 100146  012703  000002            MOV     #TSSR,R3                ;ADDRESS OF TSSR
6936 100152  060503                    ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
6937 100154  010113            22$: MOV     R1,(R3)                 ;WRITE DATA TO TSSR
6938 100156  000776                    BR      22$                     ;LOOP
6939
6940
6941 100160  105065  000000       25$: CLRB    TSDB(R5)                ;ENTER MAINTENANCE MODE
6942 100164  004737  020174            JSR     PC,GETPAT               ;READ THE DATA PATTERN
6943 100170  010001                    MOV     R0,R1                   ;DATA PATTERN FOR LOOP
6944 100172  012703  000001            MOV     #TSDBH,R3               ;ADDRESS OF HIGH BYTE OF TSDB
6945 100176  060503                    ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
6946 100200  110113            27$: MOVB    R1,(R3)                 ;WRITE THE DATA TO TSDB, HIGH BYTE
6947 100202  000776                    BR      27$                     ;LOOP UNTIL STOPPED
6948
6949
6950 100204  105065  000000       30$: CLRB    TSDB(R5)                ;ENTER MAINTENANCE MODE
6951 100210  004737  020174            JSR     PC,GETPAT               ;READ THE DATA PATTERN
6952 100214  010001                    MOV     R0,R1                   ;DATA PATTERN FOR LOOP
6953 100216  012703  000000            MOV     #TSDB,R3                ;ADDRESS OF TSSR
6954 100222  060503                    ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
6955 100224  110113            32$: MOVB    R1,(R3)                 ;WRITE DATA TO TSSR, LOW BYTE
6956 100226  000776                    BR      32$                     ;LOOP UNTIL HALTED BY OPERATOR
```

```
6957
6958 100230  004737  020174          35$:    JSR     PC,GETPAT               ;READ THE DATA PATTERN
6959 100234  010001                          MOV     R0,R1                   ;DATA PATTERN FOR LOOP
6960 100236  012703  000000                  MOV     #TSDB,R3                ;SELECT TSDB
6961 100242  060503                          ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
6962 100244  010113                  37$:    MOV     R1,(R3)                 ;WRITE THE DATA PATTERN
6963
6964 100246  000776                          BR      37$                     ;LOOP UNTIL HALTED
6965
6966 100250  004737  020174          40$:    JSR     PC,GETPAT               ;READ THE DATA PATTERN
6967 100254  010001                          MOV     R0,R1                   ;SAVE THE DATA PATTERN
6968 100256  012703  000003                  MOV     #TSSRH,R3               ;BYTE ADDRESS OF TSSR, HIGH BYTE
6969 100262  060503                          ADD     R5,R3                   ;POINT TO TSV05'S REGISTERS
6970 100264  110113                  42$:    MOVB    R1,(R3)                 ;WRITE THE DATA TO REGISTER
6971 100266  000776                          BR      42$                     ;LOOP UNTIL HALTED
6972
6973
6974                                         ;+
6975                                         ;PROCESS CONSOLE INTERRUPTS
6976                                         ;
6977
6978 100270  010046                  60$:    MOV     R0,-(SP)                ;SAVE WORK REGISTER
6979 100272  113700  177562                  MOVB    @#TTIBFR,R0             ;GET THE OPERATOR INPUT
6980 100276  042700  000200                  BIC     #200,R0                 ;STRIP OFF PARITY BIT
6981 100302  122700  000015                  CMPB    #15,R0                  ;IS IT A CARRIAGE RETURN ?
6982 100306  001021                          BNE     61$                     ;JUST EXIT IF NOT
6983 100310  012766  077676  000002          MOV     #2$,2(SP)               ;RETURN TO MASTER MENU
6984 100316  005066  000004                  CLR     4(SP)                   ;FORCE PRIORITY ZERO
6985 100322  013737  100370  000060          MOV     TVECSAV,@#TTIVEC        ;RESTORE SUPERVISOR VECTOR
6986 100330  013737  100372  000062          MOV     TPRISAV,@#TTIVEC+2      ;RESTORE SUPERVISOR PRIORITY
6987 100336  005737  100366                  TST     TTION                   ;ARE SUPERVISOR INTERRUPTS ENABLED ?
6988 100342  001403                          BEQ     61$                     ;BRANCH IF YES
6989 100344  042737  000100  177560          BIC     #100,@#TTICSR           ;TURN OFF TTI INTERRUPTS
6990 100352  012600                  61$:    MOV     (SP)+,R0                ;RESTORE REGISTER
6991 100354  000002                          RTI                             ;RETURN FROM INTERUPT
6992
6993 100356                          64$:
6994 100356                          63$:    EXIT    TST                     ;EXIT THE TEST
     100356  104432                                                                  TRAP    C$EXIT
     100360  000736                                                                  .WORD   L10077-.
6995 100362  000137  000200          65$:    JMP     200                     ;RETURN TO SUPERVISOR
6996
6997                                         ;+
6998                                         ;LOCAL STORAGE FOR THIS TEST
6999                                         ;-
7000
7001 100366  000000                  TTION:          .WORD   0               ;WORD SET IF SUPERVISOR TTI INTER OFF
7002 100370  000000                  TVECSAV:        .WORD   0               ;SAVE TTI VECTOR
7003 100372  000000                  TPRISAV:        .WORD   0               ;SAVE TTI PRIORITY
7004
7005
7006                                         ;+
7007                                         ;MENU FOR OPERATOR INPUT FOR SCOPE LOOPS
7008                                         ;-
7009
7010                                    .EVEN
7011 100374  100426  100501  100527  SCMENU: .WORD   1$,2$,3$,4$,5$,6$
```

```
TSV5 - HARDWARE TESTS   MACRO M1113  14-JUN-84 15:55                                                    SEQ 0251
TEST 12: SCOPE LOOPS

   7012 100410  100700  100736  101004          .WORD   7$,8$,9$,10$,11$,12$,0
   7013
   7014
   7015 100426     012     123     105  1$:     .ASCIZ  <12>'SELECT SCOPE LOOP FROM FOLLOWING OPTIONS:
   7016 100501     012     011     060  2$:     .ASCIZ  <12>'    0       Display This Menu'
   7017 100527     011     061     011  3$:     .ASCIZ  '        1       TSBA Read Access'
   7018 100553     011     062     011  4$:     .ASCIZ  '        2       TSSR Read Access'
   7019 100577     011     063     011  5$:     .ASCIZ  '        3       Initialize (TSSR Write Access)'
   7020 100641     011     064     011  6$:     .ASCIZ  '        4       TSDB High Byte Write Access'
   7021 100700     011     065     011  7$:     .ASCIZ  '        5       TSDB Low Byte Write Access'
   7022 100736     011     066     011  8$:     .ASCIZ  '        6       TSDB Maintenance Mode Write Access
   7023 101004     011     067     011  9$:     .ASCIZ  '        7       TSDBX (TSSR High Byte) Write Access'
   7024 101053     011     070     011  10$:    .ASCIZ  '        8       Return to Diagnostic Supervisor
   7025 101116     000             011  11$:    .ASCIZ  ''
   7026 101117     124     171     160  12$:    .ASCIZ  'Type RETURN To Stop Scope Loops'
   7027 101157     045     116     045  EXFMSG: .ASCIZ  '%NMA *** Extended Features Switch Not On ***
   7028 101235     123     164     141  T40NE:  .ASCIZ  'Stand-alone Scope Loops Not Executed'
   7029 101302     123     143     157  TST40ID:        .ASCIZ  'Scope Loops'
   7030                                          .EVEN
   7031 101316                                  ENDTST
        101316                                                                  L10077:
        101316  104401                                                                  TRAP    C$ETST
   7032 101320                                  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
                                        .WORD L10100-L$HARD/2
                                L$HARD::
                                        GPRMA   HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
                                        .WORD   T$CODE
                                        .WORD   HPM1
                                        .WORD   T$LOLIM
                                        .WORD   T$HILIM
                                        GPRMA   HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
                                        .WORD   T$CODE
                                        .WORD   HPM2
                                        .WORD   T$LOLIM
                                        .WORD   T$HILIM
                                        ;GPRMD  HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
                                        ENDHRD
                                        .EVEN
                                L10100:
                                HPM1:   .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
                                HPM2:   .ASCIZ  'INTERRUPT VECTOR   '
                                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.
                                ;--
101452                                  BGNSFT
101452  000003                          .WORD L10101-L$SOFT/2
101454                          L$SOFT::
                                ;       GPRML   SPM1,0,-1,YES                   ; GET TRANSPORT TEST FLAG.
101454                                  GPRML   SPM4,2,-1,YES                   ; GET ITERATION CONTROL.
101454  001130                          .WORD   T$CODE
101456  101512                          .WORD   SPM4
101460  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
101462                                  ENDSFT
                                        .EVEN
101462                          L10101:

101462  105     116     101     SPM1:   .ASCIZ  'ENABLE TRANSPORT TESTS '
101512  111     116     110     SPM4:   .ASCIZ  'INHIBIT ITERATIONS     '
101542  120     105     122     SPM6:   .ASCIZ  'PER TEST ERROR LIMIT   '
101572  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).
                                ;

101622                          PATCH::

101622                                  .BLKW   32.

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

TSV6 - PARAMETER CODING MACRO M1113 14-JUN-84 15:55 SEQ 0254

SYMBOL TABLE

```
ADDSSR  012126 G        C$AU  = 000052        DEVDRO  023332        FREEHI  003130        INTCPC  016150
ADR   = 000020 G        C$AUTO= 000061        DEVNRD  023251        FRESIZ  003126 G      INTFLA  016145
AMBTSS  006635          C$BRK = 000022        DEVNXR  023167        FUSI    004117        INTMAS  016144
ASSEMB= 000010          C$BSEG= 000004        DEVONL  023117        F$AU  = 000015        INTR    016216 G
A1716 = 000003          C$BSUB= 000002        DEVSUM  023062        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$DU  = 000016        INVERT  021142 G
BENBSW  002230 G        C$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$EDIT= 000003        EF.RES= 000037 G      F$PRGI= 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$MEM = 000031        EXFMSG  101157        G$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  021214 G        C$SEFG= 000046        FERCM   011722        HPM1    101342        KTENAB  003134 G
CMPMEM  017660          C$SPRI= 000041        FIFEXP  012170 G      HPM2    101376        KTFLG   003132 G
CONFIG  017250          C$SVEC= 000037        FIF1MS  012242        HPM3    101422        KTINIT  021010
COUNT   002310 G        C$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
C1ABM   003164 G        DEVCNT  002220 G      FREE    003124 G      INCERK  017044        LISTAL= 000001
```

TSV6 - PARAMETER CODING MACRO M1113 14-JUN-84 15:55 SEQ 0255

SYMBOL TABLE

```
LOE    = 040000 G      L$UNIT  002012 G      L10071  056470      OFL    = 000100      PRMNO   002320 G
LOOPCN  002216 G       L10000  002164         L10072  057734      ONEFIL= 000000       PRMSGE  014552 G
LOOPCO  013126         L10001  002176         L10073  066624      O$APTS= 000000       PRMSGO  014732
LOOPFL  003162 G       L10002  005766         L10074  064764      O$AU   = 000001      PRMSG1  014777
LOT    = 000010 G      L10003  012044         L10075  074466      O$BGNR= 000001       PRMSG2  015035
L$ACP   002110 G       L10004  012062         L10076  077640      O$BGNS= 000001       PROASC  014420
L$APT   002036 G       L10005  012100         L10077  101316      O$DU   = 000001      PR1ASC  014465
L$AU    022306 G       L10006  012106         L10100  101342      O$ERRT= 000000       PST32W  003152 G
L$AUT   002070 G       L10007  012124         L10101  101462      O$GNSW= 000001       PUNIT   022240
L$AUTO  022512 G       L10010  012142         MEMADD  013754 G     O$POIN= 000001       PW.D11= 000021
L$CCP   002106 G       L10011  012166         MEMCK   021232 G     O$SETU= 000000       PW.D13= 000022
L$CLEA  022572 G       L10012  012240         MENASC  020447       PASRPT  022010       PW.D22= 000020
L$CO    002032 G       L10013  012410         MENERR  020374       PATCH   101622 G     PW.NOP= 000000
L$DEPO  002011 G       L10014  013124         MENRES  020476       PATDAT  020230       PW.NO1= 000023
L$DESC  003410 G       L10015  013752         MIMENU  013352       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$DU    022404 G       L10024  015576         MSGLOO  013064 G     PKLOW  = 000002      PW.WMI= 000010
L$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  022236         MS.ATT= 000006       PKTGET  012064 G     P.ACK  = 100000
L$ENVI  002044 G       L10031  022402         MS.EXT= 000200       PKTMES  012110 G     P.CMD  = 000037
L$ETP   002102 G       L10032  022510         MS.RSD= 000001       PKTRAM  004745 G     P.CONT= 000012
L$EXP1  002046 G       L10033  022570         MS.RSF= 000020       PKTSSR  012046 G     P.CVC  = 040000
L$EXP4  002064 G       L10034  022616         MS.RST= 000010       PNT    = 001000 G     P.FMT  = 000140
L$EXP5  002066 G       L10035  023060         M8186   005554       PRAMPK  013776       P.FORM= 000011
L$HARD  101322 G       L10036  024364         M8189   005645       PRASC   014523       P.GETS= 000017
L$HIME  002120 G       L10037  026356         NBA    = 002000      PRBEXP  015470       P.IE   = 000200
L$HPCP  002016 G       L10040  024640         NEWPAS  021744       PRBMSG  015336       P.INIT= 000013
L$HPTP  002022 G       L10041  025104         NODEV   003114 G     PRBREC  015472       P.MODE= 007400
L$HW    002156 G       L10042  031752         NOEXTF  030146       PRBTOT  015423       P.OPP  = 020000
L$ICP   002104 G       L10043  026732         NOINIT  004335       PRBYTE  015122 G     P.POSI= 000010
L$INIT  021512 G       L10044  027222         NOINTR  004221       PRI    = 002000 G     P.READ= 000001
L$LADP  002026 G       L10045  027520         NOITS   002170 G     PRIADD  010160       P.SWB  = 010000
L$LAST  102004 G       L10046  030152         NOMAN   020534       PRIAO   010230       P.WRIT= 000005
L$LOAD  002100 G       L10047  034542         NOMEM   005460       PRIBXO  007612 G     P.WRTC= 000004
L$LUN   002074 G       L10050  032306         NP.IR  = 000200      PRIEQU  010060       P.WRTS= 000006
L$MREV  002050 G       L10051  032700         NP.LOO= 000040       PRIPKT  007370 G     QVP     002204 G
L$NAME  002000 G       L10052  033306         NP.OUT= 000100       PRIRAM  010066       RAMASC  014156
L$PRIO  002042 G       L10053  040334         NP.WRP= 000020       PRITAD  010274       RAMDAT  002242 G
L$PROT  021502 G       L10054  035626         NSI     004152       PRITSS  006024       RAMERR  015510 G
L$PRT   002112 G       L10055  036560         NSINIT  004407       PRITO   010356       RAMEXP  015530 G
L$REPP  002062 G       L10056  050446         NUL     004527       PRIT1   010421       RAMFOR  010116
L$REV   002010 G       L10057  040640         NULCR   004530       PRIXOR  007742 G     RAMSIZ  002302 G
L$RPT   022620 G       L10060  042050         NXM    = 004000      PRIO0  = 000000 G     RAMTAD  015516 G
L$SOFT  101454 G       L10061  043410         NXMFLG  003136 G     PRIO1  = 000040 G     RCVHIA  002304 G
L$SPC   002056 G       L10062  043766         NXMHI   003142 G     PRIO2  = 000100 G     RCVLOA  002306 G
L$SPCP  002020 G       L10063  045240         NXMLO   003140 G     PRIO3  = 000140 G     RDERR   005206
L$SPTP  002024 G       L10064  046304         NXMTST  021406       PRIO4  = 000200 G     RECMSG  002466 G
L$STA   002030 G       L10065  051726         NXR     003740       PRIO5  = 000240 G     RECV    002234 G
L$SW    002166 G       L10066  062554         NXRERR  005736 G     PRIO6  = 000300 G     REGSAV  020140
L$TEST  002114 G       L10067  053732         NXRX    003777       PRIO7  = 000340 G     RETERR  005372
L$TIML  002014 G       L10070  055220         NXTU    021756       PRMESS  014242       REWIND  011014 G
```

TSV6 - PARAMETER CODING MACRO M1113 14-JUN-84 15:55 SEQ 0256

SYMBOL TABLE

```
RMCHBE= 000167          S1.IID= 004000          TST40I  101302          T10     066626 G        T15BFR  033352
RMCHEN= 000200          S1.I1R= 020000          TSV2    002000 G        T11     074470 G        T15BF2  034040
RMMSGB= 000215          S1.I2R= 040000          TSV3    002176 G        T12     077642 G        T15BS0  034040
RMMSGE= 000234          S1.PAR= 100000          TSV4    021502 G        T12BFR  030242          T15BS1  034041
RMPKTB= 000201          S2.ATI= 000010          TSV5    023402 G        T12BKG  030717          T15DAT  033340
RMPKTE= 000210          S2.BTI= 000004          TSV6    101320 G        T12BLK  030274          T15L00  032004
RMR   = 010000          S2.DIM= 000200          TTIBFR= 177562 G        T12CHA  031710          T15PAC  033330
RWPACK  011110          S2.ILW= 000100          TTICSR= 177560 G        T12CKR  031470 G        T15PK2  034030
SC    = 100000          S2.INR= 000020          TTION   100366          T12CON  031276          T15RES  034432
SCE   = 020000          S2.OUT= 000040          TTION2  071414          T12DAT  030230          T15RT2  034504
SCHERR  005300          S2.UND= 000003          TTIVEC= 000060 G        T12DPR  031076          T15SSR  034046
SCME    005013          TBLEND= 003062 G        TVECSA  100370          T12GET  030456          T15S2   034042
SCMENU  100374          TCOASC  006476          TVSAV2  071416          T12HIA  030262          T15S3   034044
SDELAY  010660          TCOCOD  006676          T$ARGC= 000001          T12KT   030270          T16BEN  040220
SELASC  020442          TEMP1   003116 G        T$CODE= 001130          T12LOA  030264          T16BFR  040172
SELDAT= 000004          TEMP2   003120 G        T$ERRN= 002261          T12L00  026406          T16BFS  040212
SEL2  = 000002          TERCLS= 000016          T$EXCP= 000000          T12MSG  030621          T16CLR  040036
SETMAP  017316          TESTNO= 000014          T$FLAG= 000040          T12NIN  031005          T16DAT  040160
SETU    022042          TEXASC  006435          T$GMAN= 000000          T12NXM  031167          T16DT2  040230
SFFMSG  012102 G        TFCASC  006537          T$HILI= 000776          T12PAC  030220          T16D01  036612
SFHERR  003705          TIMEXP  015552 G        T$LAST= 000001          T12PAR  030266          T16D28  036614
SFIERR  003652          TIMSGO  015600          T$LOLI= 000000          T12SET  031646          T16D53  036616
SFIMSG  012034 G        TINERR  012021          T$LSYM= 010000          T12SWR  031600          T16D78  036620
SFPTBL  002166 G        TMPBFR  002632 G        T$LTNO= 000014          T12TBE  030426          T16LEN  037142
SIFLAG  003154 G        TNAM    016704          T$NEST= 177777          T12WRT  030532          T16L00  034562
SIMSG   011766          TPRISA  100372          T$NS0 = 000000          T121L0  026520          T16PAC  040150
SKIPT   003400          TPSAV2  071420          T$NS1 = 000005          T122L0  027054          T16PK2  040220
SOFINI  015774 G        TRANST  002166 G        T$NS2 = 000002          T123L0  027274          T16REJ  037254
SPACE   010466 G        TSBA  = 000000 G        T$NS3 = 000003          T124L0  027716          T16RES  037770
SPM1    101462          TSBAH = 000001 G        T$PTNU= 000000          T124TS  030272          T16SEX  040130
SPM4    101512          TSDB  = 000000 G        T$SAVL= 177777          T13BFR  024022          T16SRD  040062
SPM6    101542          TSDBH = 000001 G        T$SEGL= 177777          T13DAT  024010          T16SSR  036672
SPM7    101572          TSFCOD  007236          T$SEKO= 010000          T13L00  023420          T16TSB  037040
SR0   = 177572          TSREJ = 000006          T$SUBN= 000000          T13MEM  024115          T16T01  037371
SR1   = 177574          TSSDEF  006606          T$TAGL= 177777          T13NBA  024154          T16T28  037470
SR2   = 177576          TSSR  = 000002 G        T$TAGN= 010102          T13PAC  024000          T16T53  037570
SR3   = 172516          TSSRBI  003502 G        T$TEMP= 000000          T13RES  024316          T16T78  037670
SSR   = 000200          TSSRFO  006415          T$TEST= 000014          T13SSR  024227          T16WMI  040102
STATCO  012412          TSSRH = 000003 G        T$TSTM= 177777          T14BFR  025136          T162SS  036727
SVCGBL= 000000          TSSX    004020          T$TSTS= 000001          T14BS0  025130          T163SS  036773
SVCINS= 000000          TSTBLK  002752 G        T$$AU = 010031          T14BS1  025131          T17BEN  050322
SVCSUB= 000001          TSTCNT  002214 G        T$$AUT= 010033          T14BS2  025132          T17BFR  050202
SVCTAG= 000000          TSTEND  01672C          T$$CLE= 010034          T14DAT  025130          T17BFS  050222
SVCTST= 000001          TSTFLA  002314 G        T$$DU = 010032          T14DTA  025550          T17CLE  050106
S$LSYM= 010000          TSTL00  016456 G        T$$HAR= 010:00          T14L00  024404          T17CLR  047720
S0.IDB= 000010          TSTPTR  002316 G        T$$HW = 010000          T14NBA  025562          T17DAT  050170
S0.IFB= 000002          TSTSET  016510 G        T$$INI= 010030          T14NIN  026023          T17DT2  050340
S0.IFP= 000001          TST12I  030430          T$$MSG= 010025          T14PAC  025120          T17EXE  046464
S0.ILD= 000020          TST13I  024042          T$$PRO= 010027          T14PK2  025540          T17EXP  046342
S0.ION= 000040          TST14I  026201          T$$RPT= 010035          T14RES  026246          T17EXS  046362
S0.IRD= 000100          TST15I  034413          T$$SEG= 010000          T14RST  026304          T17L00  040364
S0.IRW= 000004          TST16I  036622          T$$SOF= 010101          T14SSR  025733          T17MSK  046336
S0.ISP= 000200          TST17I  046566          T$$SRV= 010026          T14TSB  026115          T17PAC  050160
S1.ICE= 002000          TST18I  051156          T$$SUB= 010074          T142RE  025636          T17PK2  050330
S1.IEO= 010000          TST19I  060142          T$$SW = 010001          T15AM2  034122          T17RFI  050066
S1.IFM= 001000          TST20I  065142          T$$TES= 010077          T15AM3  034223          T17RSF  047764
S1.IHE= 000400          TST39I  077612          T1      023402 G        T15AM4  034325          T17SET  050126
```

TSV6 - PARAMETER CODING MACRO M1113 14-JUN-84 15:55 SEQ 0257

SYMBOL TABLE

```
T17SNP  050006          T19SSR  060203          T3      026360 G        T39NBA  077165          WF.IED= 000010
T17SRD  047744          T19WCT  062102          T38FLG  003150 G        T39NE   076463          WF.IER= 000004
T17SSR  046605          T19WFI  062046          T3.1    026406          T39NFL  076460          WF.IHI= 000200
T17WFD  046464          T19WFM  062122          T3.2    026746          T39OFF  077455          WF.IRE= 000040
T17WFI  050032          T19WST  061503          T3.3    027236          T39OF2  076724          WF.IWF= 000020
T171CM  047125          T191CM  060640          T3.4    027534          T39ON   077464          WF.IWR= 000100
T172CM  047207          T192CM  060722          T38ASC  074421          T39ON2  076770          WF.I3R= 000002
T172SS  046642          T192SS  060240          T38ASN  074440          T39PAC  075710          WF.I4R= 000001
T173CM  047303          T193CM  061016          T38ASO  074316          T39PK2  076410          WRTCHR  010662 G
T173SS  046706          T193SS  060304          T38AS1  074363          T39PK3  076440          WRTERR  005113
T174CM  047367          T194SS  060351          T38BFR  071446          T39PK4  076450          WRTMSG  005056
T174SS  046753          T195CM  061104          T38BSO  071440          T39RES  077554          WSMBK   021224 G
T175CM  047452          T195SS  060414          T38BS1  071441          T39RL   077550          XFERAS  015740
T175SS  047016          T196CM  061160          T38BS2  071442          T39SBS  077403          XNXM    016376
T176CM  047526          T196SS  060460          T38CNT  074464          T39SFS  077331          XORBFO  007674
T176SS  047062          T197CM  061243          T38DAT  073754          T39SIZ  076456          XORFOR  010012
T177CM  047634          T197SS  060523          T38DLY  071422          T39SSR  077241          XSTO  = 000006 G
T188FR  051662          T198CM  061331          T38DSW  072150          T39TAD  075720          XST1  = 000010 G
T18CMP  051363          T198SS  060572          T38DTA  072140          T39WPN  077106          XST2  = 000012 G
T18DAT  051650          T199CM  061420          T38EAI  072146          T39WR   076452          XST3  = 000014 G
T18DT2  051720          T2      024366 G        T38EB   072122          T39WRT  077033          XST4  = 000016 G
T18EXP  051126          T2.1    024404          T38ID   073325          T4      031754 G        XSOBOT= 000002
T18LOO  050466          T2.2    024654          T38INT  072716          T4.1    032004          XSOEOT= 000001
T18MSK  051122          T20BEN  066502          T38MBP  074014          T4.2    032310          XSOIE = 000040
T18PAC  051640          T20BFR  066362          T38MSG  073265          T4.3    032702          XSOILA= 000400
T18PK2  051710          T20BFS  066402          T38MS1  073126          T40NE   101235          XSOILC= 001000
T18SET  051530          T20CLE  066274          T38MS2  073221          T5      034544 G        XSOLET= 020000
T18SMI  051506          T20CLR  066062          T38MS3  072265          T5.1    034562          XSOMOT= 000200
T18SRD  051466          T20CWP  065767          T38NBA  072552          T5.2    035642          XSONEF= 002000
T18SSR  051215          T20DAT  066350          T38NE   072210          T6      040336 G        XSOONL= 000100
T18XS   051152          T20DT2  066520          T38OFL  073002          T6.1    040364          XSOPED= 000010
T182SS  051252          T20EXE  065142          T38ONL  072746          T6.2    040654          XSORLL= 010000
T183SS  051316          T20EXP  065022          T38PAC  071430          T6.3    042064          XSORLS= 040000
T198EN  062432          T20EXS  065042          T38PK2  072130          T6.4    043424          XSOTMK= 100000
T198FC  057756          T20LOO  062574          T38PK3  072160          T6.5    044002          XSOVCK= 000020
T198FR  062312          T20MSK  065016          T38PK4  072200          T6.6    045254          XSOWLE= 004000
T198FS  062332          T20PAC  066340          T38RES  073756          T7      050450 G        XSOWLK= 000004
T19CLE  062160          T20PK2  066510          T38SIZ  072206          T8      051730 G        XXCOMM  003122 G
T19CLR  061714          T20RFI  066146          T38SSR  072626          T8.1    051746          X$ALWA= 000000
T19CNV  062224          T20RSF  065666          T38SST  073036          T8.2    053746          X$FALS= 000040
T19DAT  062300          T20SET  066314          T38TAD  071440          T8.3    055234          X$OFFS= 000400
T19DT2  062450          T20SEX  066256          T38WLE  072505          T8.4    056504          X$TRUE= 000020
T19EXE  060142          T20SRD  066106          T38WR   072202          T9      062556 G        X1.COR= 020000
T19EXP  060022          T20SSR  065171          T38WRL  072444          T9.1    062574          X1.DLT= 100000
T19EXS  060042          T20SWP  065557          T38WRT  072360          UAM   = 000200 G        X1.MBZ= 017375
T19LOO  051746          T20WFI  066166          T39BFR  075726          UNITN   002202 G        X1.RBP= 000400
T19MSK  060016          T20WFM  066222          T39BSO  075720          UNREC = 000006          X1.SPA= 040000
T19PAC  062270          T20WMI  066242          T39BS1  075721          USI     004123          X1.UNC= 000002
T19PK2  062440          T20WNP  066126          T39BS2  075722          WAITF   016250 G        X2.BUF= 000100
T19PRE  057754          T202SS  065226          T39DAT  077552          WC.IFA= 000200          X2.EXT= 000200
T19RFI  062026          T203SS  065272          T39DLY  075704          WC.IFE= 000002          X2.OPM= 100000
T19RSF  061760          T204SS  065337          T39DSW  076430          WC.IGO= 000001          X2.RCE= 040000
T19RST  061610          T205SS  065402          T39DTA  076420          WC.IRE= 000010          X2.REV= 000077
T19SET  062200          T206SS  065446          T39EAI  076426          WC.IRW= 000004          X2.SPA= 035400
T19SEX  062142          T208SS  065511          T39ETN  076631          WC.IOT= 000100          X2.UNI= 000007
T19SNP  062002          T23A    003144 G        T39ETS  076542          WC.I1T= 000040          X2.WCF= 002000
T19SRD  061740          T23B    003146 G        T39MCL  077473          WC.ISR= 000020          X3.DCK= 000010
```

```
TSV6 - PARAMETER CODING MACRO M1113  14 JUN-84 15:55                                    SEQ 0258
SYMBOL TABLE

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

. ABS.  102004      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  31032 WORDS  ( 122 PAGES)
DYNAMIC MEMORY:  20614 WORDS  ( 79 PAGES)
ELAPSED TIME:  00:46:12
CVTSBC,CVTSBC.LST/-SP=SVC/ML,TSV1B,TSV22B,TSV3B,TSV4,TSV55B,TSV6
.
```