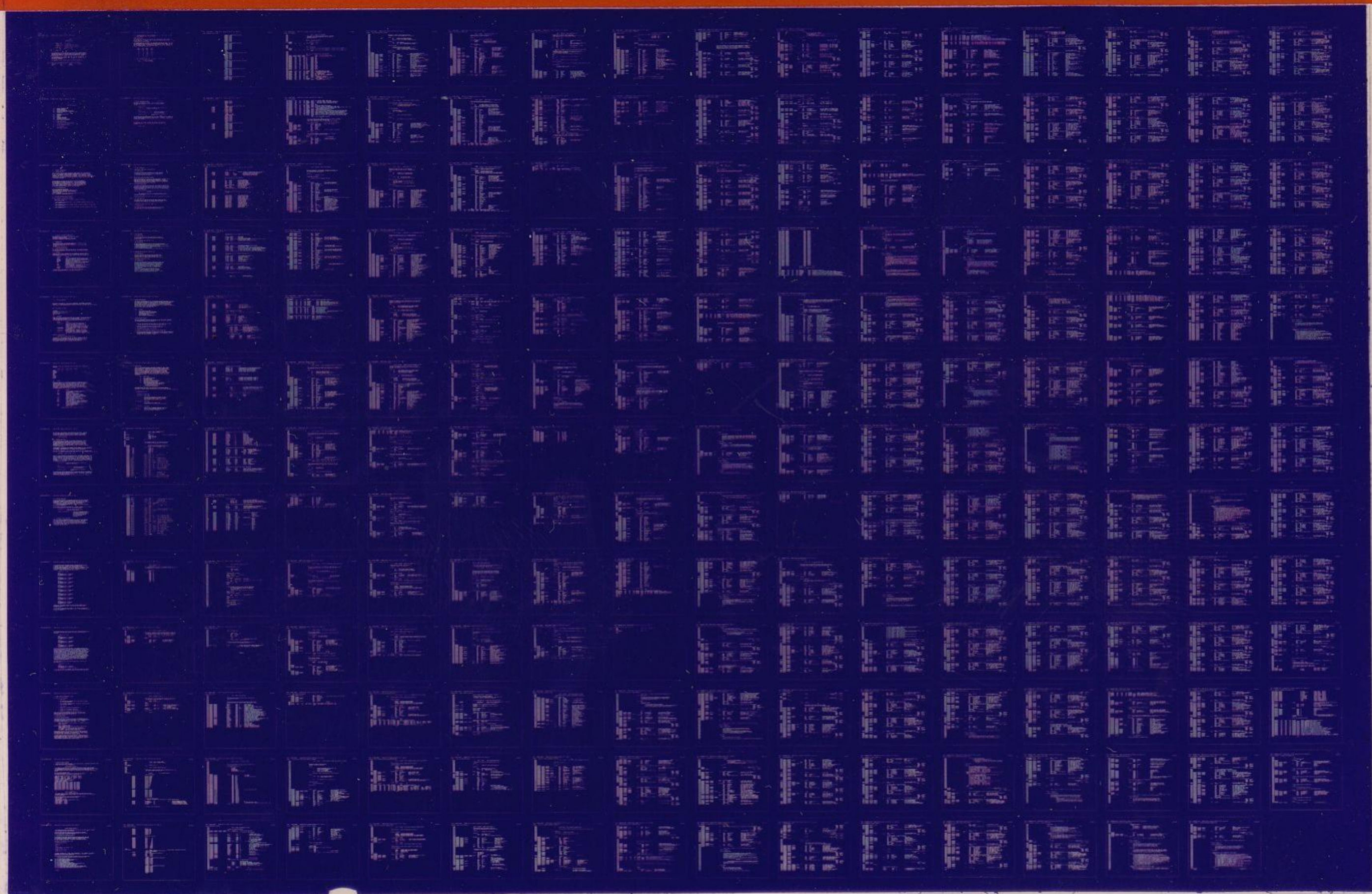




.REM_
IDENTIFICATION

PRODUCT ID: AC-T0950-MC
PRODUCT TITLE: CVTSBDO TSV05 CTRL PART 2
DEPARTMENT: COMPUTER SPECIAL SYSTEMS/PGG
DATE: AUGUST 23, 1985

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1982, 1985 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	COMMANDS
2.2	SWITCHES
2.3	FLAGS
2.4	HARDWARE QUESTIONS
2.5	SOFTWARE QUESTIONS
2.6	EXTENDED P-TABLE DIALOGUE
2.7	QUICK STARTUP PROCEDURE
3.0	ERROR INFORMATION
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES
7.0	MAINTENANCE HISTORY

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

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

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

1.2 SYSTEM REQUIREMENTS

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

1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

1. CHQUS XXDP+ USERS GUIDE; DOCUMENT NUMBER AC-F348E-MC
DATE: 14 JULY 1980.
2. TSV05 TRANSPORT SUBSYSTEM USER'S GUIDE; DOCUMENT NUMBER EK-TSV05-UG-001
DATE: AUGUST 1982
3. TSV05 TRANSPORT SUBSYSTEM TECHNICAL MANUAL; DOCUMENT NUMBER EK-TSV05-TM-001
DATE: AUGUST 1982
4. TSV05 TRANSPORT SUBSYSTEM INSTALLATION MANUAL; DOCUMENT NUMBER EK-TSV05-IN-001
DATE: AUGUST 1982

1.4 DIAGNOSTIC HIERARCHY 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 SUCCESSFULLY.

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 (0) ? 8<CR>

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

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

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

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

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

* UNITS (0) ? 8<CR>

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

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

2.7 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

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

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

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

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

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

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

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

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

3.2 SPECIFIC ERROR MESSAGES

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

ERROR MESSAGE EXAMPLE 1

THIS ERROR IS INDICATIVE OF AN INCORRECT REGISTER OR STATUS WORD RETURNED TO THE DIAGNOSTIC. THE FIRST PART DEFINES THE TEST FUNCTION AND UNIT THAT FAILED. THE SECOND PART PROVIDES 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 SIGNALS 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 MRD ERR 00121 ON UNIT 00 TST 001 SUB 002 PC: 023306
MOT BIT (XST0) NOT SET DURING REWIND (EXTENDED FEATURES MODE)
EXPD: 000312 RECV: 000112 XOR: 000200

4.0 PERFORMANCE AND PROGRESS REPORTS

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

SUCCESSFUL RUN EXAMPLE (PDP-11/23)

DR>STA/FLA:PNT:MOE:UAM

UNITS (0) ? 1

UNIT 0

DEVICE ADDRESS (0) 172520 ? <CR>

VECTOR (0) 224 ? <CR>

CHANGE SW (L) ? N<CR>

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

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

TST: 001 INITIALIZE #3 TEST
TST: 002 BASIC WRITE SUBSYSTEM MEMORY TEST
TST: 003 DMA MEMORY ADDRESSING TEST
TST: 004 RAM EXERCISER TEST
TST: 005 FIFO EXERCISER TEST
TST: 006 STATIC TRANSPORT BUS CHECK
TST: 007 TRANSPORT BUS INTERFACE CHECK VIA LOOPBACK TEST
TST: 008 READ/WRITE DATA PARITY CHECK TEST
TST: 009 MISCELLANEOUS LOGIC CHECKS TEST
TST: 010 STAND-ALONE MANUAL INTERVENTION NOT EXECUTED TEST
TST: 011 STAND-ALONE CONFIGURATION TIMEOUT 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 (0) ? <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 BSEL0 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 REVISION 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	TSD8 HIGH BYTE WRITE ACCESS
5	TSD8 LOW BYTE WRITE ACCESS
6	TSD8 MAINTENANCE-MODE WORD WRITE ACCESS
7	TSD8X (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 PATTERNS ARE 0-377. TYPING <RETURN> CAUSES AN EXIT FROM THE SCOPE LOOP BACK TO MENU LEVEL.

7.0 MAINTENANCE HISTORY

REVISION A - MARCH 1982

REVISION B - APRIL 1983

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

REVISION C - JUNE 1984

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

REVISION D - JULY 1985

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

```

930          .TITLE  TSV2 - PROGRAM HEADER
931          .SBTTL  PROGRAM HEADER
932
938          .MCALL  SVC
939 000000    SVC          ; INITIALIZE SUPERVISOR MACROS
940          .ENABLE LC
941          .MLIST  BEX,CND
947 000000    .ENABL  ABS,AMA
948          .=2000
949 002000    BGNMOD  TSV2
          002000
          TSV2::
          ;**
          ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
          ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
          ;--
          POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
          HEADER  CVTSB,D,0,655,,0
          L#NAME::          ;DIAGNOSTIC NAME
          002000    103    .ASCII  /C/
          002001    126    .ASCII  /V/
          002002    124    .ASCII  /T/
          002003    123    .ASCII  /S/
          002004    102    .ASCII  /B/
          002005    000    .BYTE   0
          002006    000    .BYTE   0
          002007    000    .BYTE   0
          002010          L#REV::          ;REVISION LEVEL
          002010    104    .ASCII  /D/
          002011          L#DEPO::          ;0
          002011    060    .ASCII  /O/
          002012          L#UNIT::          ;NUMBER OF UNITS
          002012    000000 .WORD   0
          002014          L#TIML::          ;LONGEST TEST TIME
          002014    001217 .WORD   655.
          002016          L#HPCP::          ;PTR. TO H.W. QUES.
          002016    101242 .WORD   L#HARD
          002020          L#SPCP::          ;PTR. TO S.W. QUES.
          002020    101374 .WORD   L#SOFT
          002022          L#HPTP::          ;PTR. TO DEF. H.W. PTABLE
          002022    002156 .WORD   L#HW
          002024          L#SPTP::          ;PTR. TO S.W. PTABLE
          002024    002166 .WORD   L#SW
          002026          L#LADP::          ;DIAG. END ADDRESS
          002026    102004 .WORD   L#LAST
          002030          L#STA::          ;RESERVED FOR APT STATS
          002030    000000 .WORD   0
          002032          L#CO::          ;DIAGNOSTIC TYPE
          002032    000000 .WORD   0
          002034          L#DTYP::          ;APT EXPANSION
          002034    000000 .WORD   0
          002036          L#APT::          ;PTR. TO DISPATCH TABLE
          002036    000000 .WORD   0
          002040          L#DTP::          ;DIAGNOSTIC RUN PRIORITY
          002040    002124 .WORD   L#DISPATCH
          002042          L#PRIO::

```

PROGRAM HEADER

002042	000000			
002044		L\$ENVI::	.WORD 0	;FLAGS DESCRIBE HOW IT WAS SETUP
002044	000000			
002046		L\$EXP1::	.WORD 0	;EXPANSION WORD
002046	000000			
002050		L\$MREV::	.WORD 0	;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::	.WORD 0	
002056	000000			
002060		L\$DEVP::		; POINTER TO DEVICE TYPE LIST
002060	003402		.WORD L\$DVTYP	
002062		L\$REPP::		;PTR. TO REPORT CODE
002062	022214		.WORD L\$RPT	
002064		L\$EXP4::	.WORD 0	
002064	C00000			
002066		L\$EXP5::	.WORD 0	
002066	000000			
002070		L\$AUT::		;PTR. TO ADD UNIT CODE
002070	021702		.WORD L\$AU	
002072		L\$DUT::		;PTR. TO DROP UNIT CODE
002072	022000		.WORD L\$DU	
002074		L\$LUN::		;LUN FOR EXERCISERS TO FILL
002074	000000		.WORD 0	
002076		L\$DESP::		;POINTER TO DIAG. DESCRIPTION
002076	003410		.WORD L\$DESC	
002100		L\$LOAD::		;GENERATE SPECIAL AUTOLOAD EMT
002100	104035		EMT E\$LOAD	
002102		L\$ETP::		;POINTER TO ERR_TBL
002102	000000		.WORD 0	
002104		L\$ICP::		;PTR. TO INIT CODE
002104	021106		.WORD L\$INIT	
002106		L\$CCP::		;PTR. TO CLEAN-UP CODE
002106	022166		.WORD L\$CLEAN	
002110		L\$ACP::		;PTR. TO AUTO CODE
002110	022106		.WORD L\$AUTO	
002112		L\$PRT::		;PTR. TO PROTECT TABLE
002112	021076		.WORD L\$PROT	
002114		L\$TEST::		;TEST NUMBER
002114	000000		.WORD 0	
002116		L\$DLY::		;DELAY COUNT
002116	000000		.WORD 0	
002120		L\$HIME::		;PTR. TO HIGH MEM
002120	000000		.WORD 0	

DISPATCH TABLE

```

959                                     .SBTTL  DISPATCH TABLE
960
961                                     ;**
962                                     ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
963                                     ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
964                                     ;--
965
966                                     DISPATCH 12
967                                     .WORD 12
968                                     L#DISPATCH::
969                                     .WORD T1
970                                     .WORD T2
971                                     .WORD T3
972                                     .WORD T4
973                                     .WORD T5
974                                     .WORD T6
975                                     .WORD T7
976                                     .WORD T8
977                                     .WORD T9
978                                     .WORD T10
979                                     .WORD T11
980                                     .WORD T12
981
982 002122 000014
983 002124 022776
984 002126 023756
985 002130 025750
986 002132 031344
987 002134 034134
988 002136 037726
989 002140 050040
990 002142 051320
991 002144 C62146
992 002146 066216
993 002150 074054
994 002152 077564
```

DEFAULT HARDWARE P-TABLE

```
968 .SBTTL DEFAULT HARDWARE P-TABLE
969
970 ;**
971 ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
972 ; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
973 ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
974 ;--
975 002154 BGNHW DFPTBL ;DEFAULT HARD-P-TABLE
002154 .WORD L10000-L$HW/2
002156 L$HW::
002156 DFPTBL::
976
977 002156 172520 .WORD 172520 ; 1ST (OF 2) REGISTERS.
978 002160 000224 .WORD 224 ; INTERRUPT VECTOR
979 002162 000200 .WORD PRI04 ; INTERRUPT PRIORITY.
980 002164 ENDPHW
002164 L10000:
```

SOFTWARE P-TABLE

```
982          .SBTTL  SOFTWARE P-TABLE
983
984          ;**
985          ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
986          ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
987          ;--
988          002164      BGNSW      SFPTBL
          002164      .WORD      L10001-L#SW/2
          002166
          002166      L#SW::
          002166      SFPTBL::
989
990          002166      000000      TRANSTST::      .WORD      0      ; ENABLE TEST OF TRANSPORT(S) IF =1
991          002170      000000      NOITS::      .WORD      0      ; INHIBIT ITERATION OPTION.
992          ; ... 0 = ITERATE.
993          ; ... NZ = INHIBIT ITERATE.
994          002172      000017      LERRMAX::      .WORD      15.      ; LOCAL (PER TEST) ERROR LIMIT
995          002174      000000      GERRMAX::      .WORD      200.      ; GLOBAL (PER UNIT) ERROR LIMIT
996          002176      ENDSW
          002176      L10001:
997
998          002176      ENDMOD
```

SOFTWARE P-TABLE

```

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

;
; BIT DEFINITIONS
;
100000      BIT15== 100000
040000      BIT14== 40000
020000      BIT13== 20000
010000      BIT12== 10000
004000      BIT11== 4000
002000      BIT10== 2000
001000      BIT09== 1000
000400      BIT08== 400
000200      BIT07== 200
000100      BIT06== 100
000040      BIT05== 40
000020      BIT04== 20
000010      BIT03== 10
000004      BIT02== 4
000002      BIT01== 2
000001      BIT00== 1

;
001000      BIT9==  BIT09
000400      BIT8==  BIT08
000200      BIT7==  BIT07
000100      BIT6==  BIT06
000040      BIT5==  BIT05
000020      BIT4==  BIT04
000010      BIT3==  BIT03
000004      BIT2==  BIT02
000002      BIT1==  BIT01
000001      BIT0==  BIT00

;
; EVENT FLAG DEFINITIONS
; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
;
000040      EF.START==      32.      ; START COMMAND WAS ISSUED
000037      EF.RESTART==    31.      ; RESTART COMMAND WAS ISSUED
000036      EF.CONTINUE==   30.      ; CONTINUE COMMAND WAS ISSUED
000035      EF.NEW==        29.      ; A NEW PASS HAS BEEN STARTED
000034      EF.PWR==        28.      ; A POWER-FAIL/POWER-UP OCCURRED

;
; PRIORITY LEVEL DEFINITIONS

```

GLOBAL EQUATES SECTION

```

000340      PRI07== 340
000300      PRI06== 300
000240      PRI05== 240
000200      PRI04== 200
000140      PRI03== 140
000100      PRI02== 100
000040      PRI01== 40
000000      PRI00== 0

```

; OPERATOR FLAG BITS

```

000004      EVL==      4
000010      LOT==     10
000020      ADR==     20
000040      IDU==     40
000100      ISR==    100
000200      UAM==    200
000400      BOE==    400
001000      PNT==   1000
002000      PRI==   2000
004000      IXE==   4000
010000      IBE==  10000
020000      IER==  20000
040000      LOE==  40000
100000      MOE== 100000

```

```

1034
1035 002176

```

000250

```

177572
177574
177576
172516

```

;KT11 MEMORY MANAGEMENT DEFINITIONS

; *KT11 VECTOR ADDRESS

MMVEC= 250

; *KT11 STATUS REGISTER ADDRESSES

```

SR0= 177572
SR1= 177574
SR2= 177576
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 REGISTERS

```

UDPDR0= 177620
UDPDR1= 177622
UDPDR2= 177624
UDPDR3= 177626
UDPDR4= 177630
UDPDR5= 177632
UDPDR6= 177634
UDPDR7= 177636

```

.ENDC

; DEFINE MEMORY MANAGEMENT REGISTERS

MEMORY MANAGEMENT DEFINITIONS

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

MEMORY MANAGEMENT DEFINITIONS

```
SDPAR4= 172270
SDPAR5= 172272
SDPAR6= 172274
SDPAR7= 172276
.ENDC
.ENDC
; *KERNEL "I" PAGE DESCRIPTOR REGISTERS
172300 KIPDR0= 172300
172302 KIPDR1= 172302
172304 KIPDR2= 172304
172306 KIPDR3= 172306
172310 KIPDR4= 172310
172312 KIPDR5= 172312
172314 KIPDR6= 172314
172316 KIPDR7= 172316
; IF NB
; *KERNEL "D" PAGE
DESCRIPTOR REGISTERS
KOPDR0= 172320
KOPDR1= 172322
KOPDR2= 172324
KOPDR3= 172326
KOPDR4= 172330
KOPDR5= 172332
KOPDR6= 172334
KOPDR7= 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
KOPAR0= 172360
KOPAR1= 172362
KOPAR2= 172364
KOPAR3= 172366
KOPAR4= 172370
KOPAR5= 172372
KOPAR6= 172374
KOPAR7= 172376
.ENDC
```

TSV05 REGISTER AND PACKET DEFINITIONS

```

1040          .SBTTL  TSV05 REGISTER AND PACKET DEFINITIONS
1041
1042          ;
1043          ; SOME GENERAL EQUATES.
1044          ;
1045
1046          000004  ERRVEC==      4          ; POINTER TO ERROR VECTOR FOR BUS TIME OUT.
1047          000060  TTIVEC==     60          ; INTERRUPT VECTOR FOR CONSOLE INPUT
1048          177560  TTICSR==    177560       ; BUS ADDRESS OF CONSOLE INPUT
1049          177562  TTIBFR==    177562       ; CONSOLE INPUT DATA BUFFER
1050          177520  BDVPCR==    177520       ; BDV11 PAGE CONTROL REGISTER
1051
1052          ;*
1053          ;BIT DEFINITIONS FOR TSSR REGISTER
1054          ;-
1055
1056          100000  SC=      BIT15          ;SPECIAL CONDITION
1057          040000  BIE=     BIT14          ;BUS INTERFACE ERROR
1058          C20000  SCE=     BIT13          ;SANITY CHECK ERROR
1059          010000  RMR=     BIT12          ;MODIFICATION REFUSED
1060          004000  NXM=     BIT11          ;NONEXISTANT MEMORY ERROR
1061          002000  NBA=     BIT10          ;NEED BUFFER ADDRESS
1062          001400  MIADDR= BIT9:BIT8      ;EXTENDED ADDRESS BITS
1063          000200  SSR=     BIT7           ;SUB SYSTEM READY
1064          000100  OFL=     BIT6           ;OFF LINE BIT
1065          000060  FATERR= BIT4:BIT5      ;FATAL TERMINATION ERROR CODES
1066          000016  TERCLS= BIT3:BIT2:BIT1 ;TERMINATION CODES
1067
1068          ;*
1069          ;
1070          ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
1071          ;(XST0)
1072          ;
1073          ;-
1074
1075          100000  XSOTMK= BIT15          ;TAPE MARK DETECTED
1076          040000  XSORLS= BIT14          ;RECORD LENGTH SHORT
1077          020000  XSOLET= BIT13          ;LOGICAL END OF TAPE
1078          010000  XSORLL= BIT12          ;RECORD LENGTH LONG
1079          004000  XSOWLE= BIT11          ;WRITE LOCK ERROR
1080          002000  XSONEF= BIT10          ;NON EXECUTABLE FUNCTION
1081          001000  XSOILC= BIT9           ;ILLEGAL COMMAND
1082          000400  XSOILA= BIT8           ;ILLEGAL ADDRESS
1083          000200  XSOMOT= BIT7           ;TAPE IN MOTION
1084          000100  XSOONL= BIT6           ;TRANSPORT ON LINE
1085          000040  XSOIE=  BIT5           ;INTERRUPT ENABLE
1086          000020  XSOVCK= BIT4           ;VOLUME CHECK BIT
1087          000010  XSOPED= BIT3           ;PHASE ENCODED DRIVE
1088          000004  XSOWLK= BIT2           ;WRITE LOCKED
1089          000002  XS0BOT= BIT1           ;BEGINNING OF TAPE
1090          000001  XS0EOT= BIT0           ;END OF TAPE

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

1092      ;*
1093      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
1094      ;(XST1)
1095      ;-
1096      100000      X1.DLT = BIT15      ;DATA LATE
1097      040000      X1.SPARE= BIT14      ;NOT USED
1098      020000      X1.COR = BIT13      ;CORRECTABLE DATA ERROR
1099      017375      X1.MBZ = BIT12+BIT11+BIT10+BIT9+BIT7+BIT6+BIT5+BIT4+BIT3+BIT2+BIT0 ;ALWAYS 0
1100      000400      X1.RBP = BIT8      ;READ BUS PARITY ERROR
1101      000002      Y1.UNC = BIT1      ;UNCORRECTABLE DATA OR HARD ERROR
1102
1103      ;*
1104      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
1105      ;(XST2)
1106      ;-
1107      100000      X2.OPM = BIT15      ;OPERATION IN PROGRESS (TAPE MOVING)
1108      040000      X2.RCE = BIT14      ;RAM CHECKSUM ERROR
1109      035400      X2.SPARE= BIT13+BIT12+BIT11+BIT9+BIT8 ;NOT USED BY TSV05 (ALWAYS=0)
1110      C02000      X2.WCF = BIT10      ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
1111      000200      X2.EXTF = BIT7      ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
1112      000100      X2.BUFE = BIT6      ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
1113      000077      X2.REV = 000077      ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
1114      000007      X2.UNIT = BIT2+BIT1+BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.
1115
1116      ;*
1117      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
1118      ;(XST3)
1119      ;-
1120      177400      X3.MDE = 177400      ;MICRO-DIAGNOSTIC ERROR CODE
1121      000200      X3.SPARE= BIT7      ;NOT USED BY TSV05
1122      000100      X3.OPI = BIT6      ;OPERATION INCOMPLETE
1123      000040      X3.REV = BIT5      ;REVERSE
1124      000020      X3.TRF = BIT4      ;TRANSPORT RESPONSE FAILURE
1125      000010      X3.DCK = BIT3      ;DENSITY CHECK
1126      000006      X3.MBZ =BIT2+BIT1      ;NOT USED ALWAYS 0
1127      000001      X3.RIB = BIT0      ;REVERSE INTO BOT
1128
1129      ;*
1130      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
1131      ;(XST4)
1132      ;-
1133      100000      X4.HSP = BIT15      ;HIGH SPEED
1134      040000      X4.RCE = BIT14      ;RETRY COUNT EXCEEDED
1135      020000      X4.TSM = BIT13      ;TRANSPORT SPECIAL MODE
1136      017400      X4.MBZ = BIT12+BIT11+BIT10+BIT9+BIT8 ;NOT USED ALWAYS 0
1137      000377      X4.WRC = 000377      ;WRITE RETRY COUNT FIELD
1138
1139      ;*
1140      ;
1141      ;TSSR TERMINATION CODES (BIT 0-2)
1142      ;
1143      ;-
1144
1145      000006      TSREJ= 3+2      ;COMMAND REJECTED
1146      000006      UNREC= 6      ;UNRECOVERABLE ERROR

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

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

```

TSV05 REGISTER AND PACKET DEFINITIONS

```
1199      ;*
1200      ; CONTROLLER RAM DEFINITIONS
1201      ; -
1202      000167      RMCHBEG = 167      ; CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
1203      000200      RMCHEND = 200      ; CHARACTERISTICS IO DATA END RAM ADDRESS
1204      000201      RMPKTBEGB = 201      ; COMMAND PACKET BEGIN RAM ADDRESS
1205      000210      RMPKTEND = 210      ; COMMAND PACKET END RAM ADDRESS
1206      000215      RMMSGBEG = 215      ; MESSAGE BUFFER BEGIN RAM ADDRESS
1207      000234      RMMSGEND = 234      ; MESSAGE BUFFER END RAM ADDRESS
1208      ;*
1209      ;
1210      ; REGISTER DEFINITIONS IN THE MESSAGE BUFFER
1211      ; -
1212      ; -
1213      ;
1214      000006      XST0 = 6      ; EXTENDED STATUS REGISTER 0 (WORD 4)
1215      000010      XST1 = 8      ; EXTENDED STATUS REGISTER 1 (WORD 5)
1216      000012      XST2 = 10      ; EXTENDED STATUS REGISTER 2 (WORD 6)
1217      000014      XST3 = 12      ; EXTENDED STATUS REGISTER 3 (WORD 7)
1218      000016      XST4 = 14      ; EXTENDED STATUS REGISTER 4 (WORD 8)
1219      ;
1220      ;*
1221      ;
1222      ; OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
1223      ; -
1224      ; -
1225      ;
1226      000002      PKLOW = 2      ; LOW ORDER CHARACTERISTIC DATA POINTER
1227      000004      PKHI = 4      ; HIGH ORDER CHARACTERISTIC DATA POINTER
1228      000006      PKBCNT = 6      ; NUMBER OF BYTES IN DATA PACKET
1229      ;
1230      000010      EXBCNT = 10      ; NUMBER OF BYTES IN EXTENDED DATA PACKET
1231      ;
1232      ;*
1233      ; DATA PACKET OFFSETS FOR WRITE SUBSYSTEM COMMAND
1234      ; -
1235      000000      BSELO = 0      ; BYTE 0
1236      000001      BSEL1 = 1      ; BYTE 1
1237      000002      SEL2 = 2      ; WORD 2
1238      000004      SELDATA = 4      ; WORD 3
```

TSV05 REGISTER AND PACKET DEFINITIONS

```

1240
1241      ;*
1242      ;BSEL0 SELECT CODES FOR WRITE SUBSYSTEM COMMAND
1243      ;-
1243      000000      PW.NOP          = 0          ;NO-OP
1244      000001      PW.RDRAM        = 1          ;READ RAM
1245      000002      PW.WTRAM        = 2          ;WRITE RAM
1246      000003      PW.RFIFO        = 3          ;READ FIFO
1247      000004      PW.WFIFO        = 4          ;WRITE FIFO
1248      000005      PW.RDSTAT       = 5          ;READ STATUS
1249      000006      PW.WCTL         = 6          ;WRITE TAPE CONTROL
1250      000007      PW.WFMT         = 7          ;WRITE TAPE FORMAT
1251      000010      PW.WMISC        = 10         ;WRITE MISCELLANEOUS
1252      000011      PW.WNPR        = 11         ;WRITE NPR CONTROL
1253      000020      PW.D22         = 20         ;DO MICROTEST 22
1254      000021      PW.D11         = 21         ;DO MICROTEST 11
1255      000022      PW.D13         = 22         ;DO MICROTEST 13
1256      000023      PW.NO1311      = 23         ;DISABLE MICROTEST 11 AND 13
1257      000024      PW.RDXT        = 24         ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS)
1258
1259      ;*
1260      ;BSEL1 CODES FOR WRITE TAPE CONTROL
1261      ;-
1262      000200      WC.IFAD          = BIT7        ;IFAD - FORMATTER ADDRESS
1263      000100      WC.IOTAD        = BIT6        ;ITADO - TRANSPORT ADDRESS BIT 0
1264      000040      WC.I1TAD        = BIT5        ;ITAD1 - TRANSPORT ADDRESS BIT 1
1265      000020      WC.ISRESV       = BIT4        ;IRESV5 - RESERVED #5
1266      000010      WC.IREW        = BIT3        ;IREW - REWIND
1267      000004      WC.IRWU        = BIT2        ;IRWU - REWIND AND UNLOAD
1268      000002      WC.IFEN        = BIT1        ;IFEN - FORMATTER ENABLE
1269      000001      WC.IGO         = BIT0        ;GO
1270
1271      ;*
1272      ;BSEL1 CODES FOR WRITE FORMAT
1273      ;-
1274      000200      WF.IHISP        = BIT7        ;IHISP - HIGH SPEED
1275      000100      WF.IWRT        = BIT6        ;IWRT - WRITE
1276      000040      WF.IREV        = BIT5        ;IREV - REVERSE
1277      000020      WF.IWFM        = BIT4        ;IWFM - WRITE FILE MARK
1278      000010      WF.IEDIT       = BIT3        ;IEDIT - EDIT
1279      000004      WF.IERASE      = BIT2        ;IERASE - ERASE
1280      000002      WF.I3RESV      = BIT1        ;IRESV3 - RESERVED #3
1281      000001      WF.I4RESV      = BIT0        ;IRESV4 - RESERVED #4
1282
1283      ;*
1284      ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
1285      ;-
1286      000200      MS.EXT          = BIT7        ;INVERT SENSE OF EXTENDED FEATURES SWITCH
1287      000020      MS.RSFIFO       = BIT4        ;RESET FIFO AND INPUT PARITY ERRORR
1288      000010      MS.RSTAPE      = BIT3        ;RESET TAPE STATUS IN 2 FLIP-FLOPS
1289      000006      MS.ATTN        = BIT2!BIT1    ;ATTENTION TRIGGER FIELD
1290      000001      MS.RSD         = BIT0        ;RESET TIMER A,B THEN DELAY TIMES IN SEL2

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

1292      ;*
1293      ; MS.ATTN SUBCODES
1294      ; -
1295      000000      MSA.NOP = 0*2      ;NO-OP (NOTHING TRIGGERED)
1296      000002      MSA.VOL = 1*2      ;SIMULATE ON-LINE/OFF-LINE TRANSISTION
1297      000004      MSA.NRAM= 2*2      ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
1298      000006      MSA.FRAME= 3*2      ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
1299
1300      ;*
1301      ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
1302      ; -
1303      000200      NP.IR      = BIT7      ;INTERRUPT REQUEST (0-1 TRANSITION)
1304      000100      NP.OUT     = BIT6      ;TAPE DATA DIRECTION OUT (0= IN)
1305      000040      NP.LOOP    = BIT5      ;ENABLE TRANSPORT LOOPBACK
1306      000020      NP.WRP     = BIT4      ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
1307
1308      ;*
1309      ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
1310      ; -
1311      000200      S2.DIM      = BIT7      ;WORD #9 BYTE 2 DATA IN MISS
1312      000100      S2.ILW     = BIT6      ;ILW H
1313      000040      S2.OTRDY    = BIT5      ;OUT RDY H
1314      000020      S2.INRDY    = BIT4      ;IN RDY H
1315      000010      S2.ATIMR    = BIT3      ;TIMER A FLAG H
1316      000004      S2.BTIMR    = BIT2      ;TIMER B FLAG H
1317      000003      S2.UNDEF    = BIT1,BIT0 ; (UNDEFINED)
1318      100000      S1.PARIN     = BIT15      ;WORD #8 BYTE 1 PARIN H
1319      040000      S1.I2RESV    = BIT14      ;IRESV2
1320      020000      S1.I1RESV    = BIT13      ;IRESV1
1321      010000      S1.IEOT      = BIT12      ;IEOT L
1322      004000      S1.IIDENT    = BIT11      ;IIDENT H
1323      002000      S1.ICER      = BIT10      ;ICER H
1324      001000      S1.IFMK      = BIT9       ;IFMK H
1325      000400      S1.IHER      = BIT8       ;IHER H
1326      000200      S0.ISPEED    = BIT7      ;WORD #8 BYTE 0 ISPEED H
1327      000100      S0.IRDY      = BIT6      ;IRDY L
1328      000040      S0.IONL      = BIT5      ;IONL L
1329      000020      S0.ILOP      = BIT4      ;ILOP L
1330      000010      S0.IDBY      = BIT3      ;IDBY L
1331      000004      S0.IRWD      = BIT2      ;IRWD L
1332      000002      S0.IFBY      = BIT1      ;IFBY L
1333      000001      S0.IFPT      = BIT0      ;IFPT L

```

SPECIAL MACROS AND OPDEFS.

```

1334             .SBTTL SPECIAL MACROS AND OPDEFS.
1335
1336             ;+
1337             ;SAVE GENERAL REGS 1 TO 5
1338             ;-
1339
1340             .MACRO SAVREG
1341             JSR     R5,REGSAV
1342             .ENDM
1343
1344             ;+
1345             ; MACRO TO FORCE AN ERROR
1346             ;-
1347             .MACRO FORCERROR TAG,NOTSSR
1348             .NLIST
1349             .IIF NDF LISTALL, .NLIST
1350             .LIST
1351             .IF B NOTSSR
1352             MOV     TSSR(R5),R1      ;READ TSSR
1353             .ENDC
1354             MOV     FORCER,FORCER    ;IS FORCER SET? (LEAVE C BIT ALONE)
1355             BNE     TAG              ;BR IF YES
1356             .NLIST
1357             .IIF NDF LISTALL, .LIST
1358             .LIST
1359             .ENDM
1360
1361             ;+
1362             ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
1363             ; WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
1364             ; SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
1365             ; FORCER TO 177777
1366             ; TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
1367             ;-
1368             .MACRO FORCEEXIT TAG
1369             .NLIST
1370             .IIF NDF LISTALL, .NLIST
1371             .LIST
1372             MOV     FORCER,FORCER    ;IS FORCER NEGATIVE?
1373             BMI     TAG              ;BR IF YES
1374             .NLIST
1375             .IIF NDF LISTALL, .LIST
1376             .LIST
1377             .ENDM
1378
1379             ;+
1380             ; MACRO TO INCREMENT ERROR COUNTS
1381             ;-
1382             .MACRO NEXT.ERRNO
1383             .NLIST
1384             .IIF NDF LISTALL, .NLIST
1385             ERRNO=ERRNO+1
1386             .IIF NDF LISTALL, .LIST
1387             .LIST
1388             .ENDM

```

SPECIAL MACROS AND OPDEFS.

```
1389
1390      ;+
1391      ;MACRO TO PERFORM XOR
1392      ;-
1393
1394      .MACRO XOR      A,B
1395      MOV      A,-(SP)
1396      BIC      B,(SP)
1397      BIC      A,B
1398      BIS      (SP)+,B
1399      .ENDM
1400      EN=0      000000      ; INITIALIZE ERROR NUMBER
1401      .SBTTL    FORCER - FORCE ERROR FLAG
1402
1403      ;
1404      ; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
1405      ; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
1406      ;
1407
1408      002176    000000      FORCER::      0      ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED -
1409      ; - BY THE MACRO "IFERROR"). AN ERROR NEED NOT -
1410      ; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.
```

GLOBAL DATA SECTION

```
1412          .SBTTL  GLOBAL DATA SECTION
1413
1414          ;**
1415          ;THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1416          ;IN MORE THAN ONE TEST.
1417          ;--
1418
1419          ;
1420          ;THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
1421          ;SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.
1422          ;
1423          EPRTSW::      .WORD  0      ;PRINT SWITCH
1424          UNITN::      .WORD  0      ;UNIT # UNDER TEST.
1425          QVP::        .WORD  0      ;QUICK VERIFY FLAG.
1426          CSRADDR::    .WORD  0      ;ADDRESS OF CSR FOR CURRENT DEVICE
1427          IVEC::       .WORD  224    ;INTERRUPT VECTOR
1428          IPRI::       .WORD  PRI04  ;INTERRUPT PRIORITY.
1429          TSTCNT::     .WORD  0      ;NUMBER OF TESTS RUN IN THIS PASS
1430          LOOPCNT::    .WORD  0      ;REMAINING RATION COUNT FOR TEST
1431          DEVCNT::     .WORD  0      ;NUMBER OF DEVICE UNDER TEST
1432          FATFLG::     .WORD  0      ;SET IF FATAL ERROR IS DETECTED IN TEST
1433          INTRECV::    .WORD  0      ;SET IF TAPE INTERRUPT WAS RECEIVED
1434          EXTFEA::     .WORD  0      ;EXTENDED FEATURES SOFTWARE SW 0-OFF;1-ON
1435          BENBSW::     .WORD  0      ;BUFFER ENABLE SWITCH SW 0-OFF;1-ON
1436          EXPD::       .WORD  0      ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
1437          RECV::       .WORD  0      ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
1438          ERRHI::      .WORD  0      ;HIGH ADDRESS MEMORY ERROR
1439          ERRLO::      .WORD  0      ;LOW ADDRESS MEMORY ERROR
1440          RAMDATA::    .BLKW  16.    ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
1441          RAMSIZ::     .WORD  0      ;RAM DATA SIZE FOR PRAMPKT ROUTINE
1442          RCVHIADD::   .WORD  0      ;RECEIVED BUFFER HIGH ADDRESS
1443          RCVLOADD::   .WORD  0      ;RECEIVED BUFFER LOW ADDRESS
1444          COUNT::     .WORD  0      ;TEST COUNT PATTERN
1445          DATA::      .WORD  0      ;TEST DATA
1446          TSTFLAG::    .WORD  0      ;TEST FLAG WORD
1447          TSTPTR::     .WORD  0      ;TSTBLK POINTER
1448          PRMNO::      .WORD  0      ;PRINT ROUTINE TEMP
1449          EXPMSG::     .BLKB  100.   ;EXPECTED MESSAGE BUFFER DATA
1450          RECHMSG::    .BLKB  100.   ;RECEIVED MESSAGE BUFFER DATA
1451          TMPBFR::     .BLKB  80.    ;TEMPORARY STORAGE FOR PRINT
```

TSTBLK - TEST DATA TABLE

```

1453          .SBTTL  TSTBLK  - TEST DATA TABLE
1454
1455          ;*
1456          ;THIS TABLE CONTAINS TEST DATA USED IN SEVERAL TESTS
1457          ;
1458          ;IN SEQUENCE THE DATA IS:
1459          ;
1460          ;      ALL ZEROS
1461          ;      ALL ONES
1462          ;      WALKING ONES
1463          ;      WALKING ZEROS
1464          ;      ALTERNATING ONES AND ZEROS
1465          ;
1466          ;-
1467
1468          TSTBLK::
1469          002752      000000      .WORD      0              ;ALL ZEROS
1470          002752      177777      .WORD      177777         ;ALL ONES
1471          002754      177777      .WORD      BIT0           ;DATA FOR WALKING ONES
1472          002756      000001      .WORD      BIT1
1473          002760      000002      .WORD      BIT2
1474          002762      000004      .WORD      BIT3
1475          002764      000010      .WORD      BIT4
1476          002766      000020      .WORD      BIT5
1477          002770      000040      .WORD      BIT6
1478          002772      000100      .WORD      BIT7
1479          002774      000200      .WORD      BIT8
1480          002776      000400      .WORD      BIT9
1481          003000      001000      .WORD      BIT10
1482          003002      002000      .WORD      BIT11
1483          003004      004000      .WORD      BIT12
1484          003006      010000      .WORD      BIT13
1485          003010      020000      .WORD      BIT14
1486          003012      040000      .WORD      BIT15
1487          003014      100000      .WORD      +CBIT0         ;DATA FOR WALKING ZEROS
1488          003016      177776      .WORD      +CBIT1
1489          003020      177775      .WORD      +CBIT2
1490          003022      177773      .WORD      +CBIT3
1491          003024      177767      .WORD      +CBIT4
1492          003026      177757      .WORD      +CBIT5
1493          003030      177737      .WORD      +CBIT6
1494          003032      177677      .WORD      +CBIT7
1495          003034      177577      .WORD      +CBIT8
1496          003036      177377      .WORD      +CBIT9
1497          003040      176777      .WORD      +CBIT10
1498          003042      175777      .WORD      +CBIT11
1499          003044      173777      .WORD      +CBIT12
1500          003046      167777      .WORD      +CBIT13
1501          003050      157777      .WORD      +CBIT14
1502          003052      137777      .WORD      +CBIT15
1503          003054      077777      .WORD      125252         ;ALTERNATING ONES, ZEROS
1504          003056      125252      .WORD      052525         ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
1505          003060      052525      .WORD      003062
1506          003062

```

TBLEND==.

GLOBAL ENVIRONMENT STORAGE

```

1508 .SBTTL GLOBAL ENVIRONMENT STORAGE
1509
1510 ; STORAGE FOR DEVICE REGISTERS
1511
1512 003062 000000 100000 000000 DUMMY: 0,100000,0,0 ; DUMMY DEVICE REGISTERS...
1513 003072 000000 000000 000000 0,0,0,0,0,0,0,0,0
1514 ; ...FOR MULTI-UNIT CHECKOUT.
1515
1516 003112 000000 DUFLG:: .WORD 0 ; "DROPPED UNIT" FLAG.
1517 ; INHIBITS CODE IN "CLEAN-UP".
1518 003114 000000 NODEV:: .WORD 0 ; FLAG TO SAY NO DEVICE.
1519
1520 003116 000000 TEMP1:: .WORD 0 ; SOME TEMP LOCATIONS.
1521 003120 000000 TEMP2:: .WORD 0
1522 003122 000000 XXCOMM:: .WORD 0 ; XXDP+ COMM BLOCK POINTER.
1523 003124 000000 FREE:: .WORD 0 ; 1ST FREE MEMORY ADDRESS...
1524 003126 000000 FRESIZ:: .WORD 0 ; ...AND SIZE (IN WORDS).
1525 003130 000000 FREEHI: .WORD 0 ; LAST WORD IN FREE SPACE
1526 003132 000000 KTFLG:: .WORD 0 ; KT11, MEM AVAIL FLAG -
1527 ; - .WORD 0 = <24K OR NO KT -
1528 ; - NZ = >24K AND KT.
1529 003134 000000 KTENABLE:: .WORD 0 ; SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
1530 003136 000000 NXMFLG:: .WORD 0 ; SET IF WE CAN TEST CLEARED OTHERWISE
1531 003140 000000 NXMLO:: .WORD 0 ; NXM LO ADDRESS BITS
1532 003142 000000 NXMHI:: .WORD 0 ; NXM HI ADDRESS BITS FOR DAL'S 16-21
1533 003144 000000 T23A:: .WORD 0 ; 11/23A FLAG
1534 003146 000000 T23B:: .WORD 0 ; 11/23B FLAG
1535 003150 000000 T38FLG:: .WORD 0 ; TEST 38 FLAG +0
1536 003152 002000 PST32W:: .WORD 2000 ; 32W BLOCK ADDRESS FOR 32K START
1537 003154 000000 SIFLAG:: .WORD 0
1538 003156 000000 BADDAT:: .WORD 0 ; ACTUAL DATA
1539 003160 000000 GODAT:: .WORD 0 ; EXPECTED DATA
1540 003162 000000 LOOPFL:: .WORD 0
1541 003164 CTAB:: ; CONFIGURATION TABLES.
1542 003164 000000 CTABM:: .WORD 0 ; CONFIG WORK.
1543 003166 000000
1544 003170 000000
1545 003172 000000
1546 003174 177777
1547 003176
1548 CTABE::
1549 ; ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
1550 ;
1551 ; 0 = UNIT NOT TESTED
1552 ; 100000 = UNIT ONLINE, NO ERRORS
1553 ; 10XXXX = UNIT ONLINE, ENCOUNTERED XXXX ERRORS
1554 ; 160000 = UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
1555 ; 160001 = UNIT DROPPED, NOT IDLE AT START
1556 ; 14XXXX = UNIT DROPPED, ENCOUNTERED XXXX ERRORS
1557 003176 ERTABL: .BLKW 64.
1558 003376 000000 ERTABE: .WORD 0
1559
1560 003400 000000 SKIPT: .WORD 0 ; 1-SKIP SUBTEST 0=NO SKIP OF SUBTEST

```

GLOBAL TEXT MESSAGES

```

1562          .SBTTL  GLOBAL TEXT MESSAGES
1563
1564          ;*
1565          ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
1566          ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
1567          ; MORE THAN ONE TEST.
1568          ;--
1569          ;*
1570          ; NAMES OF DEVICES SUPPORTED
1571          ;-
1571 003402      DEVTYP  <TSV05>
1571 003402      L$DVTYP:  .ASCIZ  /TSV05/
1571 003402      .EVEN
1572
1572          ;*
1572          ; TEST DESCRIPTION
1572          ;-
1572 003410      DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****>
1572 003410      L$DESC:  .ASCIZ  /**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****/
1572 003410      .EVEN
1572
1572          ;*
1572          ; BIT TO ASCII CONVERSION FOR TSSR REGISTER
1572          ;-
1572 003502      003542  003545  003551  TSSRBIT:  .WORD  1$,2$,3$,4$,5$,6$,7$,8$
1572 003522      003603  003607  003613  .WORD  9$,10$,11$,12$,13$,14$,15$,16$
1572 003542      123      103      000  1$:  .ASCIZ  'SC'
1572 003542      102      111      105  2$:  .ASCIZ  'BIE'
1572 003545      123      103      105  3$:  .ASCIZ  'SCE'
1572 003551      122      115      122  4$:  .ASCIZ  'RMR'
1572 003555      116      130      115  5$:  .ASCIZ  'NXM'
1572 003561      116      102      101  6$:  .ASCIZ  'NBA'
1572 003565      102      111      124  7$:  .ASCIZ  'BIT9'
1572 003571      102      111      124  8$:  .ASCIZ  'BIT8'
1572 003576      123      123      122  9$:  .ASCIZ  'SSR'
1572 003603      117      106      114  10$: .ASCIZ  'OFL'
1572 003607      102      111      124  11$: .ASCIZ  'BIT5'
1572 003613      102      111      124  12$: .ASCIZ  'BIT4'
1572 003620      102      111      124  13$: .ASCIZ  'BIT3'
1572 003625      102      111      124  14$: .ASCIZ  'BIT2'
1572 003632      102      111      124  15$: .ASCIZ  'BIT1'
1572 003637      102      111      124  16$: .ASCIZ  'BIT0'
1572 003644      102      111      124  .EVEN
1572
1572 003652      124      123      123  SFIERR: .ASCIZ  'TSSR ERROR AFTER SOFT INIT'
1572 003705      124      123      123  SFHERR: .ASCIZ  'TSSR ERROR AFTER BUS RESET'
1572 003740      040      040      116  NXR:  .ASCIZ  / NON-EXISTANT DEVICE REGISTER/
1572 003777      045      101      040  NXR:  .ASCIZ  /#A ADDRESS: #06/
1572 004020      045      101      040  TSSX:  .ASCII  /#A TSBA,TSSR EXP'D: #06#A,#06#N/
1572 004060      045      101      040  TSSX:  .ASCIZ  /#A TSBA,TSSR REC'D: #06#A,#06/
1572 004117      045      116      045  FUSI:  .ASCII  /#N#A/
1572 004123      040      040      125  USI:  .ASCIZ  / UNEXPECTED INTERRUPT/
1572 004152      040      040      111  NSI:  .ASCIZ  / INTERRUPT EXPECTED, NOT RECEIVED/
1572 004215      045      116      045  FNOINTR: .ASCII  /#N#A/
1572 004221      040      040      116  NOINTR: .ASCIZ  / NO INTERRUPT WAS GENERATED'
1572 004256      040      040      111  IFAULT: .ASCIZ  / INTERRUPT FAULT/

```

GLOBAL TEXT MESSAGES

```

1633 004300      045      101      040 INTX: .ASCIZ /#A CPU PC: #06#A TSBA: #06/
1634 004335      040      040      042 NOINIT: .ASCIZ / "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
1635 004407      040      040      042 NSINIT: .ASCIZ / "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
1636 004457      040      040      042 BRINIT: .ASCIZ / "BUS-RESET" DIDN'T INITIALIZE THE DPU/
1637 004527      000      000      000 NUL: .ASCIZ //
1638 004530      045      116      000 NULCR: .ASCIZ /#N/
1639 004533      045      101      040 EXPGOT: .ASCIZ /#A EXP'D: #06#A, REC'D: #06/
1640 004567      045      116      045 EXPGT2: .ASCIZ /#N#A EXP'D: #06#A, #06#N#A REC'D: #0#A, #06/
1641 004643      045      101      040 DUAD12: .ASCIZ /#A REG(W) WRITTEN TO: #06#A REG(R) READ; EXP'D: #06#A, PEC'D: #06/
1642 004745      122      101      115 PKTRAM: .ASCIZ 'RAM Contents Do Not Match Packet Sent'
1643 005013      040      040      103 SCME: .ASCIZ / CONFIG DOESN'T MATCH MFG. MASTER/
1644 005056      127      122      111 WRTMSG: .ASCIZ 'WRITE CHARACTERISTICS Failed'
1645 005113      124      123      123 WRTERR: .ASCIZ 'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
1646 005206      124      123      123 RDERR: .ASCIZ 'TSSR Incorrect After READ Command, More Bits Set Than SSR'
1647 005300      106      101      124 SCHERR: .ASCIZ 'FATAL ERROR IN SUBTEST - CHECK TAPE, CABLES, TRANSPORT etc.'
1648 005372      105      122      122 RETERR: .ASCIZ 'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
1649 005460      045      116      045 NOMEM: .ASCIZ '#N#A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****N'
1650 005554      045      116      045 M8186: .ASCIZ '#N#A ***** 11/23A SYSTEM *****N'
1651 005645      045      116      045 M8189: .ASCIZ '#N#A ***** 11/23B SYSTEM *****N'
1652      .EVEN
1653      .SBTTL GLOBAL ERROR REPORT SECTION
1654
1655      ;
1656      ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
1657      ; CALLS THAT ARE USED IN MORE THAN ONE TEST.
1658      ; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
1659      ;
1660      BGNMSG NXRRERR ;NON-EXISTANT DEVICE REGISTER.
1661      NXRRERR: PRINTX #NXRX, NODEV ;NODEV = NEXM ADDRESS.
1662      MOV NODEV, -(SP)
1663      MOV #NXRX, -(SP)
1664      MOV #2, -(SP)
1665      MOV SP, R0
1666      TRAP C#PNTX
1667      ADD #6, SP
1668      JSR PC, EXTEND ; PRINT EXTENSION IF REQUIRED.
1669      ENDMMSG
1670      L10002: TRAP C#MSG
1671      ;
1672      ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
1673      ; TO ANY OF THE ABOVE ERROR SIGNATURES.
1674      ;
1675      EXTEND: TST (PC)+
1676      EXTA: 0 ; 0 = NO EXTENSION.
1677      BEQ 1#
1678      JSR PC, #EXTA ; APPEND EXTENSION TEXT.
1679      1#: PRINTX #NULCR ; PRINT A BLANK LINE
1680      MOV #NULCR, -(SP)
1681      MOV #1, -(SP)
1682      MOV SP, R0
1683      TRAP C#PNTX
1684      ADD #4, SP
1685      RTS PC

```

PRITSSR - PRINT TSSR CONTENTS

```

1675 .SBTTL PRITSSR - PRINT TSSR CONTENTS
1676
1677
1678
1679 ;ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
1680 ;THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
1681 ;BY A MESSAGE PRINTING ROUTINE
1682
1683 ;INPUTS:
1684
1685 ; R1 CONTENTS OF TSSR
1686
1687 ;SUBORDINATE ROUTINES:
1688
1689 ; CHKAMB CHECK FOR AMBIGUOUS CONTENTS
1690
1691
1692
1693 PRITSSR:
1694 SAVREG ;SAVE GENERAL REGISTERS
1695 MOV R1,R4 ;SAVE THE TSSR CONTENTS
1696 PRINTB @TSSRFOR,R4 ;PRINT THE CONTENTS OF TSSR
1697 MOV R4,-(SP)
1698 MOV @TSSRFOR,-(SP)
1699 MOV @2,-(SP)
1700 MOV SP,R0
1701 TRAP C:PNTB
1702 ADD @6,SP
1703 MOV R4,R0 ;GET TSSR BACK FOR CHKAMB
1704 JSR PC,CHKAMB ;ARE CONTENTS AMBIGUOUS ?
1705 BCS S# ;BRANCH IF NOT
1706 PRINTX @AMBTSSR ;SHOW CONTENTS ARE AMBIGUOUS
1707 MOV @AMBTSSR,-(SP)
1708 MOV @1,-(SP)
1709 MOV SP,R0
1710 TRAP C:PNTX
1711 ADD @4,SP
1712 5#: MOV R4,R3 ;CONTENTS OF TSSR
1713 BIC @HIADDR!FATERR!TERCLS,R3 ;CLEAR ALL MULTIPLE BIT FIELDS
1714 BEQ 20# ;NO BITS ARE SET
1715 MOV @TMPBFR,R2 ;TEMPORARY ASCII BU-FER
1716 MOV @TSSRBIT,R1 ;ASCII EQUIVALENT OF BITS
1717 10#: TST R3 ;REMAINING BITS TO CONVERT
1718 BEQ 15# ;BRANCH WHEN ALL ARE DONE
1719 CLC ;CLEAR CARRY FOR SHIFT
1720 ROL R3 ;SHIFT NEXT BIT TO CARRY
1721 BCC 13# ;BRANCH IF BIT NOT SET
1722 MOV (R1),R0 ;POINTER TO BIT DEFINITION
1723 11#: MOVB (R0)+,(R2)+ ;MOVE ASCII TO BUFFER
1724 BNE 11# ;MOVE ALL BITS
1725 MOVB @'-1(R2) ;INSERT A COMMA TO TERMINATE
1726 13#: TST (R1)+ ;POINT TO NEXT DESCRIPTION
1727 BR 10# ;GET THE REMAINING BITS
1728 15#: CLRB -(R2) ;TERMINATE THE LINE
1729 PRINTX @TSSDEF,@TMPBFR ;PRINT THE BIT DEFINITIONS
1730 MOV @TMPBFR,-(SP)
1731 MOV @TSSDEF,-(SP)

```

PRITSSR - PRINT TSSR CONTENTS

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

PRITSSR - PRINT TSSR CONTENTS

1744	006360				EPRT2:		
1745	006360	045	116	045	EPRT1:	.ASCIZ	'#N#A *****REPLACE M7196*****'
1746							
1756	006415	045	116	045	TSSRFOR:	.ASCIZ	'#N#A TSSR = #06'
1757	006435	045	116	045	TEXASC:	.ASCIZ	'#N#A Extended Address Bits = #06'
1758	006476	045	116	045	TCOASC:	.ASCIZ	'#N#A Termination Class Code = #T'
1759	006537	045	116	045	TFCASC:	.ASCIZ	'#N#A Fatal Termination Class Code = #T'
1760	006606	045	116	045	TSSDEF:	.ASCIZ	'#N#A TSSR Bits Set: #T'
1761	006635	045	116	045	AMBTSSR:	.ASCIZ	'#N#A TSSR Contents Are Ambiguous'
1762						.EVEN	
1763	006676	006716	006741	006767	TCOCOD:	.WORD	1#,2#,3#,4#,5#,6#,7#,8#
1764	006716	116	157	162	1#:	.ASCIZ	'Normal Termination'
1765	006741	124	145	162	2#:	.ASCIZ	'Termination Condition'
1766	006767	124	141	160	3#:	.ASCIZ	'Tape Status Alert'
1767	007011	106	165	156	4#:	.ASCIZ	'Function Reject'
1768	007031	122	145	143	5#:	.ASCIZ	'Recoverable Error - Tape Position One Record Down'
1769	007113	122	145	143	6#:	.ASCIZ	'Recoverable Error - Tape Was Not Moved'
1770	007162	125	156	162	7#:	.ASCIZ	'Unrecoverable Error'
1771	007206	106	141	164	8#:	.ASCIZ	'Fatal Controller Error'
1772						.EVEN	
1773							
1774	007236	007246	007302	007313	TSFCOD:	.WORD	1#,2#,3#,4#
1775	007246	111	156	164	1#:	.ASCIZ	'Internal Diagnostic Failure'
1776	007302	122	145	163	2#:	.ASCIZ	'Reserved'
1777	007313	102	165	163	3#:	.ASCIZ	'Bus Interface or Sanity Check Error'
1778	007357	122	145	163	4#:	.ASCIZ	'Reserved'
1779						.EVEN	

PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

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

```

1781                                     .SBTTL PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET
1782
1783                                     ;*
1784                                     ;THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
1785                                     ;THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
1786
1787                                     ;INPUT:
1788
1789                                     ;      R0      NUMBER OF WORDS IN PACKET
1790                                     ;      R3      HIGH ORDER COMMAND PACKET ADDRESS
1791                                     ;      R4      ADDRESS OF COMMAND PACKET
1792
1793                                     ;      NOTE:   R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
1794                                     ;
1795
1796 PRIPKT::
1797     SAVREG                                ;SAVE THE REGISTERS
1798     MOV      R0,R5                        ;SAVE NO. OF WORDS IN PACKET
1799     TST      KTENABLE                     ;ABOVE 28K UNDER TEST?
1800     BNE      10$                          ;BR IF YES
1801     CLR      R3                           ;SET HIGH ORDER ADDRESS TO 0
1802     MOV      R3,R1                        ;COPY HIGH ORDER ADDRESS
1803     MOV      R4,R0                        ;GET LOWER ADDRESS
1804     ROL      R0                           ;SHIFT BIT 15 INTO C BIT
1805     ROL      R1                           ;AND INTO HIGH ORDER.
1806     PRINTB   #PKTADD,R1,R4               ;PRINT PACKET ADDRESS
1807     MOV      R4,-(SP)
1808     MOV      R1,-(SP)
1809     MOV      #PKTADD,-(SP)
1810     MOV      #3,-(SP)
1811     MOV      SP,R0
1812     TRAP     C#PNTB
1813     ADD      #10,SP
1814     MOV      R3,R0                        ;GET HIGH ORDER ADDRESS
1815     BEQ      20$                          ;BR IF NOT ABOVE 28K.
1816     MOV      R4,R1                        ;GET LOW ORDER ADDRESS
1817     JSR      PC,SETMAP                    ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
1818     MOV      R0,R4                        ;GET RETURNED PAR6 ADDRESS BIAS
1819     CLR      R1                           ;SAVE WORD NUMBER
1820     MOV      (R4)+,R2                     ;GET PACKET CONTENTS
1821     PRINTB   #PKTFRM,R1,R2               ;PRINT THE DATA
1822     MOV      R2,-(SP)
1823     MOV      R1,-(SP)
1824     MOV      #PKTFRM,-(SP)
1825     MOV      #3,-(SP)
1826     MOV      SP,R0
1827     TRAP     C#PNTB
1828     ADD      #10,SP
1829     INC      R1                           ;NEXT WORD NUMBER
1830     CMP      R1,R5                        ;DONE ALL PACKET WORDS?
1831     BLT      25$                          ;LOOP TILL ALL DONE
1832     RTS      PC                           ;RETURN

```

1820 007516 045 116 045 PKTFRM: .ASCIZ 'N#A Packet Word #D1#A = #06'
1821 007554 045 116 045 PKTADD: .ASCIZ 'N#A Packet Address = #01#05'
1822

PRIBXOR - PRINT EXPD, RECV AND XOR BYTE

```

1824 .SBTTL PRIBXOR - PRINT EXPD, RECV AND XOR BYTE
1825
1826 ;*
1827 ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
1828 ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
1829
1830 ;INPUTS:
1831 ;
1832 ; R1 RECEIVED DATA
1833 ; R2 EXPECTED DATA
1834
1835 ;OUTPUT:
1836 ;
1837 ; R0 XOR OF EXPECTED/RECEIVED DATA
1838
1839 PRIBXOR::
1840 SAVREG ;SAVE THE REGISTERS
1841 MOV R2,R3 ;EXPECTED DATA
1842 XOR R1,R3 ;FORM THE EXCLUSIVE OR
1843 MOV #C<377>,R0 ;BYTE MASK
1844 BIC R0,R1 ;SAVE LOW BYTE RECV
1845 BIC R0,R2 ;SAVE LOW BYTE EXPD
1846 BIC R0,R3 ;SAVE LOW BYTE XOR
1847 PRINTB #XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
1848 MOV R3,-(SP)
1849 MOV R1,-(SP)
1850 MOV R2,-(SP)
1851 MOV #XORBFOR,-(SP)
1852 MOV #4,-(SP)
1853 MOV SP,R0
1854 TRAP C#PNTB
1855 ADD #12,SP
1856 MOV R3,R0 ;R0 HAS XOR ON RETURN
1857 RTS PC ;RETURN TO CALLER
1858
1859 007612
1860 007612 010203
1861 007616 010203 177400
1862 007620
1863 007630 012700 177400
1864 007634 040001
1865 007636 040002
1866 007640 040003
1867 007642
1868 007642 010346
1869 007644 010146
1870 007646 010246
1871 007650 012746 007674
1872 007654 012746 000004
1873 007660 010600
1874 007662 104414
1875 007664 062706 000012
1876 007670 010300
1877 007672 000207
1878
1879 007674 045 116 045 XORBFOR: .ASCIZ 'N#A EXPD: #03#A RECV: #03#A XOR: #03'
1880
1881 .EVEN
1882 .SBTTL PRIBXOR - PRINT EXPD, RECV AND XOR
1883
1884 ;*
1885 ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
1886 ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
1887
1888 ;INPUTS:
1889 ;
1890 ; R1 RECEIVED DATA
1891 ; R2 EXPECTED DATA
1892
1893 ;OUTPUT:
1894 ;
1895 ; R0 XOR OF EXPECTED/RECEIVED DATA
1896
1897 PRIBXOR::
1898 SAVREG ;SAVE THE REGISTERS
1899 MOV R2,R3 ;EXPECTED DATA
1900 XOR R1,R3 ;FORM THE EXCLUSIVE OR
1901 PRINTB #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE

```

PRI XOR - PRINT EXPD, RECV AND XOR

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

PRIQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT

```

1879 .SBTTL PRIQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT
1880
1881 ;*
1882 ;
1883 ;ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
1884 ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
1885 ;
1886 ;INPUTS:
1887 ;
1888 ; R0 OCTAL VALUE TO CONVERT
1889 ; R1 TABLE OF POINTERS TO ASCII EQUIVALENT
1890 ;
1891 ;-
1892
1893 010060 PRIQU: SAVREG ;SAVE THE REGISTERS
1894 010060 RTS PC ;RETURN TO CALLER
1895 010064 000207
1896
1897 .SBTTL PRIRAM - PRINT RAM ADDRESS
1898
1899 ;*
1900 ;
1901 ;PRINT CONTROLLER RAM ADDRESS.
1902 ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
1903 ;
1904 ;INPUTS:
1905 ;
1906 ; R4 RAM ADDRESS
1907 ;-
1908 010066 PRIRAM: SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1909 010066 PRINTB #RAMFOR,R4 ;PRINT RAM ADDRESS IN ERROR
1910 010072 MOV R4,-(SP)
1911 010072 010446 MOV #RAMFOR,-(SP)
1912 010074 012746 010116 MOV #2,-(SP)
1913 010100 012746 000002 MOV SP,R0
1914 010104 010600 TRAP C#PNTB
1915 010106 104414 ADD #6,SP
1916 010110 062706 000006 RTS PC ;RETURN
1917 010114 000207
1918
1919 010116 045 116 045 RAMFOR: .ASCIZ 'N#A CONTROLLER RAM ADDRESS = #06'
1920 .EVEN

```

PRIADD - PRINT MEMORY ERROR ADDRESS

```

1916          .SBTTL PRIADD - PRINT MEMORY ERROR ADDRESS
1917          ;+
1918          ;
1919          ;PRINT MEMORY ADDRESS
1920          ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
1921          ;
1922          ; IMPLICIT INPUTS
1923          ;
1924          ;     ERRHI   - HIGH ORDER ADDRESS
1925          ;     ERRLO   - LOW ORDER ADDRESS
1926          ;
1927          ;-
1928          PRIADD:
1929          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
1930          MOV     ERRHI,R0      ;GET HIGH ADDRESS
1931          MOV     ERRLO,R1      ;GET LOW ADDRESS
1932          MOV     R1,R2        ;COPY LOW ADDRESS
1933          ROL     R1            ;SHIFT BIT 15 TO C BIT
1934          ROL     R0            ;SHIFT INTO HIGH ORDER
1935          PRINTB  #PRIA0,R0,R2 ;PRINT MEMORY ADDRESS IN ERROR
          MOV     R2,-(SP)
          MOV     R0,-(SP)
          MOV     #PRIA0,-(SP)
          MOV     #3,-(SP)
          MOV     SP,R0
          TRAP    C#PNTB
          ADD     #10,SP
          RTS     PC              ;RETURN

```

```

1936          010226 000207
1937
1938          010230 045 116 045 PRIA0: .ASCIZ 'MEMORY ERROR ADDRESS = #01#05'
1939          .EVEN
1940

```

.SBTTL PRITADD - PRINT MEMORY TEST ADDRESS

```

1941          ;+
1942          ;
1943          ;PRINT MEMORY ADDRESS
1944          ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
1945          ;
1946          ; IMPLICIT INPUTS
1947          ;
1948          ;     ERRHI   - HIGH ORDER ADDRESS
1949          ;     ERRLO   - LOW ORDER ADDRESS
1950          ;
1951          ;-
1952          PRITADD:
1953          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
1954          MOV     ERRHI,R2      ;GET HIGH ADDRESS
1955          MOV     ERRLO,R1      ;GET LOW ADDRESS
1956          MOV     R1,R2        ;COPY LOW ADDRESS
1957          ROL     R1            ;SHIFT BIT 15 TO C BIT
1958          ROL     R0            ;SHIFT INTO HIGH ORDER
1959          PRINTB  #PRITO,R1     ;PRINT MEMORY ADDRESS LOW IN ERROR
1960          MOV     R1,-(SP)
          MOV     #PRITO,-(SP)
          MOV     #2,-(SP)
          MOV     SP,R0
          TRAP    C#PNTB

```

PRITADD - PRINT MEMORY TEST ADDRESS

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

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

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

```

1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003 010466
2004 010466
2005 010472 012737 000764 010650
2006 010500 012737 140010 010650
2007 010506 005703
2008 010510 100403
2009 010512 010337 010652
2010 010516 000407
2011 010520 042703 100000
2012 010524 010337 010652
2013 010530 052737 000400 010650
2014 010536 012704 010650
2015 010542 010465 000000
2016 010546 004737 016250
2017 010552 103420
2018 010554
      010554 012727 000250
      010560 000000
      010562 013727 002116
      010566 000000
      010570 005367 177772
      010574 001375

      .ROUTINE TO ISSUE A SPACE RECORDS
      .COMMAND (FORWARD OR REVERSE)
      .INPUT:
      .      R3      NUMBER OF RECORDS TO BE SPACED OVER
      .              BIT15 CONTROLS DIRECTION
      .              BIT15 = 0 IS FORWARD
      .              BIT15 = 1 IS REVERSE
      .      R5      FIRST DEVICE UNIBUS ADDRESS
      .
      .      REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
      .OUTPUT:
      .      CARRY   SET - SPACE RECORDS COMMAND OK
      .              CLR - SPACE RECORDS FAILED
      .
      .      R0      THE CONTENTS OF R4 IS MOVED TO R0
      .
      .IMPLICIT OUTPUT:
      .      TAPE HAS BEEN MOVED
      .SIDE EFFECTS:
      .
      .-
SPACE::
      SAVREG
      MOV      #500.,SDELAY
      MOV      #140010.80#
      TST      R3
      BMI      5#
      MOV      R3,90#
      BR       10#
5#:      BIC      #BIT15,R3
      MOV      R3,90#
      BIS      #BIT8,80#
10#:     MOV      #80#,R4
      MOV      R4,TSDB(R5)
15#:     JSR      PC,WAITF
      BCS      20#
      DELAY     250
      MOV      #250.(PC),
      .WORD    0
      MOV      L#DLY,(PC),
      .WORD    0
      DEC      -6(PC)
      BNE      .-4

      ;SAVE THE GENERAL REGISTERS
      ;SET UP DELAY
      ;SET UP COMMAND, SPACE FORWARD
      ;CHECK FOR DIRECTION
      ;BR, IF REVERSE INDICATED
      ;LOAD UP NUMBER OF RECORDS TO SPACE
      ;GO DO COMMAND
      ;CLEAR DIRECTION BIT
      ;LOAD UP NUMBER OF RECORDS TO SPACE
      ;SET REVERSE BIT IN COMMAND PACKET
      ;SET UP R4 WITH PACKET ADDRESS
      ;SEND OUT COMMAND
      ;WAIT FOR SSR
      ;BR, IF SSR IS SET AND OK
      ;DELAY ABOUT .25 SECONDS

```

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

010576	005367	177756	DEC	-22(PC)	
010602	001367		BNE	.-20	
2019 010604	005337	010660	DEC	SDELAY	;BUMP DELAY COUNTER DOWN
2020 010610	001356		BNE	15	;BR, IF MORE DELAY
2021 010612	000411		BR	60	;BR IF TROUBLE CARRY = CLEAR
2022 010614	016501	000002	204: MOV	TSSR(R5),R1	;READ TSSR
2023 010620	012702	000200	MOV	#SSR,R2	;SET UP EXPECTED
2024 010624	020201		254: CMP	R2,R1	;ARE THEY OK
2025 010626	001401		BEQ	40	;BR, IF EQUAL = OK
2026 010630	000402		BR	60	;TROUBLE EXIT
2027 010632	000261		404: SEC		;SET CARRY NO TROUBLE
2028 010634	000401		BR	70	;EXIT
2029 010636	000241		604: CLC		;CARRY CLEAR = ERROR
2030 010640			704:		
2031 010640	010400		MOV	R4,R0	;PASS PACKET ADDRESS
2032 010642	000207		RTS	PC	;RETURN
2033					
2034					
2035					
2036					
2037					
2039	010650				
2041					
2042					
2043 010650	000000				
2044					
2045 010652	000000				
2046 010654	000000				
2047 010656	000000				
2048 010660	000000				
2049					
2050					

; PACKET FOR SPACE COMMAND

.-<.10>E177770

; COMMAND WORD

804: .WORD

; NUMBER OF RECORDS TO BE SPACED OVER WORD

904: .WORD

.WORD

.WORD

SDELAY: .WORD

0

; DELAY COUNTER

.EVEN

.SBTTL

WRTCHR - WRITE CHARACTERISTICS COMMAND

WRTCHR - WRITE CHARACTERISTICS COMMAND

```

2052
2053 ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
2054 ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
2055
2056 ;INPUT:
2057 ;       R4      ADDRESS OF PACKET FROM TEST
2058 ;       R5      FIRST DEVICE UNIBUS ADDRESS
2059 ;       REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
2060
2061 ;OUTPUT:
2062 ;       R0      TSSR CONTENTS
2063 ;       CARRY   SET - WRITE CHARACTERISTICS COMMAND OK
2064 ;              CLR - WRITE CHARACTERISTICS FAILED
2065
2066 ;IMPLICIT OUTPUT:
2067 ;
2068 ;       MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
2069 ;       SOFTWARE SWITCHES SET AS FOLLOWS:
2070 ;       EXTFEA = EXTENDED FEATURES PRESENT
2071 ;       BENBSW = BUFFER ENABLE SWITCH ON OR OFF
2072
2073 ;SIDE EFFECTS:
2074 ;
2075 WRTCHR::
2076 ;SAVREG
2077 ;CLR      BENBSW      ;SAVE THE GENERAL REGISTERS
2078 ;CLR      EXTFEA      ;CLEAR BUFFER ENABLE SWITCH
2079 ;MOV      R4,TSDB(R5) ;CLEAR EXTENDED FEATURES SW SWITCH
2080 ;JSR      PC,CHKTSSR  ;SEND OUT COMMAND
2081 ;BCS      20$         ;WAIT FOR SSR
2082 ;BR       60$         ;BR, IF SSR IS SET AND OK
2083 ;MOV      TSSR(R5),R1 ;BR IF TROUBLE CARRY = CLEAR
2084 ;MOV      #SSR,R2     ;READ TSSR
2085 ;BIT      #OFL,R1     ;SET UP EXPECTED
2086 ;BEQ      25$         ;WAS OFF LINE SET IN TSSR
2087 ;BIS      #OFL,R2     ;BR, IF NO OFL SET
2088 ;CMP      R2,R1       ;MAKE THEM LOOK ALIKE
2089 ;BEQ      40$         ;ARE THEY OK
2090 ;BR       60$         ;BR, IF EQUAL = OK
2091 ;ADD      #8,R4        ;TROUBLE EXIT
2092 ;MOV      (R4),R3      ;POINT TO WRT CHARA DATA PACKET
2093 ;BIT      #X2.EXTF,XST2(R3) ;GET ADDRESS OF MESSAGE BUFFER
2094 ;BEQ      45$         ;EXTENDED FEATURES BIT SET?
2095 ;INC      EXTFEA       ;BR IF NO
2096 ;          ;SET EXTENDED FEATURES SW SWITCH
2097 ;BIT      #X2.BUFE,XST2(R3) ;BUFFER ENABLE SWITCH SET
2098 ;BEQ      50$         ;BR, IF SWITCH NOT SET
2099 ;INC      BENBSW       ;SET SOFTWARE SWITCH FOR ENABLED
2100 ;
2101 ;SEC
2102 ;BR       70$         ;SET CARRY NO TROUBLE
2103 ;CLC
2104 ;MOV      TSSR(R5),R0 ;EXIT
2105 ;RTS      PC          ;CARRY CLEAR = ERROR
                        ;RETURN TSSR CONTENTS
                        ;RETURN

```

REWIND - POSITION TAPE (REWIND) COMMAND

```

2107 .SBTTL REWIND - POSITION TAPE (REWIND) COMMAND
2108
2109 ;*
2110 ;THIS ROUTINE WILL REWIND THE SELECTED TAPE.
2111
2112 ; CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
2113 ; TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
2114 ; SSR TO SET IN THE TSSR
2115
2116 ;CALLING SEQUENCE:
2117 ;
2118 ; DO A SOFT INIT
2119 ; DO A WRITE CHARACTERISTICS
2120 ; JSR PC,REWIND
2121
2122 ;INPUT:
2123 ;
2124 ; R5 FIRST DEVICE UNIBUS ADDRESS
2125
2126 ;OUTPUT
2127 ;
2128 ; R0 THE CONTENTS OF R4 IS PASSED TO R0
2129
2130 ;
2131 ;
2132 ;
2133 ;-
2134 REWIND::
2135 SAVREG
2136 MOV #RWPACK,R4 ;SAVE R1-R5 UNTIL NEXT RETURN
2137 MOV R4,TSDB(R5) ;GET PACKET ADDRESS
2138 MOV #360,R3 ;SEND PACKET ADDRESS TO EXECUTE
2139 JSR PC,WAITF ;ENOUGH TIME FOR 2400' REEL TO REWIND
2140 BCS 20$ ;WAIT FOR SSR TO SET
2141 DELAY 250. ;LEAVE WHEN SSR IS SET
2142 MOV #250.,(PC). ;WAIT FOR .25 SECONDS
2143 .WORD 0
2144 MOV L$DL,(PC).
2145 .WORD 0
2146 DEC -6(PC)
2147 BNE -4
2148 DEC -22(PC)
2149 BNE -20
2150 DEC R3 ;BUMP COUNTER DOWN
2151 BNE 10$ ;KEEP GOING
2152 CLC ;CLEAR CARRY TO SET ERROR
2153 MOV R4,R0 ;PASS THE PACKET ADDRESS
2154 RTS PC ;RETURN
2155
2156 RWPACK: .=<..10>E177770
2157 .WORD 102010 ;POSITION COMMAND (REWIND)
2158 .WORD 0 ;NOT USED

```

CKRAM - COMPARE RAM TO I/O PACKET

```

2155 .SBTTL CKRAM - COMPARE RAM TO I/O PACKET
2156
2157 ;*
2158 ;ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
2159 ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
2160
2161 ;INPUT:
2162
2163 ; R4 ADDRESS OF THE COMMAND PACKET
2164 ; R5 FIRST DEVICE UNIBUS ADDRESS
2165
2166 ;OUTPUT:
2167
2168 ; CARRY SET - RAM MATCHES PACKET
2169 ; CLR - RAM DOES NOT MATCH PACKET
2170
2171 ;IMPLICIT OUTPUT:
2172
2173 ; THE TABLE RAMDATA IS FILLED WITH THE
2174 ; DATA HELD IN RAM.
2175 ; RAMSIZ IS SET TO 8. FOR PRAMPKT ROUTINE
2176
2177 ;SIDE EFFECTS:
2178
2179 ; THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
2180
2181 ;-
2182
2183 CKRAM:: SAVREG ;SAVE THE GENERAL REGISTERS
2184 MOV #RAMDATA,R1 ;ADDRESS TO SAVE THE RAM DATA
2185 MOV #RMPKT8EG,R2 ;BYTE ADDRESS OF FIRST RAM DATA
2186 CLR R3 ;CLEAR THE ERROR FLAG
2187 JSR PC,CHKTSSR ;WAIT FOR SSR
2188 MOV #0,TSDB(R5) ;SET MAINTENANCE MODE
2189 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
2190 MOV R2,TSDB(R5) ;SELECT NEXT RAM ADDRESS
2191 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
2192 MOV TSBA(R5),(R1) ;READ THE RAM DATA
2193 CMPB (R1),.(R4) ;COMPARE TO EXPECTED
2194 BEQ 20$ ;BRANCH IF OK
2195 INC R3 ;SET ERROR FLAG
2196 INC R2 ;ADDRESS OF NEXT RAM LOCATION
2197 CMP R2,#RMPKTEND ;REACHED END YET ?
2198 BLE 10$ ;BRANCH TILL ALL READ
2199 TST R3 ;WAS AN ERROR FOUND ?
2200 BEQ 30$ ;BRANCH IF NOT
2201 CLC ;CLEAR CARRY TO SHOW ERROR
2202 BR 50$ ;AND EXIT
2203 SEC ;SHOW GOOD COMPARE
2204 MOV #8.,RAMSIZ ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
2205 RTS PC ;RETURN
2206

```

2183	011114						
2184	011114						
2185	011120	012701	002242				
2186	011124	012702	000201				
2187	011130	005003					
2188	011132	004737	016336				
2189	011136	112765	000000	000000			
2190	011144	004737	016336		10\$:		
2191	011150	010265	000000				
2192	011154	004737	016336				
2193	011160	116511	000000				
2194	011164	122124					
2195	011166	001401					
2196	011170	005203					
2197	011172	005202			20\$:		
2198	011174	020227	000210				
2199	011200	003761					
2200	011202	005703					
2201	011204	001402					
2202	011206	000241					
2203	011210	000401					
2204	011212	000261			30\$:		
2205	011214	012737	000010	002302	50\$:		
2206	011222	000207					

CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA

```

2208 .SBTTL CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA
2209 ;*
2210 ;
2211 ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
2212 ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
2213 ;
2214 ;INPUT:
2215 ;
2216 ;      R4      ADDRESS OF THE CHARACTERISTICS DATA
2217 ;      R5      FIRST DEVICE UNIBUS ADDRESS
2218 ;
2219 ;OUTPUT:
2220 ;
2221 ;      CARRY   SET - RAM MATCHES PACKET
2222 ;             CLR - RAM DOES NOT MATCH PACKET
2223 ;
2224 ;IMPLICIT OUTPUT:
2225 ;
2226 ;      THE TABLE RAMDATA IS FILLED WITH THE
2227 ;      DATA HELD IN RAM.
2228 ;      RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
2229 ;
2230 ;SIDE EFFECTS:
2231 ;
2232 ;      THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
2233 ;
2234 CKRAM2::
2235     SAVREG                                ;SAVE THE GENERAL REGISTERS
2236     MOV     #RAMDATA,R1                  ;ADDRESS TO SAVE THE RAM DATA
2237     MOV     #RMCHBEG,R2                  ;BYTE ADDRESS OF FIRST RAM DATA
2238     CLR     R3                           ;CLEAR THE ERROR FLAG
2239     JSR     PC,CHKTSSR                   ;WAIT FOR SSR
2240     MOVB    #0,TSDB(R5)                  ;SET MAINTENANCE MODE
2241     JSR     PC,CHKTSSR                   ;WAIT FOR SSR TO SET
2242     MOV     R2,TSDB(R5)                  ;SELECT NEXT RAM ADDRESS
2243     JSR     PC,CHKTSSR                   ;WAIT FOR SSR TO SET
2244     MOVB    TSBA(R5),(R1)                ;READ THE RAM DATA
2245     CMPB    (R1)+,(R4)+                  ;COMPARE TO EXPECTED
2246     BEQ     20#                          ;BRANCH IF OK
2247     INC     R3                           ;SET ERROR FLAG
2248     INC     R2                           ;ADDRESS OF NEXT RAM LOCATION
2249     MOV     #8,,RAMSIZ                   ;ASSUME EXTFEA NOT SET
2250     TST     EXTFEA                       ;IS THE SOFTWARE EXTENDED FEATURES SET
2251     BEQ     25#                          ;BR, IF NOT SET
2252     MOV     #10,,RAMSIZ                  ;SET RAMSIZ FOR EXTEND FEATURES
2253     CMP     R2,#RMCHEND                   ;AT END OF EXTENDED BUFFER
2254     BLE     10#                          ;BR, IF NOT AT END YET
2255     BR      27#                          ;AT END BRANCH
2256     CMP     R2,#RMCHEND-2                 ;REACHED END YET ?
2257     BLE     10#                          ;BRANCH TILL ALL READ
2258     TST     R3                           ;WAS AN ERROR FOUND ?
2259     BEQ     30#                          ;BRANCH IF NOT
2260     CLC                                     ;CLEAR CARRY TO SHOW ERROR
2261     BR      50#                          ;AND EXIT
2262     SEC                                     ;SHOW GOOD COMPARE
2263     RTS     PC                           ;RETURN

```

CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS

```

2265          .SBTTL CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS
2266          ;*
2267          ;
2268          ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
2269          ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
2270          ;ERROR PRINT ROUTINES.
2271          ;
2272          ;INPUT:
2273          ;
2274          ;      R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
2275          ;      R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
2276          ;      R2      EXPD MESSAGE BUFFER ADDRESS
2277          ;OUTPUT:
2278          ;
2279          ;      CARRY   SET - MESSAGE BUFFERS MATCH
2280          ;              CLR -MESSAGE BUFFERS DON'T MATCH
2281          ;
2282          ;IMPLICIT OUTPUT:
2283          ;
2284          ;      EXPMSG   BUFFER IS SET TO EXPD DATA
2285          ;      RECVMSG  BUFFER IS SET TO RECV DATA
2286          ;      RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
2287          ;      RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
2288          ;
2289          ;-
2290          CKMSG::
2291          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
2292          MOV             R0,RCVHIADD      ;SAVE RECV HIGH ADDRESS
2293          MOV             R1,RCVLOAD      ;SAVE RECV LOW ADDRESS
2294          TST             KTENABLE        ;TESTING ABOVE 28K?
2295          BEQ             10$            ;BR IF NO
2296          JSR             PC,SETMAP      ;RETURN ADDRESS BIASED TO PAR6 IN R0
2297          MOV             R0,R1          ;GET RETURNED ADDRESS BIASED TO PAR6
2298          10$: CLR        R4              ;WORD IN BUFFER
2299          CLR             R3              ;CLEAR ERROR SEEN FLAG
2300          MOV             R2,R5          ;GET EXPD BUFFER ADDRESS
2301          15$: MOV        (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
2302          MOV             (R1),RECVMSG(R4) ;SAVE RECV FOR ERROR REPORT
2303          CMP             (R2),.(R1)     ;EXPD EQUAL RECV?
2304          BEQ             25$            ;BR IF YES
2305          INC             R3              ;SET ERROR SEEN FLAG
2306          25$: ADD        #2,R4          ;POINT TO NEXT WORD ADDRESS
2307          CMP             R4,#14         ;DONE FIRST 7 WORDS?
2308          BLE             15$            ;BR IF NO
2309          BIT             #X2.EXTF,XST2(R5) ;IS EXTENDED FEATURES SET IN EXPD?
2310          BEQ             50$            ;BR IF NO
2311          CMP             R4,#16         ;DONE EXTENDED FEATURES WORD?
2312          BLE             15$            ;BR IF NO
2313          50$: TST        R3              ;ANY ERRORS SEEN?
2314          BEQ             55$            ;BR IF NO
2315          CLC              ;SET FAILURE
2316          BR             60$            ;
2317          55$: SEC          ;SET SUCCESS
2318          60$: RTS          PC           ;RETURN

```

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

2320 .SBTTL CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS
2321
2322 ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
2323 ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
2324 ;ERROR PRINT ROUTINES.
2325
2326 ;INPUT:
2327 ;      R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
2328 ;      R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
2329 ;      R2      EXPD MESSAGE BUFFER ADDRESS
2330 ;      R3      NUMBER OF BYTES TO COMPARE
2331
2332 ;OUTPUT:
2333 ;      CARRY    SET - MESSAGE BUFFERS MATCH
2334 ;              CLR - MESSAGE BUFFERS DON'T MATCH
2335
2336 ;IMPLICIT OUTPUT:
2337 ;      EXPMSG    BUFFER IS SET TO EXPD DATA
2338 ;      RECVMSG    BUFFER IS SET TO RECV DATA
2339 ;      RCVHIADD   SET TO HIGH ORDER ADDRESS OF RECV
2340 ;      RCVLOAD    SET TO LOW ORDER ADDRESS OF RECV
2341
2342 CKMSG2::
2343     SAVREG                      ;SAVE R1-R5 UNTIL NEXT RETURN
2344     CMP      R3,#RECVMSG-EXPMSG;000 IS COUNT ABOVE MAX ALLOWED?
2345     BLE      5#                 ;000 BR IF NO
2346     MOV      #RECVMSG-EXPMSG,R3;000
2347     PRINTF   #DEBUGMSG          ;000
2348     MOV      #DEBUGMSG,-(SP)
2349     MOV      #1,-(SP)
2350     MOV      SP,R0
2351     TRAP     C#PNTF
2352     ADD      #4,SP
2353     MOV      R0,RCVHIADD        ;SAVE RECV HIGH ADDRESS
2354     MOV      R1,RCVLOAD         ;SAVE RECV LOW ADDRESS
2355     TST      KTENABLE           ;TESTING ABOVE 28K?
2356     BEQ      10#               ;BR IF NO
2357     JSR      PC,SETMAP         ;RETURN ADDRESS BIASED TO PAR6 IN R0
2358     MOV      R0,R1             ;GET RETURNED ADDRESS BIASED TO PAR6
2359     CLR      R4                ;WORD IN BUFFER
2360     CLR      R5                ;CLEAR ERROR SEEN FLAG
2361     MOV      (R2),EXPMSG(R4)    ;SAVE EXPD FOR ERROR REPORT
2362     MOV      (R1),RECVMSG(R4)  ;SAVE RECV FOR ERROR REPORT
2363     CMP      (R2)+,(R1)+       ;EXPD EQUAL RECV?
2364     BEQ      25#               ;BR IF YES
2365     INC      R5                ;SET ERROR SEEN FLAG
2366     ADD      #1,R4             ;POINT TO NEXT BYTE
2367     CMP      R4,R3             ;DONE ALL BYTES?
2368     BGE      50#               ;BR IF YES
2369     BR       15#               ;DO NEXT BYTE
2370     TST      R5                ;ANY ERRORS SEEN?
2371     BEQ      55#               ;BR IF NO
2372     CLC
2373     BR       60#               ;SET FAILURE
2374     SEC
2375     PC
2376     RTS
2377
2378     5#:  MOV      R0,RCVHIADD        ;SAVE RECV HIGH ADDRESS
2379           MOV      R1,RCVLOAD         ;SAVE RECV LOW ADDRESS
2380           TST      KTENABLE           ;TESTING ABOVE 28K?
2381           BEQ      10#               ;BR IF NO
2382           JSR      PC,SETMAP         ;RETURN ADDRESS BIASED TO PAR6 IN R0
2383           MOV      R0,R1             ;GET RETURNED ADDRESS BIASED TO PAR6
2384           CLR      R4                ;WORD IN BUFFER
2385           CLR      R5                ;CLEAR ERROR SEEN FLAG
2386           MOV      (R2),EXPMSG(R4)    ;SAVE EXPD FOR ERROR REPORT
2387           MOV      (R1),RECVMSG(R4)  ;SAVE RECV FOR ERROR REPORT
2388           CMP      (R2)+,(R1)+       ;EXPD EQUAL RECV?
2389           BEQ      25#               ;BR IF YES
2390           INC      R5                ;SET ERROR SEEN FLAG
2391           ADD      #1,R4             ;POINT TO NEXT BYTE
2392           CMP      R4,R3             ;DONE ALL BYTES?
2393           BGE      50#               ;BR IF YES
2394           BR       15#               ;DO NEXT BYTE
2395           TST      R5                ;ANY ERRORS SEEN?
2396           BEQ      55#               ;BR IF NO
2397           CLC
2398           BR       60#               ;SET FAILURE
2399           SEC
2400           PC
2401           RTS
2402
2403     10#:  MOV      R0,RCVHIADD        ;SAVE RECV HIGH ADDRESS
2404            MOV      R1,RCVLOAD         ;SAVE RECV LOW ADDRESS
2405            TST      KTENABLE           ;TESTING ABOVE 28K?
2406            BEQ      10#               ;BR IF NO
2407            JSR      PC,SETMAP         ;RETURN ADDRESS BIASED TO PAR6 IN R0
2408            MOV      R0,R1             ;GET RETURNED ADDRESS BIASED TO PAR6
2409            CLR      R4                ;WORD IN BUFFER
2410            CLR      R5                ;CLEAR ERROR SEEN FLAG
2411            MOV      (R2),EXPMSG(R4)    ;SAVE EXPD FOR ERROR REPORT
2412            MOV      (R1),RECVMSG(R4)  ;SAVE RECV FOR ERROR REPORT
2413            CMP      (R2)+,(R1)+       ;EXPD EQUAL RECV?
2414            BEQ      25#               ;BR IF YES
2415            INC      R5                ;SET ERROR SEEN FLAG
2416            ADD      #1,R4             ;POINT TO NEXT BYTE
2417            CMP      R4,R3             ;DONE ALL BYTES?
2418            BGE      50#               ;BR IF YES
2419            BR       15#               ;DO NEXT BYTE
2420            TST      R5                ;ANY ERRORS SEEN?
2421            BEQ      55#               ;BR IF NO
2422            CLC
2423            BR       60#               ;SET FAILURE
2424            SEC
2425            PC
2426            RTS
2427
2428     15#:  MOV      R0,RCVHIADD        ;SAVE RECV HIGH ADDRESS
2429            MOV      R1,RCVLOAD         ;SAVE RECV LOW ADDRESS
2430            TST      KTENABLE           ;TESTING ABOVE 28K?
2431            BEQ      10#               ;BR IF NO
2432            JSR      PC,SETMAP         ;RETURN ADDRESS BIASED TO PAR6 IN R0
2433            MOV      R0,R1             ;GET RETURNED ADDRESS BIASED TO PAR6
2434            CLR      R4                ;WORD IN BUFFER
2435            CLR      R5                ;CLEAR ERROR SEEN FLAG
2436            MOV      (R2),EXPMSG(R4)    ;SAVE EXPD FOR ERROR REPORT
2437            MOV      (R1),RECVMSG(R4)  ;SAVE RECV FOR ERROR REPORT
2438            CMP      (R2)+,(R1)+       ;EXPD EQUAL RECV?
2439            BEQ      25#               ;BR IF YES
2440            INC      R5                ;SET ERROR SEEN FLAG
2441            ADD      #1,R4             ;POINT TO NEXT BYTE
2442            CMP      R4,R3             ;DONE ALL BYTES?
2443            BGE      50#               ;BR IF YES
2444            BR       15#               ;DO NEXT BYTE
2445            TST      R5                ;ANY ERRORS SEEN?
2446            BEQ      55#               ;BR IF NO
2447            CLC
2448            BR       60#               ;SET FAILURE
2449            SEC
2450            PC
2451            RTS
2452
2453     25#:  MOV      R0,RCVHIADD        ;SAVE RECV HIGH ADDRESS
2454            MOV      R1,RCVLOAD         ;SAVE RECV LOW ADDRESS
2455            TST      KTENABLE           ;TESTING ABOVE 28K?
2456            BEQ      10#               ;BR IF NO
2457            JSR      PC,SETMAP         ;RETURN ADDRESS BIASED TO PAR6 IN R0
2458            MOV      R0,R1             ;GET RETURNED ADDRESS BIASED TO PAR6
2459            CLR      R4                ;WORD IN BUFFER
2460            CLR      R5                ;CLEAR ERROR SEEN FLAG
2461            MOV      (R2),EXPMSG(R4)    ;SAVE EXPD FOR ERROR REPORT
2462            MOV      (R1),RECVMSG(R4)  ;SAVE RECV FOR ERROR REPORT
2463            CMP      (R2)+,(R1)+       ;EXPD EQUAL RECV?
2464            BEQ      25#               ;BR IF YES
2465            INC      R5                ;SET ERROR SEEN FLAG
2466            ADD      #1,R4             ;POINT TO NEXT BYTE
2467            CMP      R4,R3             ;DONE ALL BYTES?
2468            BGE      50#               ;BR IF YES
2469            BR       15#               ;DO NEXT BYTE
2470            TST      R5                ;ANY ERRORS SEEN?
2471            BEQ      55#               ;BR IF NO
2472            CLC
2473            BR       60#               ;SET FAILURE
2474            SEC
2475            PC
2476            RTS
2477
2478     50#:  MOV      R0,RCVHIADD        ;SAVE RECV HIGH ADDRESS
2479            MOV      R1,RCVLOAD         ;SAVE RECV LOW ADDRESS
2480            TST      KTENABLE           ;TESTING ABOVE 28K?
2481            BEQ      10#               ;BR IF NO
2482            JSR      PC,SETMAP         ;RETURN ADDRESS BIASED TO PAR6 IN R0
2483            MOV      R0,R1             ;GET RETURNED ADDRESS BIASED TO PAR6
2484            CLR      R4                ;WORD IN BUFFER
2485            CLR      R5                ;CLEAR ERROR SEEN FLAG
2486            MOV      (R2),EXPMSG(R4)    ;SAVE EXPD FOR ERROR REPORT
2487            MOV      (R1),RECVMSG(R4)  ;SAVE RECV FOR ERROR REPORT
2488            CMP      (R2)+,(R1)+       ;EXPD EQUAL RECV?
2489            BEQ      25#               ;BR IF YES
2490            INC      R5                ;SET ERROR SEEN FLAG
2491            ADD      #1,R4             ;POINT TO NEXT BYTE
2492            CMP      R4,R3             ;DONE ALL BYTES?
2493            BGE      50#               ;BR IF YES
2494            BR       15#               ;DO NEXT BYTE
2495            TST      R5                ;ANY ERRORS SEEN?
2496            BEQ      55#               ;BR IF NO
2497            CLC
2498            BR       60#               ;SET FAILURE
2499            SEC
2500            PC
2501            RTS
2502
2493     55#:  SEC
2504     60#:  RTS

```

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

2372 011632      120      122      117 DEBUGMSG: .ASCIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-' ;00D
2373 011722      045      116      045 FERCH: .ASCII /NMA ***/
2374 011733      040      040      124 ERCH: .ASCIZ / TSSR ERROR CODE REC'D = /
2375 011766      056      056      056 SIMSG: .ASCIZ /.... AFTER DOING SOFT INIT/
2376 012021      124      105      123 TINERR: .ASCIZ /TEST: .../
2377                                     .EVEN
2378                                     ;+
2379                                     ;
2380                                     ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
2381                                     ;
2382                                     ;INPUT:
2383                                     ;
2384                                     ;      R1      CONTENTS OF TSSR AT ERROR
2385                                     ;
2386                                     ;SIDE EFFECTS:
2387                                     ;
2388                                     ;      EXECUTES DROP UNIT TO CEASE TESTING
2389                                     ;
2390                                     ;-
2391
2392 012034      BGNMSG  SFMSG
2393 012034      SFMSG:: JSR      PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
2394 012040      004737  006024 JSR      PC,CKDROP      ;DROP UNIT, IF ALLOWED
2395 012044      004737  017202 ENDMMSG
2396 012044      104423 L10003: TRAP      C#MSG
2397
2398                                     ;+
2399                                     ;PRINT ROUTINE TO PRINT THE CONTENTS OF
2400                                     ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
2401                                     ;
2402                                     ;INPUTS:
2403                                     ;
2404                                     ;      R1      TSSR CONTENTS
2405                                     ;      R4      ADDRESS OF COMMAND PACKET
2406                                     ;
2407                                     ;-
2408 012046      BGNMSG  PKTSSR
2409 012046      PKTSSR:: JSR      PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
2410 012052      004737  006024 MOV      #4,R0      ;NO. OF WORDS IN PACKET
2411 012056      004737  007370 JSR      PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
2412 012062      004737  007370 ENDMMSG
2413 012062      104423 L10004: TRAP      C#MSG

```

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

2414
2415      ;*
2416      ;PRINT ROUTINE TO PRINT THE CONTENTS OF
2417      ;TSSR AND A GET STATUS COMMAND PACKET.
2418      ;
2419      ;INPUTS:
2420      ;
2421      ;      R1      TSSR CONTENTS
2422      ;      R4      ADDRESS OF COMMAND PACKET
2423      ;-
2423 012064      BGNMSG  PKTGETS
2423 012064      PKTGETS:
2424 012064 004737 006024      JSR      PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
2425 012070 012700 000002      MOV      #2,R0      ;NO. OF WORDS IN GET STATUS PACKET
2426 012074 004737 007370      JSR      PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
2427 012100      ENDMSG
2427 012100
2427 012100 104423      L10005:      TRAP      C#MSG
2428
2429      ;*
2430      ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
2431      ;
2432      ;INPUTS:
2433      ;
2434      ;      R1      TSSR CONTENTS
2435      ;      R4      ADDRESS OF COMMAND PACKET
2436      ;-
2436 012102      BGNMSG  SFFMSG
2436 012102 004737 006024      SFFMSG:
2437 012106      JSR      PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
2437 012106      ENDMSG
2437 012106 104423      L10006:      TRAP      C#MSG
2438      .SBTTL      PKTMES  - PRINT TSSR AND MESSAGE BUFFER
2439
2440      ;*
2441      ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
2442      ;BUFFER FOR ERROR REPORTS
2443      ;
2444      ;INPUTS:
2445      ;
2446      ;      R1      CONTENTS OF TSSR
2447      ;      R2      LOW ORDER MESSAGE BUFFER
2448      ;      R3      HIGH ORDER MESSAGE BUFFER ADDRESS
2449      ;      NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
2450      ;-
2450 012110      BGNMSG  PKTMES
2450 012110      PKTMES:
2451 012110 004737 006024      JSR      PC,PRITSSR      ;PRINT CONTENTS OF TSSR
2452 012114 010200      MOV      R2,R0      ;LOW ORDER ADDRESS
2453 012116 010301      MOV      R3,R1      ;HIGH ORDER ADDRESS
2454 012120 004737 014242      JSR      PC,PRMESS      ;PRINT THE MESSAGE BUFFER
2455 012124      ENDMSG
2455 012124
2455 012124 104423      L10007:      TRAP      C#MSG

```

ADDSSR - PRINT TEST ADDRESS AND TSSR

```

2457          .SBTTL  ADDSSR - PRINT TEST ADDRESS AN' TSSR
2458          ;*
2459          ;PRINT ROUTINE TO PRINT THE CONTENTS OF
2460          ;TSSR AND A MEMORY TEST ADDRESS
2461          ;
2462          ;INPUTS:
2463          ;
2464          ;      R5      FIRST DEVICE UNIBUS ADDRESS
2465          ;      ERRHI   HIGH ORDER MEMORY TEST ADDRESS
2466          ;      ERRLO   LOW ORDER MEMORY TEST ADDRESS
2467          ;
2468          ;
2469          BGNMSG  ADDSSR
012126 ADDSSR::
012126          JSR      PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
2470 012126 004737 010274          MOV      TSSR(R5),R1      ;GET CURRENT TSSR
2471 012132 016501 000002          JSR      PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
2472 012136 004737 006024          ENDMSG
2473 012142
012142 L10010:
012142 104423          TRAP      C#MSG

2474          .SBTTL  MSGEXP - PRINT WRITE CHAR. EXPD-RCV MESSAGE BUFFERS
2475          ;*
2476          ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
2477          ;
2478          ;IMPLICIT INPUTS:
2479          ;
2480          ;      EXPMSG   - EXPECTED MESSAGE BUFFER
2481          ;      RECMMSG  - RECEIVED MESSAGE BUFFER
2482          ;      RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2483          ;      RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2484          ;
2485          ;
2486          ;
2487          BGNMSG  MSGEXP
012144 MSGEXP::
012144          MOV      #7,R0      ;ASSUME NO EXT FEATURES
2488 012144 012700 000007          TST      EXTFEA      ;EXT FEATURES SET?
2489 012150 005737 002226          BEQ      S#          ;BR IF NO
2490 012154 001402          MOV      #8,R0      ;EXT FEATURE BUFFER IS 8 WORDS
2491 012156 012700 000010          JSR      PC,PRMSGEXP      ;PRINT EXPD/RCV MESSAGE BUFFERS
2492 012162 004737 014552          ENDMSG
2493 012166
012166 L10011:
012166 104423          TRAP      C#MSG

```

FIFEXP - PRINT FIFO EXP/RECV DATA

```

2495          .SBTTL FIFEXP - PRINT FIFO EXP/RECV DATA
2496          ;*
2497          ;PRINT ROUTINE TO PRINT FIFO EXP/RECV DATA
2498          ;
2499          ;      R1      - BYTE COUNT
2500          ;
2501          ;IMPLICIT INPUTS:
2502          ;
2503          ;      EXPMSG  - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY
2504          ;      RECMMSG - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
2505          ;
2506          ;-
2507          BGNMSG FIFEXP
2508          FIFEXP::
2509          PRINTX #FIF1MSG,R1      ;PRINT BYTES TRANSFERRED
2510          MOV     R1,-(SP)
2511          MOV     #FIF1MSG,-(SP)
2512          MOV     #2,-(SP)
2513          MOV     SP,R0
2514          TRAP    C#PNTX
2515          ADD     #6,SP
2516          PRINTX #FIF2MSG      ;PRINT HEADER MSG
2517          MOV     #FIF2MSG,-(SP)
2518          MOV     #1,-(SP)
2519          MOV     SP,R0
2520          TRAP    C#PNTX
2521          ADD     #4,SP
2522          MOV     R1,R0          ;GET BYTE COUNT
2523          JSR     PC,PRBYTEXP    ;PRINT FIFO BYTES IN ERROR
2524          ENDMMSG
2525          L10012:
2526          TRAP    C#MSG
2527          .ASCIZ  '#N#A NUMBER OF BYTES TRANSFERRED = #D2'
2528          .ASCIZ  '#N#A FIFO DATA BYTES IN ERROR:'
2529          .EVEN
2530
2531          012170
2532          012170
2533          012170 010146
2534          012172 012746 012242
2535          012176 012746 000002
2536          012202 C10600
2537          012204 104415
2538          012206 062706 000006
2539          012212
2540          012212 012746 012311
2541          012216 012746 000001
2542          012222 010600
2543          012224 104415
2544          012226 062706 000004
2545          012232 010100
2546          012234 004737 015122
2547          012240
2548          012240
2549          012240 104423
2550          012242 045 116 045 FIF1MSG:
2551          012311 045 116 045 FIF2MSG:
2552          2515          .EVEN

```

MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS

```

2517          .SBTTL MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
2518
2519          ;
2520          ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
2521          ;
2522          ;IMPLICIT INPUTS:
2523          ;
2524          ;      EXPMSG - EXPECTED MESSAGE BUFFER
2525          ;      RECHMSG - RECEIVED MESSAGE BUFFER
2526          ;      RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2527          ;      RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2528          ;
2529          ;-
2530          BGNMSG MSGSTAT
2531          MSGSTAT:
2532          10: MOV     #STATCOD,R1      ;ASCII ADDRESS TABLE
2533              MOV     (R1)+,R0        ;DONE ALL MSG LINES?
2534              BEQ     20:              ;BR IF YES
2535              PRINTX   R0              ;PRINT STATUS BIT NAMES
2536              MOV     R0,-(SP)
2537              MOV     #1,-(SP)
2538              MOV     SP,R0
2539              TRAP     C#PNTX
2540              ADD     #4,SP
2541              BR      10:              ;DO ANOTHER MSG LINE
2542          20: MOV     #10,R0          ;NUMBER OF WORDS IN A READ STATUS BUFFER
2543              JSR     PC,PRMSGEXP      ;PRINT EXPD/RECV MESSAGE BUFFERS
2544              ENDMMSG
2545          L10013: TRAP     C#MSG
2546
2547          STATCOD: .WORD 1#,2#,3#,4#,5#,6#,0
2548          1#: .ASCIIZ 'ENSA Tape Bus Signals in Word #8:'
2549          2#: .ASCIIZ 'PARERR<15> IEOT <12> IFMK <9> IRDY<6> IRWD<2>'
2550          3#: .ASCIIZ 'IRESV2<14> IIDENT<11> IHER <8> IONL<5> IFBY<1>'
2551          4#: .ASCIIZ 'IRESV1<13> ICER <10> ISPEED<7> ILDP<4> IFPT<0>'
2552          5#: .ASCIIZ 'ENSA Tape Bus Signals in Word #9:'
2553          6#: .ASCIIZ 'DATMIS<7> ILW<6> OUTRDY<5> INRDY<4>'
2554          .EVEN

```

MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS

```

2550 .SBTTL MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
2551
2552
2553 ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
2554
2555 ;IMPLICIT INPUTS:
2556
2557 ; EXPMSG - EXPECTED MESSAGE BUFFER
2558 ; RECMMSG - RECEIVED MESSAGE BUFFER
2559 ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2560 ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2561
2562 013064 BGNMSG MSGLOOP
013064 MSGLOOP:
2563 013064 012701 013126 MOV #LOOPCOD,R1 ;ASCII ADDRESS TABLE
2564 013070 012100 10: MOV (R1)+,R0 ;DONE ALL MSG LINES?
2565 013072 001410 BEQ 20: ;BR IF YES
2566 013074 PRINTX R0 ;PRINT STATUS BIT NAMES
013074 C10046 MOV R0,-(SP)
013076 012746 000001 MOV #1,-(SP)
013102 010600 MOV SP,R0
013104 104415 TRAP C#PNTX
013106 062706 000004 ADD #4,SP
2567 013112 000766 BR 10: ;DO ANOTHER MSG LINE
2568 013114 012700 000012 20: MOV #10,R0 ;NUMBER OF WORDS IN A READ STATUS BUFFER
2569 013120 004737 014552 JSR PC,PRMSGEXP ;PRINT EXPD/RCV MESSAGE BUFFERS
2570 013124 ENDMMSG
013124 L10014:
013124 104423 TRAP C#MSG
2571
2572 013126 013146 013221 013320 LOOPCOD: .WORD 1#,2#,3#,4#,5#,6#,7#,0
2573 013146 045 116 045 1#: .ASCIZ 'NNA Tape Bus Loopback Signals in Word #8:'
2574 013221 045 116 045 2#: .ASCIZ 'NNA PARERR<15> IRESV2<14> IRESV1<13>'
2575 013320 045 116 045 3#: .ASCIZ 'NNA IHISP=>IEOT<12> IWRT=>IIDENT<11> IREV =>ICER <10>'
2576 013417 045 116 045 4#: .ASCIZ 'NNA IWM =>IFMK<09> IEDIT=>IHER <08> IFAD =>ISPEED<07>'
2577 013516 045 116 045 5#: .ASCIZ 'NNA ITADO=>IRDY<06> ITAD1=>IONL <05> IERASE=>ILDOP <04>'
2578 013615 045 116 045 6#: .ASCIZ 'NNA IREW =>IDBY<03> IRWU =>IRWD <02> IFEN =>IFBY <01>'
2579 013714 045 116 045 7#: .ASCIZ 'NNA IGO =>IFPT<00>'
2580 .EVEN

```

MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER

```

2582          .SBTTL MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
2583      ;*
2584      ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
2585      ;
2586      ;IMPLICIT INPUTS:
2587      ;
2588      ;     EXPMSG - EXPECTED MESSAGE BUFFER
2589      ;     RECMSG - RECEIVED MESSAGE BUFFER
2590      ;     RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2591      ;     RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2592      ;
2593      ;-
2594      BGNMSG MSGSUB
2595      MSGSUB:
2596      MOV     #10,R0          ;SIZE OF WRITE SUBSYSTEM BUFFER
2597      JSR     PC,PRMSGEXP     ;PRINT EXPD/RCV MESSAGE BUFFERS
2598      ENDMSG
2599      L10015:
2600      TRAP    C#MSG
2601      ;
2602      .SBTTL MEMADD - PRINT MEMORY ADDRESS DATA ERROR
2603      ;*
2604      ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
2605      ;
2606      ;IMPLICIT INPUTS:
2607      ;
2608      ;     ERRHI - MEMORY ERROR HIGH ORDER ADDRESS
2609      ;     ERRLO - MEMORY ERROR LOW ORDER ADDRESS
2610      ;     EXP   - EXPECTED DATA
2611      ;     RCV   - RECEIVED DATA
2612      ;
2613      ;-
2614      BGNMSG MEMADD
2615      MEMADD:
2616      JSR     PC,PRIADD       ;PRINT MEMORY ADDRESS IN ERROR
2617      MOV     EXPD,R1         ;GET EXPD DATA
2618      MOV     RCV,R2          ;GET RECEIVED DATA
2619      JSR     PC,PRIOR        ;PRINT EXPD/RCV
2620      ENDMSG
2621      L10016:
2622      TRAP    C#MSG

```

PRAMPKT - PRINT RAM AND PACKET DATA

```

2619 .SBTTL PRAMPKT - PRINT RAM AND PACKET DATA
2620 ;
2621 ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
2622 ;WHEN THE RAM DATA DOES NOT MATCH.
2623 ;
2624 ;INPUTS:
2625 ;
2626 ; R4 POINTER TO COMMAND PACKET
2627 ;IMPLICIT INPUTS:
2628 ; RAMDATA DATA AS READ FROM THE RAM
2629 ; RAMSIZ NUMBER OF BYTES IN PACKET
2630 ; IF RAMSIZ=0 THEN DEFAULT TO 8.
2631 ;
2632 ;IMPLICIT OUTPUTS:
2633 ; RAMSIZ SET TO 0
2634 ;
2635 PRAMPKT:
2636 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2637 MOV #RAMDATA,R1 ;DATA FROM THE RAM
2638 CLR R2 ;INIT BYTE NUMBER
2639 5$: CMPB (R1)+,(R4)+ ;COMPARE EXPECTED, RECEIVED
2640 BNE 7$ ;BR IF NO MATCH
2641 FORCERROR 7$,NOTSSR
2642 BR 10$ ;
2643 7$: MOVB -1(R1),R5 ;GET RECV RAM DATA
2644 MOVB -1(R4),R3 ;GET EXPD PACKET DATA
2645 XOR R5,R3 ;XOR EXPD/RECV
2646 BIC #177400,R3 ;LOW BYTE ONLY
2647 MOVB -1(R1),RECV ;GET RECEIVED RAM DATA
2648 MOVB -1(R4),EXPD ;GET EXPECTED RAM DATA
2649 PRINTB #RAMASC,R2,RECV,EXPD,R3
2650 MOV R3,-(SP)
2651 MOV EXPD,-(SP)
2652 MOV RECV,-(SP)
2653 MOV R2,-(SP)
2654 MOV #RAMASC,-(SP)
2655 MOV #5,-(SP)
2656 MOV SP,R0
2657 TRAP C:PNTB
2658 ADD #14,SP
2659 10$: INC R2 ;UPDATE BYTE COUNT
2660 TST RAMSIZ ;DEFAULT TO 8.?
2661 BEQ 15$ ;BR IF YES
2662 CMP R2,RAMSIZ ;DONE ALL BYTES?
2663 BLE 5$ ;BR IF NO
2664 BR 25$ ;
2665 15$: CMP R2,#8. ;DONE DEFAULT NUMBER OF BYTES?
2666 20$: BLT 5$ ;BR IF NO
2667 25$: CLR RAMSIZ ;SET DEFAULT RAMSIZ
2668 RTS PC ;RETURN
2669
2670 045 RAMASC: .ASCIZ 'N#A BYTE: #D2#A RAM: #03#A Packet: #03#A XOR:#03'
2671 .EVEN

```

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```

2664 .SBTTL PRMESS - PRINT CONTENTS OF MESSAGE BUFFER
2665 ;
2666 ; THIS ROUTINE PRINTS THE CONTENTS OF
2667 ; THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE TSV-05.
2668 ;
2669 ; INPUT:
2670 ; R0 LOW ORDER ADDRESS OF MESSAGE BUFFER
2671 ; R1 HIGH ORDER ADDRESS OF MESSAGE BUFFER
2672 ; NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
2673 ; THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
2674 ;
2675 PRMESS: SAVREG ;SAVE THE REGISTERS
2676 MOV R0,R5 ;SAVE LOW ORDER ADDRESS
2677 TST KTENABLE ;ADDRESS ABOVE 28K?
2678 BNE 10$ ;BR IF YES
2679 CLR R1 ;SET HIGH ORDER ADDRESS TO 0
2680 10$: MOV R1,R3 ;SAVE HIGH ORDER ADDRESS
2681 ROL R0 ;SHIFT BIT15 TO C BIT
2682 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
2683 PRINTX @PROASC,R1,R5 ;PRINT MESSAGE BUFFER ADDRESS
2684 014266 010546
2685 014270 010146
2686 014272 012746 014420
2687 014276 012746 000003
2688 014302 010600
2689 014304 104415
2690 014306 062706 000010
2691 014312
2692 014312 012746 014465
2693 014316 012746 000001
2694 014322 010600
2695 014324 104415
2696 014326 062706 000004
2697 014332 005004
2698 014334 010501
2699 014336 010300
2700 014340 001403
2701 014342 004737 017316
2702 014346 010005
2703 20$: PRINTX @PRASC,R4,(R5) ;PRINT THE CONTENTS OF MEMORY BUFFER
2704 MOV (R5),-(SP)
2705 MOV R4, -(SP)
2706 MOV @PRASC, -(SP)
2707 MOV @3, -(SP)
2708 MOV SP,R0
2709 TRAP C#PNTX
2710 ADD #10,SP
2711 INC R4 ;NUMBER OF THE NEXT
2712 CMP R4,#7 ;DONE ALL YET?
2713 BGT 50$ ;BRANCH IF ALL DONE
2714 BLT 20$ ;PRINT FIRST 7 WORDS
2715 BIT @X2.EXTF,XST2(R3);EXTENDED FEATURES ON?
2716 BNE 20$ ;PRINT EXTENDED STATUS WORD
2717 RTS PC ;RETURN
2718 50$: PROASC: .ASCIZ 'NMA Message Buffer Address = #01#05'
2719 PR1ASC: .ASCIZ 'NMA Message Buffer Contents:'
2720 PRASC: .ASCIZ 'NMA Word#D1#A: #0'

```

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```

2703 .EVEN
2704 .SBTTL PRMSGEXP - PRINT EXPD/RCV MESSAGE BUFFERS
2705
2706 ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
2707 ;RO - NUMBER OF WORDS IN BUFFER
2708 ;IMPLICIT INPUTS:
2709 ;EXPMSG - EXPECTED MESSAGE BUFFER
2710 ;RECMMSG - RECEIVED MESSAGE BUFFER
2711 ;RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2712 ;RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2713
2714 PRMSGEXP::
2715 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2716 MOV R0,R5 ;SAVE NUMBER OF WORDS
2717 MOV RCVLOADD,R0 ;GET RECV LOW ADDRESS
2718 MOV R0,R4 ;COPY LOW ADDRESS
2719 MOV RCVHIADD,R1 ;GET RECV HIGH ADDRESS
2720 ROL R0 ;SHIFT BIT15 TO C BIT
2721 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
2722 PRINTX @PRMSG0,R1,R4 ;PRINT MESSAGE BUFFER ADDRESS
2723 MOV R4,-(SP)
2724 MOV R1,-(SP)
2725 MOV @PRMSG0,-(SP)
2726 MOV @3,-(SP)
2727 MOV SP,R0
2728 TRAP C#PNTX
2729 ADD #10,SP
2730 PRINTX @PRMSG1 ;PRINT HEADER FOR CONTENTS
2731 MOV @PRMSG1,-(SP)
2732 MOV #1,-(SP)
2733 MOV SP,R0
2734 TRAP C#PNTX
2735 ADD #4,SP
2736 CLR R4 ;NUMBER OF THE CURRENT WORD
2737 MOV @EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
2738 MOV @RECMMSG,R2 ;GET RECV BUFFER ADDRESS
2739 MOV (R1),R0 ;GET EXPD
2740 MOV (R2),R3 ;GET RECV
2741 XOR R0,R3 ;XOR EXPD/RECV
2742 PRINTX @PRMSG2,R4,(R1),-(R2),R3
2743 MOV R3,-(SP)
2744 MOV (R2),-(SP)
2745 MOV (R1),-(SP)
2746 MOV R4,-(SP)
2747 MOV @PRMSG2,-(SP)
2748 MOV #5,-(SP)
2749 MOV SP,R0
2750 TRAP C#PNTX
2751 ADD #14,SP
2752 INC R4 ;NUMBER OF THE NEXT
2753 CMP R4,R5 ;DONE ALL YET?
2754 BGE 50$ ;BR IF YES
2755 BR 20$ ;DO ANOTHER
2756 RTS PC ;RETURN
2757
2758 50$:
2759 PRMSG0: .ASCIZ 'N#A Message Buffer Address - #01#05'
2760 PRMSG1: .ASCIZ 'N#A Message Buffer Contents:'
2761 PRMSG2: .ASCIZ 'N#A WORD #D2#A EXPD: #06#A RECV: #06#A XOR: #06'

```

PRMSGEXP - PRINT EXPD/RCV MESSAGE BUFFERS

```

2740 .EVEN
2741 .SBTTL PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
2742
2743 ;*
2744 ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
2745 ; ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
2746
2747 ; R0 - NUMBER OF BYTES IN BUFFER
2748
2749 ;IMPLICIT INPUTS:
2750
2751 ; EXPMSG - EXPECTED MESSAGE BUFFER
2752 ; RECMMSG - RECEIVED MESSAGE BUFFER
2753
2754 PRBYTEXP::
2755 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2756 MOV R0,R5 ;SAVE NUMBER OF BYTES
2757 CLM PRMNO ;INIT ERROR COUNT
2758 CLR R4 ;NUMBER OF THE CURRENT BYTE
2759 MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
2760 MOV #RECMMSG,R2 ;GET RCV BUFFER ADDRESS
2761 MOV (R1),R0 ;GET EXPD BYTE
2762 BIC #C<377>,R0 ;CLEAR UPPER BYTE
2763 MOV R0,PRBEXP ;SAVE FOR ERROR REPORT
2764 MOV (R2),R3 ;GET RCV BYTE
2765 BIC #C<377>,R3 ;CLEAR UPPER BYTE
2766 MOV R3,PRBREC ;FOR ERROR REPORT
2767 XOR R0,R3 ;XOR EXPD/RCV
2768 CMPB (R1)+,(R2)+ ;EXPD = RCV?
2769 BEQ 30$ ;BR IF YES
2770 INC PRMNO ;UPDATE ERROR COUNT
2771 CMP PRMNO,#8 ;PRINTED 8?
2772 BHI 30$ ;BR IF YES
2773 PRINTX #PRBMSG,R4,PRBEXP,PRBREC,R3
2774 MOV R3,-(SP)
2775 MOV PRBREC,-(SP)
2776 MOV PRBEXP,-(SP)
2777 MOV R4,-(SP)
2778 MOV #PRBMSG,-(SP)
2779 MOV #5,-(SP)
2780 MOV SP,R0
2781 TRAP C#PNTX
2782 ADD #14,SP
2783 FORCEXIT 50$ ;880
2784 BR 35$ ;880
2785
2786 30$: FORCERROR 27$,NOTSSR ;880
2787
2788 35$:
2789 INC R4 ;NUMBER OF THE NEXT
2790 CMP R4,R5 ;DONE ALL YET?
2791 BGE 50$ ;BR IF YES
2792 BR 20$ ;DO ANOTHER
2793 50$: PRINTX #PRBTOT,PRMNO ;PRINT TOTAL ERROR COUNT
2794 MOV PRMNO,-(SP)
2795 MOV #PRBTOT,-(SP)
2796 MOV #2,-(SP)
2797 MOV SP,R0

```

PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

```

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

```

RAMERR - PRINT RAM AND PACKET DATA

```

2833      ;IMPLICIT INPUTS:
2834      ;
2835      ;      RAMDATA      DATA AS READ FROM THE RAM
2836      ;      RAMSIZ      NUMBER OF BYTES IN PACKET
2837      ;                      IF RAMSIZ=0 THEN DEFAULT TO 8.
2838      ;
2839      ;IMPLICIT OUTPUTS:
2840      ;
2841      ;      RAMSIZ  SET TO 0
2842      ;
2843      ;
2844      015510      BGNMSG  RAMERR
2845      015510      RAMERR:: JSR      PC,PRAMPKT      ;PRINT RAM/PACKET DATA
2846      015514      004737 013776      ENDMSG
2847      015514      L10021: TRAP      C#MSG
2848      015514      104423      .SBTTL  RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
2849      ;
2850      ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
2851      ;
2852      ;IMPLICIT INPUTS:
2853      ;
2854      ;      R4      POINTER TO COMMAND PACKET
2855      ;
2856      ;IMPLICIT INPUTS:
2857      ;
2858      ;      RAMDATA      DATA AS READ FROM THE RAM
2859      ;      RAMSIZ      NUMBER OF BYTES IN PACKET
2860      ;                      IF RAMSIZ=0 THEN DEFAULT TO 8.
2861      ;      ERRHI      HIGH ORDER TEST ADDRESS
2862      ;      ERRLO      LOW ORDER TEST ADDRESS
2863      ;
2864      ;IMPLICIT OUTPUTS:
2865      ;
2866      ;      RAMSIZ  SET TO 0
2867      ;
2868      ;
2869      ;
2870      015516      BGNMSG  RAMTADD
2871      015516      RAMTADD:: JSR      PC,PRITADD      ;PRINT TEST ADDRESS
2872      015522      004737 010274      JSR      PC,PRAMPKT      ;PRINT RAM/PACKET DATA
2873      015526      004737 013776      ENDMSG
2874      015526      L10022: TRAP      C#MSG
2875      015526      104423      .SBTTL  RAMEXP - PRINT RAM EXPD/RECV DATA
2876      ;
2877      ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2878      ;
2879      ;IMPLICIT INPUTS:
2880      ;
2881      ;      R1      RECEIVED DATA
2882      ;      R2      EXPECTED DATA
2883      ;

```

RAMEXP - PRINT RAM EXPD/RECV DATA

```

2884      ;      R4      CONTROLLER RAM ADDRESS
2885      ; -
2886
2887 015530      BGNMSG  RAMEXP
2888 015530      RAMEXP::
2889 015530 042701 177400      BIC      #C<377>,R1      ;SAVE EXPD RAM DATA BYTE
2890 015534 042702 177400      BIC      #C<377>,R2      ;SAVE EXPD RAM DATA BYTE
2891 015540 004737 010066      JSR      PC,PRIRAM      ;PRINT THE RAM ADDRESS
2892 015544 004737 007742      JSR      PC,PRIXOR      ;PRINT THE DATA
2892 015550      ENDMSG
2893 015550      L10023:
2894 015550 104423      TRAP      C#MSG
2895
2896      .SBTTL  TIMEXP - PRINT TIMER A,B AND EXP/REC
2897
2898 ;+
2899 ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2900 ;AND TIMER A,B HEADER MESSAGE
2901 ;
2902 ;INPUTS:
2903 ;
2904 ;      R1      RECEIVED DATA
2905 ;      R2      EXPECTED DATA
2906 ; -
2907 015552      BGNMSG  TIMEXP
2908 015552      TIMEXP::
2909 015552 012746 015600      PRINTX  #TIMSGO      ;PRINT HEADER
2910 015556 012746 000001      MOV      #TIMSGO,-(SP)
2911 015562 010600      MOV      #1,-(SP)
2912 015564 104415      MOV      SP,R0
2913 015566 062706 000004      TRAP      C#PNTX
2914 015572 004737 007742      ADD      #4,SP
2915 015576      JSR      PC,PRIXOR      ;PRINT THE DATA
2916 015576      ENDMSG
2917 015576      L10024:
2918 015576 104423      TRAP      C#MSG
2919
2920 015600      045      116      045  TIMSGO: .ASCIZ  'N#A TIMER A STATUS IS IN BIT 3N#A TIMER B STATUS IS IN BIT 2'
2921      .EVEN
2922      .SBTTL  BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS
2923
2924 ;+
2925 ;PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
2926 ;
2927 ;INPUTS:
2928 ;
2929 ;      R1      CONTENTS OF TSSR
2930 ;      R2      DATA WRITTEN (8 BITS)
2931 ; -
2932 015700      BGNMSG  BADSSR
2933 015700      BADSSR::
2934 015700 010246      MOV      R2,-(SP)      ;SAVE DATA TRANSFERRED
2935 015702 042702 177400      BIC      #177400,R2      ;GET JUST ONE BYTE

```

BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

```

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

```

GLOBAL SUBROUTINES SECTION

```

2935      .SBTTL  GLOBAL SUBROUTINES SECTION
2936
2937      ;**
2938      ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2939      ; THAT ARE USED IN MORE THAN ONE TEST.
2940      ;--
2941      .SBTTL  SOFINIT - SOFT INITIALIZE OF CONTROLLER
2942
2943      ;*
2944      ;ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
2945      ;BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
2946      ;THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
2947      ;DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
2948
2949      ;
2950      ;INPUTS:
2951      ;
2952      ;      R5      ADDRESS OF FIRST REGISTER
2953
2954      ;OUTPUTS:
2955      ;
2956      ;      R0      CONTENTS OF TSSR, IF ERROR
2957      ;      CARRY   SET IF INIT WAS OKAY
2958      ;              CLEAR IF FATAL ERROR
2959
2960      ;CALLING SEQUENCE:
2961      ;
2962      ;      MOV      #ADDRESS,R5
2963      ;      JSR      PC,SOFINIT
2964      ;      BCS      CONTINUE
2965      ;      ERROF                      ;REPORT FATAL ERROR
2966
2967      ;-
2968
2969      SOFINIT::
2970      SAVREG
2971      MOV      #0,TSSR(R5)      ; SAVE THE REGISTERS
2972      JSR      PC,WAITF          ; DO THE INIT.
2973      MOV      TSSR(R5),R0      ; WAIT FOR SSR
2974      MOV      R0,R4            ; GET THE TSSR REGISTER
2975      BIC      #C<HIADDR!OFL>,R4 ; TSSR CONTENTS
2976      BIS      #SSR!NBA,R4      ; R4 HAS EXPECTED CONTENTS
2977      CMP      R4,R0            ; ONLY EXPECTED BITS SET ?
2978      BEQ      5$               ; BRANCH IF OKAY
2979      CLC                          ; CLEAR THE CARRY FOR ERROR
2980      BR      10$               ; GO TO EXIT
2981      5$:      SEC               ; SET THE CARRY BIT
2982      10$:     RTS              PC ; RETURN TO CALLER

```

CHKAMB - CHECK TSSR FOR AMBIGUITY

```

2984 .SBTTL CHKAMB - CHECK TSSR FOR AMBIGUITY
2985
2986 ;*
2987 ;
2988 ;THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
2989 ;FOR AMBIGUITY
2990 ;
2991 ;INPUT:
2992 ;
2993 ;      R0      CONTENTS OF TSSR
2994 ;
2995 ;OUTPUT:
2996 ;
2997 ;      R0      CONTENTS OF TSSR
2998 ;
2999 ;      CARRY   SET - NO AMBIGUITY
3000 ;             CLR - AMBIGUOUS CONTENTS
3001 ;
3002 ;-
3003
3004 CHKAMB:
3005     SAVREG      - ;SAVE THE GENERAL REGISTERS
3006     MOV         R0,R4      ;CONTENTS OF TSSR
3007     BIT         #SC,R0     ;IS BIT 15 SET ?
3008     BNE         5$         ;BRANCH IF YES
3009     BIT         #C<NBA!OFL!SSR!HIADDR>,R0 ;ANY OTHER BITS SET ?
3010     BNE         40$        ;MUST BE AN ERROR
3011     BR          45$        ;RETURN WITH SUCCESS
3012     5$: BIT      #SSR,R0   ;IS READY BIT SET ?
3013     BNE         10$        ;BRANCH IF READY BIT IS SET.
3014     BIT         #BIT5,R0   ;IS FATAL ERROR BIT SET ?
3015     BEQ         40$        ;ERROR IF NOT
3016     BIC         #CTERCLS,R4 ;CLEAR ALL BUT TERMINATION CODE
3017     CMP         R4,#16     ;ALL THREE BITS MUST BE SET
3018     BNE         40$        ;ERROR IF NOT SET
3019     BR          45$        ;OK IF ALL ARE SET
3020     10$: BIT      #BIT5,R0  ;IS FATAL ERROR BIT SET ?
3021     BEQ         45$        ;ERROR IF BIT IS SET WITH SSR
3022     BIT         #BIT2:BIT1,R0 ;IS THIS A FUNCTION REJECT
3023     BNE         45$        ;BR, IF TSSR IS OK
3024     40$: CLC              ;AMBIGUOUS CONTENTS
3025     BR          50$
3026     45$: SEC              ;SHOW SUCCESS - NO AMBIGUITY
3027     50$: RTS             PC ;RETURN TO CALLER

```

ENAIN,DSBINT - ENABLE/DISABLE INTERRUPTS

```

3029      .SBTTL ENAIN,DSBINT - ENABLE/DISABLE INTERRUPTS
3030      ;
3031      ; DEFAULT DISPLAY INTERRUPT HANDLERS.
3032      ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
3033      ; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
3034      ;
3035      ;
3036      ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
3037      ;
3038      IOKCKIN=BIT7      ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
3039      IOKSTP=BIT0       ; EXPECT "STOP" INTERRUPT.
3040      ;
3041      ; INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
3042      INTMASK: .BYTE 0
3043      ; INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
3044      INTFLAG: .BYTE 0
3045      ;
3046      ; SAVED INTERRUPT VECTOR:
3047      INTVEC: .WORD 0
3048      ; SAVE CPU PC
3049      INTCPC: .WORD 0
3050      ;
3051      ; SUBROUTINE TO ENABLE INTERRUPTS:
3052      ENAIN: MOV RO, -(SP)      ; SAVE RO
3053      MOV IVEC, RO      ; GET POINTER TO VECTORS
3054      MOV #INTR, (RO)+      ; SET UP INTERRUPT VECTOR
3055      MOV #PRI07, (RO)+
3056      MOV (SP)+, RO      ; RESTORE RO
3057      MOV (SP), -(SP)
3058      MOV #0, 2(SP)      ; SET CPU TO LEVEL 0
3059      RTI
3060      ;
3061      ; SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
3062      DSBINT: MOV (SP), -(SP)
3063      MOV #PRI07, 2(SP)
3064      RTI
3065      .SBTTL INTR - INTERRUPT HANDLERS
3066      ;
3067      BGNSRV INTR      ; DEFINE INTERRUPT ENTRY
3068      INTR:: MOV #1, INTRECV      ; SET FLAG TO SHOW INTERRUPT RECEIVED
3069      CLRB INTFLAG      ; CLEAR FLAG TO SAY WE GOT INTERRUPT
3070      BITB #IOKSTP, INTMASK      ; EXPECTING STOP INTERRUPT?
3071      BNE 1$      ; BR IF YES
3072      BISB #IOKSTP, INTFLAG      ; NO. SET THE ERROR FLAG.
3073      ;
3074      ; SAVE REGISTERS, MSG BUFFER, ETC.
3075      1$:
3076      ENDSRV
3077      L10026:
3078      RTI

```

WAITF - WAIT FOR SUBSYSTEM READY

```

3078 .SBTTL WAITF - WAIT FOR SUBSYSTEM READY
3079
3080 ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
3081 ;
3082 ; INPUTS:
3083 ;
3084 ; R5 ADDRESS OF FIRST DEVICE REGISTER
3085 ;
3086 ; OUTPUTS:
3087 ;
3088 ; R0 CONTENTS OF LAST TSSR READ
3089 ; CARRY SET - READY BIT SET
3090 ; CLR - TIMEOUT WAITING FOR READY
3091 ;
3092 WAITF:: BR 1# ;NOP WHEN SUPER FIXED
3093 BREAK ; DO A SUPVSR BREAK FIRST.
3094 TRAP C#BRK
3095 1# : MOV #11000,-(SP) ;25-APRIL-83 REV B - 1100 MSEC TIMER
3096 2# : MOV TSSR(R5),R0 ;READ THE TSSR REGISTER
3097 TSTB R0 ;TEST FOR READY BIT SET
3098 BMI 3# ; EXIT ON STOP FLAG.
3099 DELAY 1 ; WAIT 100 USEC
016270 MOV #1,(PC)+
016274 .WORD 0
016276 MOV L#DLY,(PC)+
016302 .WORD 0
016304 DEC -6(PC)
016310 BNE -4
016312 DEC -22(PC)
016316 BNE -20
3100 DEC (SP) ;REDUCE DELAY COUNT
3101 BNE 2# ;RETRY UNTIL TIMER EXPIRES
3102 CLC ; C = 0, CONTROLLER STILL RUNNING...
3103 BR 4# ;...OR HUNG-UP AFTER 300 MSEC.
3104 SEC ; C = 1, CONTROLLER IS STOPPED.
3105 3# : DEC (SP)+ ;RESTORE STACK WITHOUT CHANGING CARRY BIT
3106 4# : RTS PC

```

CHKTSSR - CHECK TSSR FOR READY

```

3108 .SBTTL CHKTSSR - CHECK TSSR FOR READY
3109 ;
3110 ; THIS ROUTINE WAITS FOR READY IN THE TSSR
3111 ; AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
3112 ;
3113 ; INPUT:
3114 ;       R5      ADDRESS OF CSR REGISTERS
3115 ;
3116 ; OUTPUT:
3117 ;       R0      CONTENTS OF TSSR
3118 ;       CARRY   SET - OKAY
3119 ;              CLR - NOT READY AMBIGUOUS. OR SC SET
3120 ;
3121 CHKTSSR:
3122     JSR     PC, WAITF      ; WAIT FOR READY
3123     BCC     20H            ; BRANCH IF TIME OUT
3124     JSR     PC, CHKAMB    ; TSSR AMBIGUOUS?
3125     BCC     10H            ; BR IF YES
3126     BIT     @SC, R0       ; SPECIAL CONDITION SET?
3127     BEQ     15H            ; BR IF NO
3128     BIT     @<SCE!BIE!RMR!NXM>, R0 ; ANY ERROR BITS SET?
3129     BEQ     15H            ; BR IF NO
3130     CLC
3131     BR      20H            ; SET FAILURE
3132     SEC
3133     RTS     PC             ; SET SUCCESS
3134     .SBTTL XNXM - CHECK FOR NONEXISTENT MEMORY
3135 ;
3136 ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
3137 ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
3138 ; "C" = 0, ALL ADDRESSES OK.
3139 ;
3140 ; CALL:  MOV ADR1, R1
3141 ;        MOV ADR2, R2
3142 ;        JSR PC, NXM
3143 ;        RETURN
3144 XNXM:  MOV     @2H, B04    ; TEST "C" AND PROCEED.
3145       MOV     @PRI04, B06  ; SET BUSERR VECTOR.
3146       CLR     R3          ; FLAG.
3147       TST     (R1)        ; TEST THE ADDRESS(ES).
3148       CMP     R1, R2      ; IF ANY TRAP, CONTINUE AT 2H.
3149       BEQ     3H          ; OTHERWISE, CONTINUE HERE.
3150       ADD     @2, R1      ; BR IF FINISHED (NO NEXM'S).
3151       BR      1H          ; SET NEXT ADDRESS...
3152       BR      1H          ; ...AND CONTINUE.
3153       COM     R3          ; GOT ONE, SET FLAG...
3154       MOV     @3H, (SP)
3155       RTI
3156       CLRVEC @4          ; ...AND DISMISS INTERRUPT...
3157       MOV     @4, R0      ; ...AND GIVE BACK THE VECTOR.
3158       TRAP    C0VEC
3159       TST     R3
3160       BEQ     .+4         ; DID WE CATCH ONE ??
3161       SEC     .+4         ; NO, "C" = 0, SKIP NEXT.
3162       RTS     PC         ; YES, "C" = 1, (R1) = NEXM ADDR.

```

TSTLOOP - CHECK ITERATION COUNT

```

3162          .SBTTL  TSTLOOP - CHECK ITERATION COUNT
3163
3164          ;*
3165          ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
3166          ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
3167          ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
3168          ;
3169          ; CALL: LOOPTO  ARG
3170          TSTLOOP::
3171              TST      NOITS          ; ITERATIONS INHIBITED?
3172              BNE      1$             ; YES.
3173              TST      QVP            ; NO.
3174              BMI      1$             ; LOOPS DISALLOWED IN QUICK PASS.
3175              DEC      LOOPCNT        ; BUMP LOOP COUNTER.
3176              BNE      2$
3177          1$:      CLC                  ; LOOP DISALLOWED, OR DONE.
3178              BR       3$
3179          2$:      SEC                  ; LOOP ENABLED.
3180          3$:      RTS      PC
3181
3182          .SBTTL  TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
3183
3184          ;*
3185          ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
3186          ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
3187          ; IN THE CURRENT RUN SEQUENCE.
3188          ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
3189          ;
3190          ; INPUT:
3191          ;       RO      POINTER TO TEST ID ASCIZ STRING
3192          ;
3193          ; OUTPUT:
3194          ;
3195          ;       R5      ADDRESS OF FIRST DEVICE REGISTER
3196          ;
3197          ; IMPLICIT OUTPUTS:
3198          ;
3199          ;       TSTCNT  UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
3200          ;
3201          ; SIDE EFFECTS:
3202          ;
3203          ;       INTERRUPT LEVEL IS RASIED TO LEVEL OF
3204          ;       THE DEVICE UNDER TEST
3205          ;
3206          ; -
3207
3208          TSTSETUP::
3209              MOV      RO, -(SP)      ; SAVE THE TEST ID MESSAGE
3210              CLR      SIFLAG          ; CLEAR "SOFT INIT" FLAG
3211              CLR      ERRK            ; CLEAR LOCAL ERROR COUNTER.
3212              CLR      EXTA            ; CLEAR ERROR EXTENSION FLAG.
3213              CLRB     INTMASK         ; CLEAR INTERRUPT MASK (CHECK ERROR)
3214              MOV      UNITN, RO       ; GET THE UNIT NUMBER.
3215              ASL      RO              ; ... AND MAKE IT A WORD OFFSET.
3216              TST      NODEV           ; DID STARTUP FIND THE DEVICE?
3217              BEQ      4$              ; BR IF YES
3218              BPL      3$              ; BR IF NOT IDLE

```

TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS

```

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

```

TSTEND - PRINT ERRORS RECEIVED

3250	016750	062706	000006						
	016754	000207			11:	ADD	PC	SP	
3251						RTS			
3252	016756	000000			ERRK:	0			; LOCAL ERROR COUNT.
3253	016760	045	101	040	ESUM:	.ASCIZ	/A D A ERRORS/		
3254	016777	105	122	122	EMAXDU:	.ASCIZ	/ERROR LIMIT REACHED	DROPPING UNIT/	
3255						.EVEN			

INCERK - INCREMENT LOCAL ERROR COUNT

```

3257          .SBTTL  INCERK  - INCREMENT LOCAL ERROR COUNT
3258
3259          ;* ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
3260          ;
3261 017044 005237 016756      INCERK: INC      ERRK          ; INCREMENT LOCAL ERROR COUNT
3262 017050 010046          MOV      RO,-(SP)      ; SAVE RO
3263 017052 013700 002202    MOV      UNITN,RO    ; GET UNIT NUMBER,
3264 017056 006300          ASL      RO           ; ... AND MAKE IT A WORD OFFSET.
3265 017060 062700 003176    ADD      @ERTABL,RO   ; RO GETS ADDRESS OF ERROR TABLE ENTRY.
3266 017064 005210          INC      (RO)         ; INCREMENT THE DEVICE ERROR COUNT
3267 017066 032710 007777    BIT      @7777,(RO)  ; DID WE OVERFLOW THE FIELD?
3268 017072 001001          BNE     1$           ; BR IF NO.
3269 017074 005310          DEC      (RO)         ; YES -- BACK IT UP TO 7777.
3270 017076 012600          1$:  MOV      (SP)+,RO ; RESTORE RO
3271 017100 000207          RTS      PC           ; RETURN TO CALLER.
3272
3273 017102 010046          CKEMAX: MOV      RO,-(SP) ; SAVE RO
3274 017104 013700 002202    MOV      UNITN,RO    ; GET UNIT NUMBER
3275 017110 006300          ASL      RO           ; ... AND MAKE IT A WORD OFFSET
3276 017112 016000 003176    MOV      ERTABL(RO),RO ; GET ERROR TABLE ENTRY
3277 017116 042700 170000    BIC      @170000,RO  ; EXTRACT ERROR COUNT FIELD
3278 017122 020037 002174    CMP      RO,GERRMAX  ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
3279 017126 103004          BHIS    1$           ; BR IF YES
3280 017130 023737 016756 002172  CMP      ERRK,LERRMAX ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
3281 017136 103417          BLO     2$           ; BR IF NO
3282 017140          1$:  RFLAGS    RO           ; GET OPERATOR FLAGS
3283 017142 032700 000040    TRAP     C#RFLA
3284 017146 001013          BIT      @IDU,RO      ; IS DROPPING INHIBITED?
3285 017150 012737 177777 003112  BNE     2$           ; BR IF YES.
3286 017156          MOV      @-1,DUFLG          ; NO -- DROP THE UNIT
3287 017156 104455          ERDF     4,EMAXDU
3288 017160 000004          TRAP     C#ERDF
3289 017162 016777          .WORD   4
3290 017164 000000          .WORD   EMAXDU
3291 017166          .WORD   0
3292 017166 013700 002202    DODU     UNITN
3293 017172 104451          MOV      UNITN,RO
3294 017174          TRAP     C#DODU
3295 017174 104444          DOCLN    TRAP     C#DCLN
3296 017176 012600          2$:  MOV      (SP)+,RO ; RESTORE RO
3297 017200 000207          RTS      PC           ; RETURN TO CALLER

```

CKDROP - CHECK IF UNIT SHOULD BE DROPPED

```

3292          .SBTTL CKDROP - CHECK IF UNIT SHOULD BE DROPPED
3293
3294          ; CHECK IF UNIT SHOULD BE DROPPED
3295          ;
3296 017202 010046          CKDROP: MOV      R0,-(SP)
3297 017204          FORCERROR      1$,NOTSSR
3298 017214          RFLAGS      R0
3299 017216 032700 000040          TRAP      C#RFLA
3300 017222 001010          BIT      #IDU,R0
3301 017224 011600          BNE      1$
3302 017226 012737 177777 003112          MOV      (SP),R0
3303 017234          MOV      #-1,DUFLG
3304 017242          DODU      UNITN
3305 017244 013700 002202          MOV      UNITN,R0
3306 017246 C00207          TRAP      C#DODU
3307
3308          DOCLN          ;ABORT THE PASS
3309          TRAP      C#DCLN
3310          MOV      (SP)+,R0
3311          RTS      PC
3312
3313          .SBTTL CONFIG - DETERMINE CONFIGURATION OF SYSTEM
3314          ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
3315          ;
3316          CONFIG:
3317          JSR      PC,SOFINIT
3318          RTS      PC
3319          .SBTTL KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT
3320          ; SUBROUTINE - ENABLE MEM MGT.
3321          ;
3322          KTON: TST      KTFLG          ; GOT KT?
3323          BEQ      1$          ; NO.
3324          MOV      #1,SRO          ; YES. ENABLE KT11.
3325          RTS      PC
3326
3327          ; SUBROUTINE - DISABLE MEM MGT.
3328          ;
3329          KTOFF: TST      KTFLG          ; GOT KT11?
3330          BEQ      1$          ; NO.
3331          NOP
3332          NOP
3333          MOV      #0,SRO          ; DISABLE KT.
3334          RTS      PC

```

SETMAP - SETUP PAR6 MAPPING

```

3335 .SBTTL SETMAP - SETUP PAR6 MAPPING
3336
3337 ;*
3338 ;THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
3339 ;AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
3340 ;IS RETURNED BIASED TO PAR6.
3341
3342 ;INPUTS:
3343
3344 ;      R0      HIGH ORDER ADDRESS BITS
3345 ;      R1      LOW ORDER ADDRESS BITS
3346
3347 ;OUTPUTS:
3348
3349 ;      R0      OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
3350 ;      CARRY   SET IF SUCCESS
3351 ;              CLR IF ERROR
3352
3353 ;-
3354 SETMAP:
3355     SAVREG                      ;SAVE R1-R4 UNTIL NEXT RETURN
3356     TST      KTFLG              ;SYSTEM HAVE ABOVE 28K?
3357     BEQ      10$                ;BR IF NO
3358     MOV      R1,R2              ;SAVE LOW ORDER BITS
3359     .REPT    6
3360     ASR      R0
3361     ROR      R1
3362     .ENDR
3363     BIC      #177,R1            ;ALIGN FOR LOWER 4K BOUNDARY
3364     CMP      R1,KTFLG           ;HIGHER THAN EXISTING MEMORY?
3365     BHIS     10$                ;BR IF YES
3366     MOV      R1,#KIPAR5         ;SETUP MAPPING REGISTER PAR5
3367     BIC      #160000,R2         ;SETUP DISPLACEMENT IN PAGE
3368     ADD      #120000,R2         ;ADD IN PAR5 BIAS
3369     MOV      R2,R0              ;RETURN IN R0
3370     SEC
3371     BR       15$                ;SET SUCCESS
3372     10$:    CLC                  ;SET FAILURE
3373     15$:    RTS      PC          ;RETURN
3374     .SBTTL  REGSAV - SAVE R1-R5 ON STACK
3375
3376 ;*
3377 ;ROUTINE TO
3378 ;SAVE R1 THROUGH R5 ON THE STACK
3379
3380 ;CALLING SEQUENCE:
3381
3382 ;      JSR      R5,REGSAV
3383
3384 ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
3385 ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
3386 ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
3387 ;REGISTERS.
3388
3389 ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
3390 ;CALLED VIA A JSR PC INSTRUCTION
3391

```

REGSAV - SAVE R1-R5 ON STACK

```
3392
3393
3394 017422
3395 017422 010446
3396 017424 010346
3397 017426 010246
3398 017430 010146
3399 017432 010546
3400 017434 016605 000012
3401 017440 004736
3402 017442 012601
3403 017444 012602
3404 017446 012603
3405 017450 012604
3406 017452 012605
3407 017454 000207
```

;-

REGSAV:

```
MOV R4,-(SP)
MOV R3,-(SP)
MOV R2,-(SP)
MOV R1,-(SP)
MOV R5,-(SP)
MOV 10.(SP),R5
JSR PC,8(SP)+
MOV (SP)+,R1
MOV (SP)+,R2
MOV (SP)+,R3
MOV (SP)+,R4
MOV (SP)+,R5
RTS PC
```

GETPAT - GET 8 BIT PATTERN FROM OPERATOR

```

3409          .SBTTL  GETPAT - GET 8 BIT PATTERN FROM OPERATOR
3410          ;*
3411          ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
3412          ;
3413          ;INPUTS:      NONE.
3414          ;
3415          ;OUTPUTS:
3416          ;      R0      OCTAL NUMBER FROM THE OPERATOR
3417          ;
3418          ;CALLING SEQUENCE:
3419          ;      JSR      PC.GETPAT
3420          ;-
3421          GETPAT::
3422          1$: SAVREG          ;SAVE THE GENERAL REGISTERS
3423          GMANID DATASC,PATDAT,0,377,0,377,NO
3424          TRAP    C$GMAN
3425          BR      10000$
3426          .WORD  PATDAT
3427          .WORD  T$CODE
3428          .WORD  DATASC
3429          .WORD  377
3430          .WORD  T$LOLIM
3431          .WORD  T$HILIM
3432          10000$: BNCOMPLETE 1$      ;RETRY IF ERROR
3433          BCC     1$
3434          MOV     PATDAT,R0      ;DATA PATTERN FROM OPERATOR
3435          RTS     PC            ;RETURN TO CALLER
3436
3437          ;*
3438          ;LOCAL DATA AREA
3439          ;-
3440          PATDAT: .WORD  0      ;TEMPORARY STORAGE FOR DATA
3441          DATASC: .ASCIZ 'ENTER DATA PATTERN'
3442          .EVEN

```

GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE

```

3436 .SBTTL GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE
3437 ;*
3438 ;ROUTINE TO ISSUE A MENU AND GET THE OPERATOR'S RESPONSE.
3439 ;
3440 ;INPUTS:
3441 ;      R0      ADDRESS OF ASCIZ STRING OF MENU
3442 ;      R1      MAXIMUM ALLOWABLE OPERATOR RESPONSE
3443 ;
3444 ;OUTPUTS:
3445 ;      R0      NUMBER OF THE OPERATOR'S SELECTION
3446 ;
3447 GETSEL::
3448     SAVREG                ;SAVE GENERAL REGISTERS
3449     MOV      R0,R2        ;SAVE THE MENU ADDRESS
3450     MOV      R2,R3        ;START OF MENU STRING
3451     1$:      TST      (R3)  ;END OF ASCII ?
3452     2$:      BEQ      3$    ;BRANCH IF ALL LINES DISPLAYED
3453     PRINTF   #SELASC,(R3)+ ;DISPLAY THE MENU
3454     MOV      (R3)+,-(SP)
3455     MOV      #SELASC, -(SP)
3456     MOV      #2, -(SP)
3457     MOV      SP,R0
3458     TRAP     C$PNTF
3459     ADD      #6,SP
3460     BR       2$
3461     3$:      GMANID      MENASC,MENRES,D,-1,0,-1,NO
3462     TRAP     C$GMAN
3463     BR       10001$
3464     .WORD    MENRES
3465     .WORD    T$CODE
3466     .WORD    MENASC
3467     .WORD    -1
3468     .WORD    T$LOLIM
3469     .WORD    T$HILIM
3470     10001$: BNCOMplete    1$    ;RETRY IF ERROR
3471     BCC      1$
3472     MOV      MENRES,R0      ;GET THE OPERATOR'S REPLY
3473     CMP      R0,R1          ;COMPARE TO MAXIMUM ALLOWED
3474     BLOS     5$             ;BRANCH IF OK
3475     PRINTF   #MENERR        ;DISPLAY ERROR MESSAGE
3476     MOV      #MENERR, -(SP)
3477     MOV      #1, -(SP)
3478     MOV      SP,R0
3479     TRAP     C$PNTF
3480     ADD      #4,SP
3481     BR       1$             ;RETRY
3482     5$:      RTS      PC    ;RETURN TO CALLER
3483     MENERR:   .ASCIZ      'MNA *** Menu Selection Too Large ***'
3484     SELASC:   .ASCIZ      'MNT'
3485     MENASC:   .ASCIZ      'Enter Menu Selection: '
3486     .EVEN
3487     MENRES:   .WORD       0

```

CHKMAN - CHECK MANUAL INTERVENTION LEGALITY

```

3469          .SBTTL  CHKMAN - CHECK MANUAL INTERVENTION LEGALITY
3470          ;*
3471          ;
3472          ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
3473          ;
3474          ;INPUT:
3475          ;
3476          ;      NONE.
3477          ;
3478          ;OUTPUT:
3479          ;
3480          ;      CARRY    0      MANUAL INTERVENTION NOT ALLOWED
3481          ;              1      MANUAL INTERVENTION IS OK
3482          ;
3483          ;SIDE EFFECTS:
3484          ;
3485          ;      A MESSAGE IS DISPLAYED WARNING THAT TEST IS
3486          ;      NOT EXECUTED IF MANUAL INTERVENTION IS NOT
3487          ;      ALLOWED.
3488          ;
3489          ;-
3490
3491          CHKMAN::
3492          SAVREG          ;SAVE THE REGISTERS
3493          MANUAL          ;SEE IF MANUAL INTERVENTION OK
3494          TRAP    C$MANI
3495          BCOMPLETE 1$    ;BRANCH IF ALLOWED
3496          BCS     1$
3497          PRINTF  $NOMAN  ;PRINT THE WARNING MESSAGE
3498          MOV     $NOMAN, -(SP)
3499          MOV     $1, -(SP)
3500          MOV     SP, R0
3501          TRAP    C$PNTF
3502          ADD     $4, SP
3503          CLC          ;CLEAR CARRY FOR ERROR
3504          RTS     PC    ;RETURN
3505
3506          1$:
3507          NOMAN: .ASCIZ  'N/A *** Manual Intervention not Allowed - Test Aborted ***'
3508          .even

```

ENVIRN - SETUP FREE DIAGNOSTIC SPACE

```

3502          .SBTTL  ENVIRN  - SETUP FREE DIAGNOSTIC SPACE
3503          ;
3504          ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
3505          ;
3506          ENVIRN: MEMORY  R0
3507          020112  104431      003124      TRAP      C#MEM
3508          020114  010037      003124      MOV       R0,FREE      ; GET 1ST FREE ADDRESS...
3509          020120  062737      000002      003124      ADD       #2,FREE
3510          020126  011037      003126      MOV       (R0),FRESIZ  ; ...AND WORD COUNT.
3511          020132  162737      000004      003126      SUB       #4,FRESIZ
3512          020140  013702      002012      MOV       L#UNIT,R2    ; GET NUMBER OF UNITS
3513          020144  162737      000007      003126  10#: SUB      #7,FRESIZ  ; TAKE AWAY 7 WORDS PER UNIT
3514          020152  005302      DEC        R2
3515          020154  001373      BNE        10#
3516          020156  013700      003124      MOV       FREE,R0      ;GET FIRST FREE ADDRESS
3517          020162  063700      003126      ADD       FRESIZ,R0    ;POINT TO LAST FREE ADDRESS
3518          020166  162700      000002      SUB       #2,R0        ;BACKUP 1 WORD
3519          020172  010037      003130      MOV       R0,FREEMH    ;STORE LAST FREE ADDRESS
3520          020176  C00240      NOP
3521          020200  0127J1      177520      MOV       #BDVPCR,R1    ;GET BDV11 PCR ADDRESS
3522          020204  010102      MOV       R1,R2                ;COPY TO R2
3523          020206  062702      000002      ADD       #2,R2        ;SET THE RANGE
3524          020212  004737      016376      JSR        PC,XNXM    ;SEE IF WE HAVE ONE
3525          020216  103001      BCC        15#                ;OK TO SET FLAGS
3526          020220  000423      BR         40#                ;RETURN WITH FLAGS CLEAR
3527          020222  013701      177520  15#: MOV       BDVPCR,R1    ;SAVE PCR CONTENTS
3528          020226  062701      000001      ADD       #1,R1        ;ADD ONE TO IT
3529          020232  012702      177520      MOV       #BDVPCR,R2    ;GET BDV11 PCR ADDRESS
3530          020236  005212      INC        (R2)                ;TRY TO WRITE TO IT
3531          020240  013703      177520      MOV       BDVPCR,R3    ;GET RESULTS
3532          020244  020103      CMP        R1,R3                ;DID IT CHANGE?
3533          020246  001006      BNE        20#                ;NO, MUST BE 11/23B
3534          020250  005237      003144      INC        T23A        ;SET THE FLAG
3535          020254  042737      170000      002120  BIC        #170000,L#HIME ;SUPERVISOR COULD BE WRONG
3536          ;
3537          ; PRINTF      #M8186    ;BR 40# FOR RELEASE
3538          020262  000402      BR         40#                ;TELL THE SYSTEM TYPE
3539          020264  005237      003146  20#: INC        T23B        ;RETURN
3540          ;
3541          ; PRINTF      #M8189    ;SET THE FLAG
3542          020270  000207      40#: NOP        ;BR 40# FOR RELEASE
3543          ;
3544          ; PRINTF      ;TELL THE SYSTEM TYPE
3545          ;
3546          ; RTS        PC      ;RETURN

```

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

3544 .SBTTL KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS
3545 ;*
3546 ;ROUTINE TO INIT KT-11
3547 ;
3548 ;-
3549
3550
3551 KTINIT:
3552 020272 005037 003132 CLR KTFLG ; INIT >28K MEMORY FLAG
3553 020276 005037 003134 CLR KTENABLE ; INIT TEST >28K FLAG
3554 020302 023727 002120 001577 CMP L#HIME,#1577 ; GOT ENOUGH MEMORY (>28K)?
3555 020310 101454 BLOS 9# ; NO.
3556 020312 013700 000004 MOV @ERRVEC,R0 ; SAVE OLD ERR VEC PTR.
3557 020316 012737 020430 000004 MOV @2#,@ERRVEC ; SET ERR VEC PTR.
3558 020324 005737 177572 TST @SRO ; GOT KT11?
3559 020330 000240 NOP ; (TRAP IF NO).
3560 020332 013737 002120 003132 MOV L#HIME,KTFLG ; YES. SET KT FLAG.
3561 020340 022737 010000 003132 CMP #10000,KTFLG ; MORE THAN 18 BITS?
3562 020346 100404 BMI 4# ; NO
3563 020350 042737 003777 003132 BIC #3777,KTFLG ;
3564 020356 000403 BR 5# ;
3565 020360 042737 000177 003132 4# BIC #177,KTFLG ;
3566 020366 010037 000004 5# MOV R0,@ERRVEC ; RESTORE OLD ERR VEC PTR.
3567 020372 005000 CLR R0 ; R0 = AR DATA.
3568 020374 012701 172340 MOV @KIPAR0,R1 ; R1 = KI REGS PTR.
3569 020400 012761 077406 177740 1# MOV #77406,-40(R1) ; SET DESCRIPTOR REG.
3570 020406 010021 MOV R0,(R1)+ ; SET KIPAR REG.
3571 020410 062700 000200 ADD #200,R0 ; BUMP AR DATA BY "4K".
3572 020414 020027 002000 CMP R0,#2000 ; AT "I/O"?
3573 020420 001367 BNE 1# ; NO.
3574 020422 012741 177600 MOV #177600,-(R1) ; YES. SET KTPAR7 FOR I/O.
3575 020426 000405 BR 9# ;
3576
3577 020430 012716 020436 2# MOV #6#,(SP) ; SET UP RETURN
3578 020434 000002 RTI ; RTI TO NEXT LOCATION
3579
3580 020436 010037 000004 6# MOV R0,@ERRVEC ; RESTORE OLD ERR VEC PTR.
3581
3582 020442 000207 9# RTS PC

```

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597 020444
3598 020444 013705 002206
3599 020450 004737 015774
3600 020454 103406
3601 020456 010001
3602 020460
      020460 104455
      020462 002261
      020464 003652
      020466 012034
3603 020470 104455
3604 020472
      020472 104406
3605 020474 013737 002202 076300
3606 020502 012704 076260
3607 020506 004737 010662
3608 020512 103406
3609 020514 010001
3610 020516
      020516 104456
      020520 002261
      020522 005056
      020524 012034
3611 020526 104456
3612 020530
      020530 104406
3613 020532 013701 075610
3614 020536 032701 000200
3615 020542 001020
3616 020544 012737 100206 020610
3617 020552 012737 020620 020612
3618 020560 012737 000006 020616
3619 020566 012737 100010 020620
3620 020574 012704 020610
3621 020600 004737 010662
3622 020604 000207
3623
3624
3625
3626
3627
3628 020610 000000
3629 020612 000000
3630 020614 000000

```

;*
 ; 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::
 mov ceraddr,r5
 jsr pc,sofinit
 bcs 25:
 mov r0,r1
 errdf errno,efierr,efimsg
 TRAP CIERDF
 .WORD 1201
 .WORD efierr
 .WORD efimsg
 trap clierdf
 25:
 ckloop
 TRAP CCLP1
 mov unitn,t39dex
 mov #t39pk2,r4
 jsr pc,wrtchr
 bcs 50:
 mov r0,r1
 errhrd errno,wrtmsg,efimsg
 TRAP CIERHRD
 .WORD 1201
 .WORD wrtmsg
 .WORD efimsg
 trap clierhrd
 50:
 ckloop
 TRAP CCLP1
 mov t39bfr+12,r1
 bit #bit7,r1
 bne 10:
 10:
 MOV #100206,CHDPKT ; WRT SUB-SYS MEM CMD
 MOV #WSMBK,CHDPKT+2 ; MSG BUF ADDR
 MOV #6,CHDPKT+6 ; BYTE COUNT
 MOV #100010,WSMBK ; INVERT THE SWITCH
 MOV #CHDPKT,R4 ; SET CHDPKT INTO R4
 JSR PC,WRTCHR ; DO IT
 RTS PC ; RETURN

;
 ; COMMAND PACKET.
 ;
 ; " <.3>6177774 ; MUST BE ON MOD 4 BOUNDARY.
 ;
 CHDPKT:: 0 ; 1ST WORD IS TS05 COMMAND.
 0 ; 2ND WORD IS THE BUFFER LOW ADDRESS.
 0 ; 3RD WORD IS THE BUFFER HIGH ADDRESS.

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

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

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

```

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

3683
3684
3685
3686 021002 013701 002120      NXMTST: MOV      L#HIME,R1      ;GET TOP OF MEMORY
3687 021006 062701 000200      ADD      #200,R1      ;MAKE IT I/O BLOCK OR OTHER NXM
3688 021012 042701 000177      BIC      #177,R1
3689 021016 010102 000006      MOV      R1,R2      ;RESAVE RESULTS
3690                      .REPT      6
3691                      ASL      R1      ;PUT IN PLACE FOR XFER
3692                      .ENDR
3693 021034 010137 003140      MOV      R1,NXMLO      ;SAVE TEST ADDRESS LOW
3694                      .REPT      10
3695                      ASR      R2      ;PUT IN PLACE FOR XFER
3696                      .ENDR
3697 021064 042702 177700      BIC      #177700,R2      ;DON'T WANT ILA!
3698 021070 010237 003142      MOV      R2,NXMHI      ;SAVE TEST ADDRESS HIGH
3699 021074 000207              RTS      PC      ;RETURN
3700
3701 021076                      ENDMOD
3710                      .TITLE    TSV4 - MISCELLANEOUS SECTIONS
3711
3712 021076                      BGNMOD    TSV4
3713
3719
3720
3721
3722                      .SBTTL    PROTECTION TABLE
3723 021076                      BGNPROT
3724 021076 177777 177777 177777 L#PROT:: .WORD    -1. -1. -1. 1      ;NO DEVICE PROTECTION REQUIRED.
3725 021106                      ENDPROT

```

INITIALIZE SECTION

```

3727                                     .SBTTL  INITIALIZE SECTION
3728
3729                                     ;**
3730                                     ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
3731                                     ;AT THE BEGINNING OF EACH PASS.
3732                                     ;
3733                                     ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
3734                                     ;IF "CONTINUE", NOTHING IS REQUIRED.
3735                                     ;
3736                                     ;--
3737                                     ;*
3738                                     ;INSERT TEMPORARY JUMP TO JDT
3739                                     ;-
3740                                     BGNINIT
3741 021106 L$INIT:
3742 021106 40$: CLR      EXTFEA
3743 021112 005037 002226 CLR      NXMFLG
3744 021116 012737 006360 002200 MOV     @EPRT1,EPRTSW      ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
3745 021124 C05037 003154 CLR      SIFLAG      ;CLEAR "SOFT INIT" FLAG
3746 021130 005037 003134 CLR      KTENABLE    ;CLEAR TEST ABOVE 28K FLAG
3747 021134 005037 002302 CLR      RAMSIZ      ;CLEAR RAM SIZE FOR RAMERR ROUTINE
3748 021140 READEF  @EF.CONTINUE
3749 021140 012700 000036 MOV     @EF.CONTINUE,R0
3750 021144 104447 TRAP     C$REFG
3751 021146 BNCOMPLETE 1$
3752 021150 103023 BCC      1$
3753 021156 023737 002202 002012 CMP     UNITN,L$UNIT      ;UNIT IN RANGE?
3754 021160 103070 BHS      4$      ;BR IF NO.
3755 021164 005737 003112 TST     DUFLG      ;DROPPED UNIT?
3756 021166 100472 BMI      NXTU      ;BR IF YES
3757 021172 013701 002202 MOV     UNITN,R1
3758 021174 006301 ASL      R1
3759 021176 005761 003176 TST     ERTABL(R1)
3760 021200 001516 BEQ      SETU
3761 021202 032761 040000 003176 BIT     @BIT14,ERTABL(R1)      ;DROPPED?
3762 021210 001060 BNE      NXTU
3763 021212 EXIT     INIT      ;DO NOTHING IF "CONTINUE".
3764 021214 104432 TRAP     C$EXIT
3765 021216 000416 .WORD    L10030-
3766 021218 1$: READEF  @EF.NEW
3767 021220 012700 000035 MOV     @EF.NEW,R0
3768 021222 104447 TRAP     C$REFG
3769 021224 BNCOMPLETE NXTU      ;TAKE NEXT UNIT IF NOT NEW PASS.
3770 021226 103052 BCC      NXTU
3771 021228 READEF  @EF.START
3772 021230 012700 000040 MOV     @EF.START,R0
3773 021232 104447 TRAP     C$REFG
3774 021234 BCOMPLETE 2$
3775 021236 103404 BCS      2$
3776 021238 READEF  @EF.RESTART
3777 021240 012700 000037 MOV     @EF.RESTART,R0
3778 021242 104447 TRAP     C$REFG
3779 021244 BNCOMPLETE 31$
3780 021246 103031 BCC      31$
3781 021248 2$: BRESET
3782 021250 104433 TRAP     C$RESET      ;1ST PASS, BUS-INIT...
                                     ;BUS RESET.

```

INITIALIZE SECTION

3768	021250	005037	002214		CLR	TSTCNT		;NUMBER OF TESTS RUN IN PASS
3769	021254	005037	002222		CLR	FATFLG		;CLEAR FATAL ERROR COUNT
3770	021260	005037	003144		CLR	T23A		;CLEAR 11/23A FLAG
3771	021264	005037	003146		CLR	T23B		;CLEAR 11/23B FLAG
3772					MOV	#340,-(SP)		
3773					MOV	#20,-(SP)		;RETURN TO DEBUGGER
3774					JMP	0.00T		;ENTER THE DEBUGGER
3775	021270	005037	003400		CLR	SKIPT		;CLEAR THE SUBTEST "SKIPPER"
3776	021274			20:				
3777	021274	012737	177777	002204	MOV	#-1,QVP		;...QUICK VERIFY...
3778	021302	004737	020112		JSR	PC,ENVIRN		;SET ENVIRONMENT.
3779	021306	004737	020272		JSR	PC,KTINIT		;INITIALIZE KT MEMORY MANAGEMENT
3780	021312	012700	003176		MOV	#ERTABL,RO		
3781	021316	005020		30:	CLR	(RO)+		;CLEAR THE ERROR TABLE
3782	021320	020027	003376		CMP	RO,#ERTABE		
3783	021324	103774			BLO	30		
3784	021326	000404			BR	4		
3785	021330	005037	002204	31:	CLR	QVP		
3786	021334	000137	021404		JMP	PASRPT		;GO REPORT THE STATUS
3787								
3788	021340			4:				
3789	021340	012737	177777	002202	NEWPAS:	MOV #-1,UNITN		;INIT UNIT NUMBER...
3790	021346	005037	002220		CLR	DEVCNT		;CLEAR COUNT OF DEVICES RUNNING
3791	021352				NXTU:	BREAK		
3792	021352	104422			TRAP	C#BRK		
3793	021354	005237	002202		INC	UNITN		;...AND SET NEXT UNIT NUMBER.
3794	021360	023737	002202	002012	CMP	UNITN,L#UNIT		
3795	021366	103423			BLO	SETU		
3796	021370	012737	177777	003112	MOV	#-1,DUFLG		
3797	021376	000401			BR	11		
3798	021400				DOCLN			;ABORT, NO MORE UNITS.
3799	021400	104444			TRAP	C#DCLN		
3800	021402	000240		11:	NOP			
3801	021404			PASRPT:				
3802	021404	023727	002012	000001	CMP	1#UNIT,#1		;HOW MANY UNITS SELECTED?
3803	021412	101752			BLOS	NEWPAS		;BR IF ONLY 1
3804	021414	005737	002220		TST	DEVCNT		;ARE ANY STILL RUNNING?
3805	021420	001747			BEQ	NEWPAS		;BR IF NO
3806	021422				RFLAGS	RO		
3807	021422	104421			TRAP	C#RFLA		
3808	021424	032700	000100		BIT	#ISR,RO		;SHOULD WE PRINT STATISTICS
3809	021430	001343			BNE	NEWPAS		;BR IF NO
3810	021432				DORPT			
3811	021432	104424			TRAP	C#DRPT		
3812	021434	000741			BR	NEWPAS		
3813	021436			10:				
3814	021436	013700	002202		SETU:	GPHARD UNITN,RO		;GET UNIT N P-TABLE POINTER.
3815	021442	104442			MOV	UNITN,RO		
3816	021444				TRAP	C#GPHRD		
3817	021444	103342			BNCOMPLETE	NXTU		;BR IF UNIT NOT AVAILABLE.
3818	021446	005037	003112		BCC	NXTU		
3819	021452	005237	002220		CLR	DUFLG		;CLEAR "DROPPED" FLAG.
3820	021456	012001			INC	DEVCNT		
3821	021460	010137	002206		MOV	(RO)+,R1		;GET 1ST REGISTER ADDRESS.
3822					MOV	R1,CSRADDR		;ADDRESS OF REGISTERS OF UNIT UNDER TEST

INITIALIZE SECTION

```

3818
3819 021464 012001      MOV      (R0)+,R1      ;GET VECTOR ADDRESS.
3820                  ;MOV      (R0),R2      ;GET INTERRUPT PRIORITY
3821                  ;MOV      R2,IPRI      ;SET INTERRUPT PRIORITY.
3822 021466 010137 002210 MOV      R1,IVEC      ;SET INTERRUPT VECTOR POINTER...
3823 021472 012721 016216 MOV      @INTR,(R1)+    ;...VECTOR...
3824 021476 013721 002212 MOV      IPRI,(R1)+    ;...AND PRIORITY.
3825
3826 021502             1$:
3827                  ; TST      QVP          ;1ST PASS ??
3828                  ; BEQ      5$          ;NO. SKIP THE PASS 1 STUFF.
3829
3830
3831                  ;
3832                  ;1ST PASS. CHECK THAT DEVICE ADDRESSES ARE VALID, AND
3833                  ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
3834
3834 021502 013701 002202      MOV      UNITN,R1
3835 021506 006301            ASL      R1
3836 021510 C52761 100000 003176 BIS      @BIT15,ERTABL(R1)    ;SAY DEVICE RUNNING
3837 021516 005037 005772      CLR      EXTA      ;CLEAR ERROR EXTENSION FLAG.
3838 021522 023727 002012 000001 CMP      L$UNIT,@1    ;ARE WE TESTING MULTIPLE UNITS?
3839 021530 101416            BLOS     10$      ;BR IF NO.
3840 021532            RFLAGS     RO      ;YES -- GET OPERATOR FLAGS.
3841 021534 104421            TRAP     C$RFLA
3842 021540 001412            BIT      @PNT,RO    ;SHOULD WE PRINT UNIT #?
3843 021542            BEQ      10$      ;BR IF NOT.
3844 021542            PRINTF     @PUNIT,UNITN    ;PRINT THE UNIT #
3845 021542 013746 002202      MOV      UNITN,-(SP)
3846 021546 012746 021634      MOV      @PUNIT,-(SP)
3847 021552 012746 000002      MOV      @2,-(SP)
3848 021556 010600            MOV      SP,RO
3849 021560 104417            TRAP     C$PNTF
3850 021562 062706 000006      ADD      @6,SP
3851
3851 021566            10$:
3852 021566 005037 003114      CLR      NODEV
3853 021572 013701 002206      MOV      CSRADDR,R1    ;ADDRESS OF FIRST REGISTER
3854 021576 010102            MOV      R1,R2      ;START OF REGISTERS
3855 021600 062702 000002      ADD      @TSSR,R2    ;ADDRESS OF TSSR REGISTER
3856 021604 004737 016376      JSR      PC,XNXM    ;TEST BOTH CONTROLLER REGISTERS...
3857 021610 103005            BCC      2$      ;...AND BR IF ALL OK.
3858 021612 010137 003114      MOV      R1,NODEV    ;FLAG DEVICE AS NON-EXISTENT
3859 021616 012737 177777 003112 MOV      @-1,DUFLG    ;DROP THIS UNIT.
3860 021624
3861
3861                  ;
3862                  ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
3863
3863 021624            2$:
3864 021624 012700 000000      5$:
3865 021630 104441            SETPRI   @PRI00      ;ENABLE INTERRUPTS.
3866 021632            MO      @PRI00,RO
3867 021632            TRAP     C$SPRI
3868 021632            ENDINIT
3869
3869 021632 104411            L10030:
3870 021634 045 116 045 PUNIT: TRAP      C$INIT
3871
3871                  ;.ASCIZ  /%N%N%A***** TESTING UNIT %D2%A *****/
3872                  ;.EVEN

```

```
.SBTTL  ADD AND DROP UNITS SECTIONS
```

```

3863                                     .SBTTL ADD AND DROP UNITS SECTIONS
3864
3865                                     ;**
3866                                     ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
3867                                     ; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
3868                                     ; OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
3869                                     ;--
3870 021702                                BGNAU
3871 021702                                L$AU::
3872 021702 010001                        MOV      R0,R1                ; GET UNIT TO BE ADDED (R0)
3873 021704 006301                        ASL      R1                  ; MAKE IT A WORD INDEX
3874 021706 052761 100000 003176        BIS      #100000,ERTABL(R1)    ; SET THE "ACTIVE" BIT
3875 021714 042761 040000 003176        BIC      #40000,ERTABL(R1)    ; CLEAR THE "DROPPED" BIT
3876 021722                                PRINTF      #1$,R0
3877 021722 010046                        MOV      R0,-(SP)
3878 021724 012746 021750                MOV      #1$,-(SP)
3879 021730 012746 000002                MOV      #2,-(SP)
3880 021734 010600                        MOV      SP,R0
3881 021736 104417                        TRAP     C$PNTF
3882 021740 062706 000006                ADD      #6,SP
3883 021744                                EXIT      AU
3884 021744 000167                        .WORD    J$JMP
3885 021746 000026                        .WORD    L10031-2-.
3886 021750 045 116 045 1$:              .ASCIZ  /$N$A UNIT $D$A ADDED/
3887                                     .EVEN
3888
3889                                ENDAU                                ; UNUSED.
3890 021776                                L10031:
3891 021776 104452                        TRAP     C$AU
3892
3893                                     ;**
3894                                     ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
3895                                     ; TO BE REMOVED FROM THE TEST LIST.
3896                                     ;
3897                                     ; SUPVSR DOES THE "DROPPING". THIS IS JUST TO TELL THE MAN.
3898                                     ; "DROPPED" UNITS ARE RE-SELECTED ON OPERATOR "STA" OR "ADD"
3899                                     ; COMMAND, OTHERWISE REMAIN INACTIVE. THE "DISPLAY" COMMAND
3900                                     ; WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
3901                                     ; WHICH ARE STILL ACTIVE.
3902                                     ; UPON ENTRY, R0 CONTAINS THE UNIT TO BE DROPPED.
3903
3904 022000                                BGNDU
3905 022000                                L$DU::
3906 022000 012737 177777 003112        MOV      #-1,DUFLG
3907 022006 010001                        MOV      R0,R1
3908 022010 006301                        ASL      R1
3909 022012 052761 140000 003176        BIS      #140000,ERTABL(R1)    ; SAY DROPPED
3910 022020 000240 000240 000240        240,240,240                ; ??????????
3911 022026                                PRINTF      #1$,R0
3912 022026 010046                        MOV      R0,-(SP)
3913 022030 012746 022054                MOV      #1$,-(SP)
3914 022034 012746 000002                MOV      #2,-(SP)
3915 022040 010600                        MOV      SP,R0
3916 022042 104417                        TRAP     C$PNTF
3917 022044 062706 000006                ADD      #6,SP
3918 022050                                EXIT      DU
3919 022050 000167                        .WORD    J$JMP
3920 022052 000030                        .WORD    L10032-2-.

```

```

3900 022054      045      116      045 14:      .ASCIZ  /NNA UNIT NDA DROPPED/
3901      .EVEN
3902 022104      L10032: ENDDU
      022104
      022104 104453      TRAP      C#DU

```

```

3903      022104      104455      TRAP      C450
3904      ;
3905      ; AUTO-DROP CODE SECTION.
3906      ; --
3906      022106      BGNAUTO
3907      022106      L$AUTO::
3908      022106      013705      002206      MOV      CSRADDR,R5      ;POINT TO DEVICE REGISTER
3909      022112      012703      000550      MOV      #360.,R3      ;ENOUGH TIME FOR 2400' REEL TO REWIND
3910      022116      004737      016250      10$:      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
3911      022122      103420      BCS      20$      ;LEAVE WHEN SSR IS SET
3911      022124      DELAY      250.      ;WAIT FOR .25 SECONDS
3911      022124      012727      000372      MOV      #250.,(PC)+
3911      022130      000000      .WORD      0
3911      022132      013727      002116      MOV      L$DLY,(PC)+
3911      022136      C00000      .WORD      0
3911      022140      005367      177772      DEC      -6(PC)
3911      022144      001375      BNE      .-4
3911      022146      005367      177756      DEC      -22(PC)
3911      022152      001367      BNE      .-20
3912      022154      005303      DEC      R3      ;BUMP COUNTER DOWN
3913      022156      001357      BNE      10$      ;KEEP GOING
3914      022160      004737      017202      JSR      PC,CKDROP      ;TRY AND DROP UNIT
3915      022164      20$:
3916      022164      ENDAUTO      ; UNUSED.
3916      022164      L10033:
3916      022164      104461      TRAP      C$AUTO

```

CLEAN-UP AND REPORT CODING SECTIONS

```

3918 .SBTTL CLEAN-UP AND REPORT CODING SECTIONS
3919
3920 ;**
3921 ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
3922 ; EXECUTED AT THE END OF EACH PASS (OR SUB-PASS).
3923 ; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
3924 ;--
3925 022166 BGNCLN
      022166 L$CLEAN::
3926 022166 013705 002206      MOV      CSRADDR,R5      ;POINT TO DEVICE REGISTER
3927 022172 005737 003112      TST      DUFLG          ;"DROPPED" FLAG IS SET ON...
3928 022176 100405              BMI      1$              ;...AND GROSS CONTROLLER FAULT...
3929                               ;...DON'T TRY TO XCT CLEANUP CODE.
3930
3931 022200 012765 000000 000002      MOV      #0,TSSR(R5)      ;DO SOFT INIT
3932 022206 004737 016250              JSR      PC,WAITF
3933 022212
3934 022212 104412      1$:
      022212      2$:      ENDCLN
      022212      L10034:      TRAP      C$CLEAN
3935
3936 ;**
3937 ; THE REPORT CODING SECTION CONTAINS THE
3938 ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
3939 ;--
3939 022214 BGNRPT
      022214 L$RPT::
3940 022214 PRINTS #DEVSUM
      022214 012746 022456      MOV      #DEVSUM,-(SP)
      022220 012746 000001      MOV      #1,-(SP)
      022224 010600              MOV      SP,R0
      022226 104416              TRAP      C$PNTS
      022230 062706 000004      ADD      #4,SP
3941 022234 010246              MOV      R2,-(SP)
3942 022236 010346              MOV      R3,-(SP)
3943 022240 010446              MOV      R4,-(SP)
3944 022242 012704 003176              MOV      #ERTABL,R4      ; GET START OF ERROR TABLE.
3945 022246 005003              CLR      R3      ; CLEAR UNIT NUMBER
3946 022250 011402      1$:      MOV      (R4),R2      ; GET ERROR TABLE ENTRY & TEST IT.
3947 022252 001467              BEQ      4$      ; ZERO IF UNIT NOT RUN
3948 022254 100066              BPL      4$
3949 022256 032702 040000              BIT      #BIT14,R2      ; WAS UNIT DROPPED?
3950 022262 001015              BNE      2$      ; BR IF YES
3951 022264 042702 170000              BIC      #1C7777,R2      ; GET ERROR COUNT FIELD
3952 022270 PRINTS #DEVONL,R3,R2      ; PRINT
      022270 010246              MOV      R2,-(SP)
      022272 010346              MOV      R3,-(SP)
      022274 012746 022513              MOV      #DEVONL,-(SP)
      022300 012746 000003              MOV      #3,-(SP)
      022304 010600              MOV      SP,R0
      022306 104416              TRAP      C$PNTS
      022310 062706 000010      ADD      #10,SP
3953 022314 000446              BR      4$
3954 022316 020227 160000      2$:      CMP      R2,#160000      ; WAS UNIT NON-EXISTENT?
3955 022322 001012              BNE      3$      ; BR IF NO
3956 022324 PRINTS #DEVNXR,R3
      022324 010346              MOV      R3,-(SP)
      022326 012746 022563              MOV      #DEVNXR,-(SP)

```

CLEAN-UP AND REPORT CODING SECTIONS

```

022332 012746 000002      MOV    #2,-(SP)
022336 010600      MOV    SP,R0
022340 104416      TRAP   C#PNTS
022342 062706 000006      ADD    #6,SP
3957 022346 000431      BR     4#
3958 022350 020227 160001 3# :  CMP    R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
3959 022354 001012      BNE    30#          ; BR IF NO.
3960 022356      PRINTS  #DEVNRD,R3
      022356 010346      MOV    R3,-(SP)
      022360 012746 022645      MOV    #DEVNRD,-(SP)
      022364 012746 000002      MOV    #2,-(SP)
      022370 010600      MOV    SP,R0
      022372 104416      TRAP   C#PNTS
      022374 062706 000006      ADD    #6,SP
3961 022400 000414      BR     4#
3962 022402 042702 170000 30# : BIC    #C7777,R2
3963 022406      PRINTS  #DEVDRD,R3,R2
      022406 010246      MOV    R2,-(SP)
      022410 C10346      MOV    R3,-(SP)
      022412 012746 022726      MOV    #DEVDRD,-(SP)
      022416 012746 000003      MOV    #3,-(SP)
      022422 010600      MOV    SP,R0
      022424 104416      TRAP   C#PNTS
      022426 062706 000010      ADD    #10,SP
3964 022432 062704 000002 4# :  ADD    #2,R4
3965 022436 005203      INC     R3
3966 022440 020427 003376      CMP    R4,#ERTABE
3967 022444 103701      BLO    1#
3968 022446 012604      MOV    (SP)+,R4
3969 022450 012603      MOV    (SP)+,R3
3970 022452 012602      MOV    (SP)+,R2
3971 022454      ENDRPT      ; UNUSED.
      022454      L10035:
      022454 104425      TRAP   C#RPT
3972
3973 022456 045 116 045 DEVSUM: .ASCIZ /#N#ADEVICE STATUS SUMMARY:#N/
3974 022513 045 101 040 DEVONL: .ASCIZ /#A UNIT #D3#A ONLINE, ERRORS = #D#N/
3975 022563 045 101 040 DEVNXR: .ASCIZ /#A UNIT #D3#A DROPPED, NON-EXISTENT REGISTER#N/
3976 022645 045 101 040 DEVNRD: .ASCIZ /#A UNIT #D3#A DROPPED, NOT READY AT STARTUP#N/
3977 022726 045 101 040 DEVDRD: .ASCIZ /#A UNIT #D3#A DROPPED, ERRORS = #D#N/
3978      .EVEN
3979
3980 022776      ENDMOD
3981

```

CLEAN-UP AND REPORT CODING SECTIONS

```
3985                                .TITLE  TSV5 - HARDWARE TESTS
3986
3993
3994 022776                        BGNMOD  TSV5
      022776                      TSV5::
4000
4008
```

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

```

4010          .SBTTL TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS
4011
4012          ;* TEST DESCRIPTION:
4013
4014          ; This test verifies that a Hardware Initialize command
4015          ; invoked after a Write Characteristics command sets up
4016          ; the Command, Message and Characteristic image blocks
4017          ; in the controller ram correctly.
4018
4019          ; TEST STEPS:
4020
4021          ; REPEAT FOR LOOPCNT
4022          ; BEGIN
4023          ; Do WRITE CHARACTERISTICS command.
4024          ; If the NBA bit in the TSSR register is NOT=0 then Print Error.
4025          ; Write to TSSR register to soft initialize the controller
4026          ; If controller RAM 310-377 NOT=0 then Print Error
4027          ; END
4028          ;--
4029
4030          BGNTST
4031          022776
4032          022776
4033          022776 012700 023432      MOV    #TST13ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
4034          023002 004737 016510      JSR    PC,TSTSETUP      ;DO INITIAL TEST SETUP
4035          023006 012737 000012 002216  MOV    #10.,LOOPCNT    ;PERFORM 10 ITERATIONS
4036          023014
4037          023014 004737 023706      JSR    PC,T13REST      ;SET PACKET TO START-UP VALUES
4038          023020
4039          023020 012703 002764      MOV    #TSTBLK+10.,R3    ;START OF TEST DATA
4040          023024 012704 023370      MOV    #T13PACKET,R4    ;GET THE ADDRESS OF COMMAND PACKET
4041          023030 012764 000010 000006  MOV    #8.,PKBCNT(R4)  ;START WITH MINIMUM ALLOWABLE VALUE
4042          023036
4043          023036 004737 015774      JSR    PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
4044          023042 103405              BCS    10$            ;BR IF SOFT INIT OKAY
4045          023044 010001              MOV    R0,R1            ;SAVE CONTENTS OF TSSR
4046          023046              ERRDF   ERRNO,SFIERR,SFIMSG    ;DEVICE FATAL DURING INIT
4047          023046 104455              TRAP    C#ERDF
4048          023050 000144              .WORD   100
4049          023052 003652              .WORD   SFIERR
4050          023054 012034              .WORD   SFIMSG
4051
4052          ;Do WRITE CHARACTERISTICS command.
4053          023056 005037 002222      10$: CLR    FATFLG      ;CLEAR FATAL ERROR FLAG
4054          023062 010465 000000      MOV    R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
4055          023066 004737 016336      JSR    PC,CHKTSSR     ;WAIT FOR SSR TO SET
4056          023072              FORCERROR 12$                ;SOFORCE ERROR IF FORCER=1
4057          023106 103407              BCS    15$            ;BR IF CARRY SET (GOOD RETURN)
4058          023110 010001              MOV    R0,R1            ;SAVE CONTENTS OF TSSR
4059          023112              NEXT.ERRNO
4060          023112 104455              12$: ERRDF   ERRNO,T13SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
4061          023112 000145              TRAP    C#ERDF
4062          023116 023617              .WORD   101
4063          023120 012046              .WORD   T13SSR
4064          023122 005237 002222      15$: INC    FATFLG      ;SET FATAL ERROR FLAG
4065          023126              CKLOOP    ;LOOP ON ERROR, IF FLAG SET

```

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

```

4063 023126 104406 000002      MOV    TSSR(R5),R1      ;GET THE CONTENTS OF TSSR
4064 023130 016501 000200      MOV    #SSR,R2      ;EXPECTED CONTENTS OF TSSR
4065 023134 012702 000100      BIT    #OFL,R1      ;IS OFF-LINE BIT SET ?
4066 023140 032701 000100      BEQ     25$      ;BRANCH IF NOT OFF-LINE
4067 023144 001402 000100      BIS     #OFL,R2      ;SET OFF-LINE IN EXPECTED DATA
4068 023146 052702 000100
4069
4070 023152      ;If the NBA bit in the TSSR register is NOT=0 then Print Error.
4071 023152 25$:      FORCERROR 27$      ;BBD
4072 023166 020201      CMP     R2,R1      ;DOES EXPECTED MATCH RECEIVED ?
4073 023170 001404      BEQ     30$      ;OKAY IF MATCH
4074 023172      NEXT.ERRNO
4075 023172 27$:      ERRHRD  ERRNO,T13NBA,PKTSSR      ;NBA NOT ZERO
4076 023172 104456      TRAP     C$ERHRD
4077 023174 000146      .WORD    102
4078 023176 023544      .WORD    T13NBA
4079 023200 012046      .WORD    PKTSSR
4080 023202 30$:      CKLOOP      ;LOOP ON ERROR ?
4081 023202 104406      TRAP     C$CLP1
4082
4083 023204      ;Write to TSSR register to soft initialize the controller
4084 023204 40$:      JSR     PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
4085 023204 004737 015774      FORCERROR 42$      ;BBD
4086 023210 004737      BCS     50$      ;BR IF SOFT INIT OKAY
4087 023210 103405      MOV     R0,R1      ;SAVE CONTENTS OF TSSR
4088 023224 010001      NEXT.ERRNO
4089 023226 42$:      ERRDF  ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
4090 023230 104455      TRAP     C$ERDF
4091 023230 000147      .WORD    103
4092 023232 003652      .WORD    SFIERR
4093 023234 012034      .WORD    SFIMSG
4094
4095 023240      ;If controller RAM 310-377 NOT=0 then Print Error
4096 023240 50$:      MOV     #310,R4      ;START WITH LOC 310
4097 023244 005002      CLR     R2      ;MEMORY EXPECTED SHOULD BE 000000
4098 023246 105065 000000      CLRB   TSDB(R5)      ;SET MAINTENANCE MODE
4099 023252 004737 016336      JSR     PC,CHKTSSR      ;WAIT FOR SSR READY
4100 023256 010465 000000 60$:      MOV     R4,TSDB(R5)      ;SELECT RAM ADDRESS
4101 023262 004737 016336      JSR     PC,CHKTSSR      ;WAIT FOR SSR READY
4102 023266 116501 000000      MOVB   TSBA(R5),R1      ;READ LOC CONTENTS
4103 023272 004737 016336      FORCERROR 62$,NOTSSR      ;BBD
4104 023272 120102      CMPB    R1,R2      ;CHECK MEMORY FOR 000000
4105 023272 001406      BEQ     70$      ;BRANCH IF DATA OKAY
4106 023272 62$:      NEXT.ERRNO
4107 023272 001406      ERRDF  ERRNO,T13MEM,AMEXP      ;MEMORY NOT ZERO AFTER INIT.
4108 023272 104455      TRAP     C$ERDF
4109 023272 000150      .WORD    104
4110 023272 023505      .WORD    T13MEM
4111 023272 015530      .WORD    AMEXP
4112 023272 005237 002222 70$:      INC     FATFLG      ;SET THE FATAL ERROR FLAG
4113 023272 104406      CKLOOP
4114 023272 005237 002222      TRAP     C$CLP1
4115 023272 104406      ESCAPE  TST      ;EXIT ON FATAL ERROR
4116 023272 104410      TRAP     C$ESCAPE
4117 023272 000426      .WORD    L10036-.

```

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

```

4103
4104 023330 005204      824: INC R4 ;LOOK AT NEXT RAM LOC.
4105 023332 020427 000400 CMP R4,#400 ;AT TOP OF RAM ADDRESS SPACE
4106 023336 001347 BNE 604 ;BRANCH TILL ALL MEMORY TESTED
4107
4108
4109 023340 005737 002222      TST FATFLG ;ANY FATAL ERRORS ?
4110 023344 001402 BEQ 1604 ;BRANCH IF NOT
4111 023346 004737 017202      JSR PC,CKDROP ;TRY TO DROP THE UNIT
4112 023352 004737 016456      JSR PC,TSTLOOP ;DONE ALL ITERATIONS?
4113 023356 103002 BCC 1654 ;BR IF YES
4114 023360 000137 023014      JMP T13LOOP ;LOOP UNTIL ITERATION COUNT DONE
4115 023364
4116 023364 104432      EXIT TST
      023366 000366      TRAP C#EXIT
      .WORD L10036-.

4117
4118
4119 ;*
4120 ;LOCAL STORAGE FOR THIS TEST
4121 ;*
4122
4126 023370      T13PACKET: ;COMMAND PACKET FOR TEST
4127 023370 100004      .WORD 100004 ;WRITE CHARACTERISTICS COMMAND, WITH ACK
4128 023372 023400      .WORD T13DATA ;ADDRESS OF CHARACTERISTICS BLOCK
4129 023374 000000      .WORD 0
4130 023376 000010      .WORD 8. ;STARTING VALUE OF BLOCK SIZE
4131
4132 023400      T13DATA: ;CHARACTERISTICS DATA BLOCK
4133 023400 023412      .WORD T13BFR ;ADDRESS OF MESSAGE BUFFER
4134 023402 000000      .WORD 0
4135 023404 000016      .WORD 14. ;LENGTH OF MESSAGE BUFFER
4136 023406 000000 000000      .WORD 0.0
4137
4138 023412      T13BFR: .BLKW 8. ;MESSAGE BUFFER
4139 ;LOCAL TEXT MESSAGES FOR TEST
4140 ;*
4141
4142 023432 111 156 151 TST13ID: .ASCIZ 'Initialization After WRITE CHARACTERISTICS'
4143 023505 111 156 143 T13MEM: .ASCIZ 'Incorrect RAM Data After Init'
4144      .EVEN
4145 023544 127 122 111 T13NBA: .ASCIZ 'WRITE CHARACTERISTICS Command Not Accepted'
4146 023617 103 157 156 T13SSR: .ASCIZ 'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
4147
4148
4149 ;*
4150 ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
4151 ;*
4152
4153      .EVEN
4154
4155
4156
4157 023706      T13REST: ;SAVE THE REGISTERS
4158 023706      SAVREG ;START OF THE PACKET
4159 023712 012701 023370      MOV #T13PACKET,R1 ;WRITE CHARACTERISTICS WITH ACK
4160 023716 012721 100004      MOV #100004,(R1)

```


TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

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

Line	Address	Hex	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op41
------	---------	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```

4354
4355
4357      025130
4359 025130      100204      T14PK2:  .=<..+10>E177770      ;COMMAND PACKET FOR TEST
4360 025130      100204      .WORD      100204      ;WRITE CHARA. MEM. CMND., WITH IE, ACK
4361 025132      025140      .WORD      T14DTA      ;ADDRESS OF SELECT DATA BLOCK
4362 025134      000000      .WORD      0
4363 025136      000010      .WORD      8.          ;STARTING VALUE OF BLOCK SIZE
4364
4365
4366 025140      024526      T14DTA:  .WORD      T14BFR      ;SELECT DATA BLOCK
4367 025140      000000      .WORD      0          ;ADDRESS OF MESSAGE BUFFER
4368 025142      000000      .WORD      0
4369 025144      000400      .WORD      256.        ;LENGTH OF MESSAGE BUFFER
4370 025146      000000      000000      .WORD      0.0
4371
4372
4373      ;*
4374      ;LOCAL TEXT MESSAGES FOR TEST
4375      ;-
4376
4377 025152      127      122      111  T14NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
4378 025226      127      122      111  T142REJ: .ASCIZ  'WRITE SUBSYSTEM MEMORY Not Rejected With Non-Zero Mode Field'
4379 025323      103      157      156  T14SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
4380 025413      105      170      160  T14NINT: .ASCIZ  'Expected Interrupt Not Received On WRITE SUBSYSTEM MEMORY'
4381 025505      111      156      143  T14TSBA: .ASCIZ  'Incorrect TSBA Address After WRITE SUBSYSTEM MEMORY'
4382 025571      102      141      163  TST14ID: .ASCIZ  'Basic WRITE SUBSYSTEM MEMORY Command'
4383
4384      .EVEN
4385
4386      ;*
4387      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
4388      ;WRITE SUBSYSTEM MEMORY COMMAND
4389      ;-
4390
4391
4392
4393 025636      012701      024510      T14REST: SAVREG      ;SAVE THE REGISTERS
4394 025636      012721      100206      MOV      #T14PACKET,R1      ;START OF THE PACKET
4395 025642      012721      024520      MOV      #100206,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK, IE
4396 025646      012721      024520      MOV      #T14DATA,(R1)+      ;ADDRESS OF DATA BLOCK
4397 025652      005021      000006      CLR      (R1)+      ;EXTENDED ADDRESS
4398 025656      012721      000006      MOV      #6.,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
4399 025660      005021      000006      CLR      (R1)+      ;CLEAR BSEL0 AND BSEL1
4400 025664      005021      000006      CLR      (R1)+      ;CLEAR SEL2
4401 025666      005011      000006      CLR      (R1)      ;CLEAR DATA AREA
4402 025670      000207      000006      RTS      PC          ;RETURN
4403 025672      000207
4404
4405
4406 025674      012701      025130      T14RST: SAVREG      ;SAVE THE REGISTERS
4407 025674      012721      100204      MOV      #T14PK2,R1      ;START OF THE PACKET
4408 025700      012721      025130      MOV      #100204,(R1)+      ;WRITE CHARA. WITH ACK, IE
4409 025704      012721      025140      MOV      #T14DTA,(R1)+      ;ADDRESS OF CHARAISTICS DATA BLOCK
4410 025710      005021      000010      CLR      (R1)+      ;EXTENDED ADDRESS
4411 025714      012721      000010      MOV      #8.,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
4412 025716

```

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

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

L10037: TRAP C#ETST

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```

4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449 025750
025750
4454 025750 012700 030020
4455 025754 004737 016510
4456 025760 012737 000012 002216
4457 025766 005237 003150
4458 025772 004737 020626
4459
4460 025776
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481

          .SBTTL TEST 3: DMA MEMORY ADDRESSING

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

          BGNTST
                                T3::
          MOV    #TST12ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
          JSR    PC,TSTSETUP      ;DO INITIAL TEST SETUP
          MOV    #10,LOOPCNT      ;PERFORM 10 ITERATIONS
          INC    T3BFLG           ;SET TEST FLAG
          JSR    PC,MEMCK         ;CHECK MEMORY

T12LOOP:      ;LOOP ON TEST LABEL

          .SBTTL TEST 3: SUBTEST 1: GET STATUS SELECTED LOCATIONS

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

```

TEST 3: SUBTEST 1: GET STATUS SELECTED LOCATIONS

```

4482      ; REPEAT FOR SELECTED VALID ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
4483      ; BEGIN
4484      ; Get a valid modulo-4 test address
4485      ; Do a GET STATUS command from the test address
4486      ; END
4487      ; END
4488      ; --
4489
4490 025776      BGNSUB                      ;////////// BEGIN SUBTEST //////////
4491 025776      T3.1:                      TRAP      C#BSUB
4492 025776      104402
4493
4494 026000      004737      015774      ;Write to TSSR to soft initialize
4495 026004      103405      JSR      PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
4496 026006      010001      BCS      15$      ;BR IF SOFT INIT = OK
4497 026006      010001      NEXT.ERRNO
4498 026010      104455      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4499 026010      000455      ERDF      ERRNO,SFIERR,SFMSG      ;DEVICE FATAL ERROR DURING INIT
4500 026012      003652      TRAP      C#ERDF
4501 026014      012034      .WORD      301
4502 026016      012034      .WORD      SFIERR
4503 026016      012034      .WORD      SFMSG
4504
4505 026020      012704      027610      ;Do a WRITE CHARACTERISTICS to setup a message buffer
4506 026024      004737      031170      15$:      MOV      #T12PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
4507 026030      005037      003134      JSR      PC,T12SWRT      ;RESTORE PACKET TO STARTING VALUES
4508 026034      010465      000000      CLR      KTENABLE      ;TURN OFF KT-11
4509 026040      004737      016336      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS
4510 026044      103405      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
4511 026060      010001      FORCERROR      17$      BCS      20$      ;BR IF SSR SET IN CHKTSSR
4512 026064      104455      17$:      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4513 026066      000456      17$:      NEXT.ERRNO
4514 026070      030122      17$:      ERDF      ERRNO,T12WRTSSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
4515 026072      012046      TRAP      C#ERDF
4516 026072      012046      .WORD      302
4517 026072      012046      .WORD      T12WRTSSR
4518 026072      012046      .WORD      PKTSSR
4519
4520 026074      005037      002222      ;Verify a Get Status can be fetched from each address
4521 026100      005037      027660      ;Get a valid modulo-4 test address
4522 026104      012702      027664      ;Do a GET STATUS command from the test address
4523 026110      005037      003134      20$:      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
4524 026114      012201      027660      CLR      T12KT      ;TEST ABOVE 28K SWITCH
4525 026116      005000      T121LOOP:      MOV      #T12BLK,R2      ;POINT TO TEST PATTERN TABLE
4526 026120      005737      027660      CLR      KTENABLE      ;TURN OFF ABOVE 28K TEST FLAG
4527 026124      001407      177776      MOV      (R2)+,R1      ;GET TEST PATTERN ADDRESS
4528 026126      016200      177774      CLR      R0      ;ASSUME NO TEST ABOVE 28K
4529 026132      042700      177774      TST      T12KT      ;TEST ABOVE 28K THIS TIME?
4530 026136      012737      000001      003134      BEQ      25$      ;BR IF NO
4531 026144      004737      030666      MOV      -2(R2),R0      ;GET TEST PATTERN AGAIN
4532 026144      004737      030666      BIC      #C<A1716>,R0      ;SAVE 16 BIT ADDRESS ONLY
4533 026144      004737      030666      MOV      #1,KTENABLE      ;TURN ON ABOVE 28K TEST FLAG
4534 026144      004737      030666      25$:      JSR      PC,T12CONVERT      ;CONVERT TEST PATTERN TO TEST ADDRESS

```

Address	Hex	Hex	Hex	Label	Code	Comment
4529	026150	103034			BCC 65\$	;BR IF INVALID PACKET ADDRESS
4530	026152	013704	027654		MOV T12LOAD,R4	;COPY CURRENT PACKET LOW ADDRESS
4531	026156	013703	027652		MOV T12HIADD,R3	;COPY CURRENT PACKET HIGH ADDRESS
4532	026162	004737	031236		JSR PC,T12SETGET	;SETUP CURRENT PACKET TO GET STATUS
4533	026166	042703	177774		BIC #1C<A1716>,R3	;SAVE ADDRESS BITS 17-16
4534	026172	050304			BIS R3,R4	;SETUP 16 BIT PACKET ADDRESS
4535	026174	004737	017274		JSR PC,KTOFF	;TURN OFF KT-11
4536	026200	010465	000000		MOV R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE
4537	026204	004737	016336		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET
4538	026210				FORCERROR 32\$	
4539	026224	103405			BCS 40\$	;BR IF SSR SET IN CHKTSSR
4540	026226	010001			MOV R0,R1	;SAVE CONTENTS OF TSSR
4541	026230				NEXT.ERRNO	
4542	026230			32\$:	ERRDF ERRNO,T12GETSSR,PKTGETS	;DEVICE FATAL SSR FAILED TO SET
	026230	104455				TRAP C\$ERDF
	026232	000457				.WORD 303
	026234	030046				.WORD T12GETSSR
	026236	012064				.WORD PKTGETS
4543	026240			40\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
	026240	104406				TRAP C\$CLP1
4544	026242			65\$:		
4545	026242				FORCEXIT 80\$	
4546	026252	020227	030016		CMP R2,#T12TBE	;DONE ALL 1STBLK TEST PATTERNS?
4547	026256	103002			BHIS 70\$	;BR IF YES
4548	026260	000137	026110		JMP T121LOOP	;DO ANOTHER MODULO- 4 ADDRESS
4549	026264	005737	027660	70\$:	TST T12KT	;DONE ABOVE 28K TESTING TOO?
4550	026270	003012			BGT 80\$	;BR IF YES
4551	026272	005737	003132		TST KTFLG	;ANY MEMORY ABOVE 28K ON SYSTEM?
4552	026276	001407			BEQ 80\$	;BR IF NO
4553	026300	012737	000001	027660	MOV #1,T12KT	;SET SWITCH
4554	026306	012702	027664		MOV #T12BLK,R2	;RESET TEST PATTERN TABLE
4555	026312	000137	026110		JMP T121LOOP	;DO ABOVE 28K TESTING
4556	026316	004737	017274	80\$:	JSR PC,KTOFF	;TURN OFF KT11
4557	026322				ENDSUB	;////////// END SUBTEST //////////
	026322					L10043:
	026322	104403				TRAP C\$ESUB
4558	026324	005737	002222		TST FATFLG	;ANY FATAL ERRORS ?
4559	026330	001402			BEQ 100\$	;BRANCH IF NOT
4560	026332	004737	017202		JSR PC,CKDROP	;TRY TO DROP THE UNIT
4561	026336			100\$:		

```

4562      .SBTTL  TEST 3: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS
4563
4564      ;**
4565      ; TEST 3: SUBTEST 2:
4566      ;
4567      ; SUBTEST DESCRIPTION:
4568      ;
4569      ;       This subtest verifies the controller can deposit message packets
4570      ;       to all available memory locations.
4571      ;       Write Characteristics commands are executed with message
4572      ;       buffer addresses set to various addresses in each available
4573      ;       memory location.
4574      ;       The various addresses are determined by floating a 1 then a 0
4575      ;       through the address bits.
4576      ;
4577      ; TEST STEPS:
4578      ;

```

TEST 3: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS

```

4579      ; BEGIN
4580      ;   Write to TSSR to soft initialize
4581      ;   Do a WRITE CHARACTERISTICS to setup a message buffer to compare
4582      ;
4583      ;   REPEAT FOR SELECTED ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
4584      ;   BEGIN
4585      ;       Get a valid modulo-4 test address
4586      ;       Set the packet message buffer to the TEST ADDRESS
4587      ;       Do a WRITE CHARACTERISTICS
4588      ;       Restore the test message buffer to background pattern
4589      ;   END
4590      ; END
4591      ; --
4592
4593      BGNSUB                                ;////////// BEGIN SUBTEST //////////
4594      026336                                T3.2:
4595      026336                                TRAP      C#BSUB
4596      026336 104402
4597      ;Write to TSSR to soft initialize
4598      JSR      PC,SOFINIT                    ;DO SOFT INIT OF CONTROLLER
4599      BCS      15$                          ;BR IF SOFT INIT = OK
4600      NEXT.ERRNO
4601      MOV      R0,R1                        ;SAVE CONTENTS OF TSSR
4602      ERDF     ERRNO,SFIERR,SFMSG          ;DEVICE FATAL ERROR DURING INIT
4603      026350 104455                        TRAP      C#ERDF
4604      026352 000460                        .WORD     304
4605      026354 003652                        .WORD     SFIERR
4606      026356 012034                        .WORD     SFMSG
4607
4608      ;Do a WRITE CHARACTERISTICS to setup a message buffer to compare
4609      15$:
4610      MOV      #T12PACKET,R4                ;GET THE ADDRESS OF COMMAND PACKET
4611      JSR      PC,T12SWRT                   ;SET PACKET TO WRITE CHARACTERISTICS
4612      JSR      PC,KTOFF                     ;TURN OFF KT-11
4613      MOV      R4,TSD0(R5)                  ;SET THE PACKET ADDRESS
4614      JSR      PC,CHKTSSR                   ;WAIT FOR SSR TO SET
4615      FORCERROR 17$
4616      BCS      20$                          ;BR IF SSR SET IN CHKTSSR
4617      MOV      R0,R1                        ;SAVE CONTENTS OF TSSR
4618      NEXT.ERRNO
4619      17$: ERDF     ERRNO,T12WRTSSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
4620      026424 104455                        TRAP      C#ERDF
4621      026426 000461                        .WORD     305
4622      026430 030122                        .WORD     T12WRTSSR
4623      026432 012046                        .WORD     PKTSSR
4624
4625      ;Get a valid modulo-4 test address
4626      ;Set the packet message buffer to the test address
4627      ;Do a WRITE CHARACTERISTICS
4628      20$: CLR      FATFLG                    ;CLEAR FATAL ERROR FLAG
4629      MOV      #T12BLK,R3                   ;POINT TO TEST PATTERN TABLE
4630      T122LOOP:
4631      MOV      (R3)+,R1                     ;GET TEST PATTERN ADDRESS
4632      MOV      R1,R0                        ;GET ADDRESS ALL "18 BITS"
4633      BIC      #177774,R0                   ;LEAVE ONLY A17 AND A16
4634      BIC      #3,R1                       ;GET RID OF A17 AND A16
4635
4636      005037 002222
4637      026440 012703 027664
4638      026444
4639      026444 012301
4640      026446 010100
4641      026450 042700 177774
4642      026454 042701 000003

```

TEST 3: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS

```

4626 026460 004737 030666      JSR      PC,T12CONVERT      ;CONVERT TEST PATTERN TO TEST ADDRESS
4627 026464 103402              BCS      25$              ;BR IF VALID MESSAGE BUFFER ADDRESS
4628 026466 000137 026564      JMP      150$              ;GET ANOTHER TEST PATTERN TO TRY
4629 026472 012704 027610      MOV      #T12PACKET,R4      ;SET THE COMMAND PACKET ADDRESS
4630 026476 004737 031170      JSR      PC,T12SWRT      ;SETUP T12PACKET TO WRITE CHAR.
4631 026502 013737 027654      MOV      T12LOAD,T12DATA      ;SETUP LOW ORDER MESSAGE BUFFER ADD.
4632 026510 013737 027652      MOV      T12HIADD,T12DATA+2      ;SETUP HIGH ORDER MESSAGE BUFFER ADD.
4633 026516 004737 017274      JSR      PC,KTOFF      ;TURN OFF KT-11
4634 026522 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
4635 026526 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
4636 026532              FORCERROR      32$
4637 026546 103405              BCS      50$              ;BR IF SSR SET IN CHKTSSR
4638 026550 010001              MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4639 026552              NEXT.ERRNO
4640 026552              ERRDF      ERRNO,T12WRTSSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
4641 026562              104455
4642 026564              000462
4643 026566              030122
4644 026568              C12046
4645 026570              50$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
4646 026572              104406
4647 026574              150$:
4648 026576              FORCEEXIT      160$
4649 026578              CMP      R3,#T12TBE      ;DONE ALL TST12BLK TEST PATTERN?
4650 026580              BHIS      160$      ;BR IF YES
4651 026582              JMP      T122LOOP      ;DO ANOTHER MODULO- 4 ADDRESS
4652 026584              160$:      JSR      PC,KTOFF      ;TURN OFF KT11
4653 026586              ENDSUB
4654 026588              ;///////////////// END SUBTEST ///////////////////
4655 026590              L10044:
4656 026592              TST      FATFLG      ;ANY FATAL ERRORS ?
4657 026594              BEQ      180$      ;BRANCH IF NOT
4658 026596              JSR      PC,CKDROP      ;TRY TO DROP THE UNIT
4659 026598              180$:

```

.SBTTL TEST 3: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS

```

;
; TEST 3: SUBTEST 3:

```

SUBTEST DESCRIPTION:

```

;
; This subtest verifies the controller can fetch a
; Write Characteristics data block from all available
; memory locations.
; Write Characteristics commands are executed with
; characteristic data blocks at various memory addresses.
; The various memory addresses are determined by floating
; a 1 then a 0 through the address bits.

```

TEST STEPS:

```

; BEGIN
; Write to TSSR to soft initialize
;
; REPEAT FOR SELECTED VALID ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K

```

TEST 3: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS

```

4676      ;          BEGIN
4677      ;          Get a valid test address
4678      ;          Set the test packet characteristic data pointer to the
4679      ;          test address.
4680      ;          Store expected characteristic data in test address block
4681      ;          Do a WRITE CHARACTERISTIC command
4682      ;          END
4683      ;          END
4684      ;          END
4685      ;          END
4686      026626      BGNSUB          ;//////////////// BEGIN SUBTEST //////////////////
4687      026626      T3.3:          TRAP      C#BSUB
4688      026626      104402
4689      ;Write to TSSR to soft initialize
4690      026630      004737      015774      JSR      PC,SOFINIT          ;DO SOFT INIT OF CONTROLLER
4691      026634      103405      BCS      20#          ;BR IF SOFT INIT = OK
4692      026636      NEXT.ERRNO
4693      026636      010001      MOV      R0,R1          ;SAVE CONTENTS OF TSSR
4694      026640      ERRDF      ERRNO,SFIERR,SFMSG      ;DEVICE FATAL ERROR DURING INIT
4695      026640      104455      TRAP      C#ERRDF
4696      026642      000463      .WORD      307
4697      026644      003652      .WORD      SFIERR
4698      026646      012034      .WORD      SFMSG
4699      ;Get a valid test address
4700      20#      CLR      FATFLG          ;CLEAR FATAL ERROR FLAG
4701      20#      CLR      T12KT          ;TEST ABOVE 28K SWITCH
4702      20#      MOV      #T12BLK,R3      ;POINT TO TEST PATTERN TABLE
4703      T123LOOP:
4704      CLR      KTENABLE          ;TURN OFF ABOVE 28K TEST FLAG
4705      MOV      (R3),R1          ;GET TEST PATTERN ADDRESS
4706      MOV      R1,R0          ;GET ADDRESS ALL "18 BITS"
4707      BIC      #177774,R0      ;LEAVE ONLY A17 AND A16
4708      BIC      #3,R1          ;GET RID OF A17 AND A16
4709      TST      T12KT          ;TEST ABOVE 28K THIS TIME?
4710      BEQ      25#          ;BR IF NO
4711      MOV      -2(R3),R0      ;GET TEST PATTERN AGAIN
4712      BIC      #177774,R0      ;SAVE 18 BIT ADDRESS ONLY
4713      MOV      #1,KTENABLE      ;TURN ON ABOVE 28K TEST FLAG
4714      25#      JSR      PC,T12CONVERT      ;CONVERT TEST PATTERN TO TEST ADDRESS
4715      BCS      30#          ;BR IF VALID TEST ADDRESS
4716      JMP      60#          ;GET NEXT TEST PATTERN
4717      ;Set the test packet characteristic data pointer to the test address
4718      30#      MOV      #T12PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
4719      JSR      PC,T12SWRT      ;RESTORE PACKET TO STARTING VALUES
4720      MOV      T12LOAD,PKLOW(R4)      ;STORE CHAR. DATA PTR LOW ADDRESS
4721      MOV      T12HIADD,PKHI(R4)      ;STORE CHAR. DATA PTR HIGH ADDRESS
4722      JSR      PC,T12CHAR      ;STORE EXPECTED DATA IN DATA BLOCK
4723      ;Do a WRITE CHARACTERISTIC command
4724      JSR      PC,KTOFF          ;TURN OFF KT-1.1
4725      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
4726      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
4727      FORCERROR      32#
4728      BCS      40#          ;BR IF SSR SET IN CHKTSSR
4729      MOV      R0,R1          ;SAVE CONTENTS OF TSSR

```

TEST 3: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS

```

4727 027026
4728 027026 104455
      027030 000464
      027032 030122
      027034 012046
4729 027036 104406
      027036 104406
4730 027040
4731 027040 020327 030016
4732 027044 103002
4733 027046 000137 026664
4734 027052 005737 027660
4735 027056 003012
4736 027060 005737 003132
4737 027064 001407
4738 027066 012737 000001 027660
4739 027074 012703 027664
4740 027100 C00137 026664
4741 027104 004737 017274
4742 027110
      027110
      027110 104403
4743 027112 005737 002222
4744 027116 001402
4745 027120 004737 017202
4746 027124

```

```

32:  NEXT.ERRNO
     ERRDF  ERRNO,T12WRTSSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                     TRAP  C0ERDF
                                     .WORD 308
                                     .WORD T12WRTSSR
                                     .WORD PKTSSR
40:  CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                     TRAP  C0CLP1
60:
65:  CMP      R3,#T12TBE ;DONE ALL TSTBLK TEST PATTERNS?
     BMIS     65: ;BR IF YES
     JMP      T123LOOP ;DO ANOTHER MODULO- 4 ADDRESS
     TST      T12KT ;DONE ABOVE 28K TESTING TOO?
     BGT      70: ;BR IF YES
     TST      KTFLG ;ANY MEMORY ABOVE 28K ON SYSTEM?
     BEQ      70: ;BR IF NO
     MOV      #1,T12KT ;SET SWITCH
     MOV      #T12BLK,R3 ;RESET TEST PATTERN TABLE
     JMP      T123LOOP ;DO ABOVE 28K TESTING
70:  JSR      PC,KTOFF ;TURN OFF KT11
     ENDSUB ;////////// END SUBTEST ////////////
                                     L10045:
                                     TRAP  C0ESUB
75:  TST      FATFLG ;ANY FATAL ERRORS ?
     BEQ      75: ;BRANCH IF NOT
     JSR      PC,CKDROP ;TRY TO DROP THE UNIT

```

.SBTTL TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

; TEST 3: SUBTEST 4:

; SUBTEST DESCRIPTION:

```

; This subtest verifies the NXM error bit in the TSSR
; register is set when attempting to fetch data (a characteristic
; data block) from selected nonexistent locations.
; If NXM fails to set it is likely that an LSI 11 Bus driver is
; failing to assert an address line.
; Addresses tested include all combinations of high-order address
; bits (i.e bits 16-21).
; *****

```

CAUTION

```

; The LSI BUS drivers for all available address lines(16-21)
; are only checked when running on a 11/238 system with more than
; 128K words of memory!
; *****

```

; TEST STEPS:

; BEGIN

```

; Write to TSSR to soft initialize
; Do a write characteristic command
; Invert the extended features switch
;

```

```

; REPEAT FOR SELECTED NON-EXISTENT MEMORY ADDRESSES

```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

4777      ;          BEGIN
4778      ;          Get an invalid test address
4779      ;          Set the test packet characteristics data pointer to the
4780      ;          test address.
4781      ;          Do a WRITE CHARACTERISTIC command
4782      ;          If TSSR register NXM bit not set then print error message
4783      ;          END
4784      ;          END
4785      ;          END
4786      ;          END
4787      027124      BGNSUB          ;//////////////// BEGIN SUBTEST //////////////////
      027124      T3.4:          TRAP      C#BSUB
      027124      104402

4788
4789
4790      027126      005737      003144      TST      T23A          ;26-APR-83 REV B - CHK FOR 23A CPU
4791      027132      001406      BEQ      5$          ;26-APR-83 REV B - BR, IF NOT 23A
4792      027134      023727      002120      007777      CMP      L#HIME,#7777      ;26-APR-83 REV B - CHK FOR > 256KB
4793      027142      103402      BLO      5$          ;26-APR-83 REV B - BR, IF < 256KB
4794      027144      000137      027536      JMP      NOEXTF      ;26-APR-83 REV B - JMP OVER 256KB
4795      027150      5$:          TST      NXMFLG          ;GOT ENOUGH MEMORY?
4796      027150      005737      003136      BNE      10$          ;IF SET STAY
4797      027154      001002      JMP      NOEXTF          ;LEAVE IF NOT SET
4798      027156      000137      027536
4799
4800      ;Write to TSSR to soft initialize
4801
4802      027162      004737      015774      10$:      JSR      PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
4803      027166      103405      BCS      11$          ;BR IF SOFT INIT = OK
4804      027170      NEXT.ERRNO
4805      027170      010001      MOV      R0,R1          ;SAVE CONTENTS OF TSSR
4806      027172      ERRDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL ERROR DURING INIT
      027172      104455      TRAP      C#ERDF
      027174      000465      .WORD      309
      027176      003652      .WORD      SFIERR
      027200      012034      .WORD      SFIMSG

4807
4808      ;Do a WRITE CHARACTERISTIC command so to invert switch
4809
4810      027202      104406      11$:      CKLOOP          ;LOOP IF SELECTED
      027202      012704      027610      MOV      #T12PACKET,R4      TRAP      C#CLP1
4811      027210      004737      031170      JSR      PC,T12SWRT      ;GET THE ADDRESS OF COMMAND PACKET
4812      027214      005037      003134      CLR      KTENABLE      ;RESTORE PACKET TO STARTING VALUES
4813      027220      010465      000000      MOV      R4,TSD8(R5)      ;TURN OFF KT-11
4814      027224      004737      016336      JSR      PC,CHKTSSR      ;SET THE PACKET ADDRESS
4815      027230      103405      FORCERROR      15$          ;WAIT FOR SSR TO SET
4816      027244      010001      BCS      17$          ;BR IF SSR SET IN CHKTSSR
4817      027246      010001      MOV      R0,R1          ;SAVE CONTENTS OF TSSR
4818      027250      NEXT.ERRNO
4819      027250      104455      15$:      ERRDF      ERRNO,T12WRTSSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      027252      000466      TRAP      C#ERDF
      027254      030122      .WORD      310
      027256      012046      .WORD      T12WRTSSR
      027260      104406      .WORD      PKTSSR

4821      027260      104406      17$:      CKLOOP          ;LOOP IF SELECTED
      027260      104406      TRAP      C#CLP1

```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

4822 027262 004737 020444          JSR      PC,INVERT          ;INVERT THE SWITCH
4823
4824          ;Get an invalid test address
4825
4826 027266 005037 002222      20$:   CLR      FATFLG          ;CLEAR FATAL ERROR FLAG
4827 027272          25$:
4828 027272 013737 003142 027652      MOV      NXMHI,T12HIADD      ;SAVE TEST ADDRESS HIGH
4829 027300 013737 003140 027654      MOV      NXML0,T12LOADD      ;SAVE TEST ADDRESS LOW
4830 027306          T124LOOP:
4831
4832          ;Set the test packet characteristics data pointer to the
4833          ; test address.
4834
4835 027306 012704 027610      30$:   MOV      @T12PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
4836 027312 004737 031170          JSR      PC,T12SWRT          ;RESTORE PACKET TO STARTING VALUES
4837 027316 013764 027654 000002      MOV      T12LOADD,PKLOW(R4)      ;STORE CHAR. DATA PTR LOW ADDRESS
4838 027324 013764 027652 000004      MOV      T12HIADD,PKHI(R4)      ;STORE CHAR. DATA PTR HIGH ADDRESS
4839
4840          ;Do a WRITE CHARACTERISTIC command
4841 027332 004737 017274          JSR      PC,KTOFF          ;TURN OFF KT-11
4842 027336 010465 000000          MOV      R4,TSD8(R5)          ;SET THE PACKET ADDRESS TO EXECUTE
4843 027342 004737 016250          JSR      PC,WAITF          ;WAIT FOR SSR TO SET
4844 027346          FORCERROR      32$
4845 027362 103407          BCS      40$          ;BR IF SSR SET IN CHKTSSR
4846 027364 010001          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
4847 027366          NEXT.ERRNO
4848 027366          32$:   ERROF      ERRNO,T12WRTSSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      311
                                .WORD      T12WRTSSR
                                .WORD      PKTSSR
                                TRAP      C$CLP1
                                .WORD      L10046-.
4849 027376 005237 002222          INC      FATFLG          ;SET FATAL ERROR FLAG
4850 027402          40$:   CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                .WORD      L10046-.
4851 027404          FORCERROR      45$,NOTSSR
4852 027414          ESCAPE      SUB          ;BY PASS SUBTEST IF FATAL ERROR
                                TRAP      C$ESCAPE
                                .WORD      L10046-.
4853          ;If TSSR register NXM bit not set then print error message
4854 027420          45$:
4855 027420 016501 000002      MOV      TSSR(R5),R1          ;GET TSSR CONTENTS
4856 027424          FORCERROR      52$
4857 027440 032701 004000      BIT      @NXM,R1          ;NXM SET?
4858 027444 001012          BNE      60$          ;BR IF YES
4859 027446          NEXT.ERRNO
4860 027446 013737 027654 002240      52$:   MOV      T12LOADD,ERRLO      ;MEMORY TEST ADDRESS LOW
4861 027454 013737 027652 002236      MOV      T12HIADD,ERRHI      ;MEMORY TEST ADDRESS HIGH
4862 027462          ERHRD      ERRNO,T12NXM,ADDSSR      ;REPORT ADDRESS AND TSSR ERROR
                                TRAP      C$ERHRD
                                .WORD      312
                                .WORD      T12NXM
                                .WORD      ADDSSR
4863
4864 027472          60$:   CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                .WORD      L10046-.
4865 027474          FORCEXIT      90$
4866 027504 005737 003144          TST      T23A          ;IS IT A 11/23A?

```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

4867 027516 001012      BNE      901
4868 027512 013700 027652      MOV      T12HIADD,RO
4869 027516 005200      651:    INC      RO
4870 027520 020027 000077      CMP      RO,#17
4871 027524 101004      BHI      901
4872 027526 010037 027652      751:    MOV      RO,T12HIADD
4873 027532 000137 027306      JMP      T124LOOP
4874 027536
4875 027536
4876 027536 004737 017274      901:    NOEXTF:  JSR      PC,KTOFF
4877 027542      ENDSUB
                                ;TURN OFF KT11
                                ;////////// END SUBTEST //////////
                                L10046:  TRAP      C$ESUB
                                ;ANY FATAL ERRORS ?
                                ;BRANCH IF NOT
                                ;TRY TO DROP THE UNIT
                                ;SHOULD WE DO ITERATIONS?
                                ;BR IF NO
                                ;LOOP UNTIL ITERATION COUNT DONE
                                ;TURN OFF MEMORY MANAGEMENT
                                ;CLEAR TEST FLAG
                                ;ALL DONE THIS TEST
                                TRAP      C$EXIT
                                .WORD    L10042-.

4878 027544 005737 002222      TST      FATFLG
4879 027550 001402      BEQ      1001
4880 027552 004737 017202      JSR      PC,CKDROP
4881 027556 004737 016456      1001:  JSR      PC,TSTLOOP
4882 027562 103002      BCC      1051
4883 027564 C00137 025776      JMP      T12LOOP
4884 027570
4885 027570 004737 017274      1051:  JSR      PC,KTOFF
4886 027574 005037 003150      CLR      T3BFLG
4887 027600      EXIT      TST
                                ;LOCAL STORAGE FOR THIS TEST
                                ;-
                                ;<..10>E177770
                                T12PACKET:
                                .WORD    100004
                                .WORD    T12DATA
                                .WORD    0
                                .WORD    8.
                                ;COMMAND PACKET FOR TEST
                                ;WRITE CHARACTERISTICS COMMAND, WITH ACK
                                ;ADDRESS OF CHARACTERISTICS BLOCK
                                ;STARTING VALUE OF BLOCK SIZE
                                ;CHARACTERISTICS DATA BLOCK
                                ;LOW ADDRESS OF MESSAGE BUFFER
                                ;HIGH ORDER OF MESSAGE BUFFER
                                ;LENGTH OF MESSAGE BUFFER
                                T12DATA:
                                .WORD    T12BFR
                                .WORD    0
                                .WORD    14.
                                .WORD    0.0
                                ;MESSAGE BUFFER
                                T12BFR: .BLKW 8.
                                ;HIGH ADDRESS
                                ;LOW ADDRESS
                                ;ADDRESS IN PAR FORMAT
                                ;TEST ABOVE 28K SWITCH
                                ;ADDRESS TEST BIT
                                ;*
                                ;
                                ;TABLE OF ADDRESSES
                                ;
                                ;-
                                T12BLK: .WORD 000001

4888
4889
4890
4891
4892
4893
4895      027610
4897 027610
4898 027610 100004
4899 027612 027620
4900 027614 000000
4901 027616 000010
4902
4903 027620
4904 027620 027632
4905 027622 000000
4906 027624 000016
4907 027626 000000 000000
4908
4909 027632
4910
4911 027652 000000
4912 027654 000000
4913 027656 000000
4914 027660 000000
4915 027662 000000
4916
4917
4918
4919
4920
4921 027664 000001

```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

4922	027666	000002	.WORD	000002
4923	027670	000003	.WORD	000003
4924	027672	000005	.WORD	000005
4925	027674	000006	.WORD	000006
4926	027676	000007	.WORD	000007
4927	027700	000011	.WORD	000011
4928	027702	000012	.WORD	000012
4929	027704	000013	.WORD	000013
4930	027706	000021	.WORD	000021
4931	027710	000022	.WORD	000022
4932	027712	000023	.WORD	000023
4933	027714	000041	.WORD	000041
4934	027716	000042	.WORD	000042
4935	027720	000043	.WORD	000043
4936	027722	000101	.WORD	000101
4937	027724	000102	.WORD	000102
4938	027726	000103	.WORD	000103
4939	027730	000201	.WORD	000201
4940	027732	000202	.WORD	000202
4941	027734	000203	.WORD	000203
4942	027736	000401	.WORD	000401
4943	027740	000402	.WORD	000402
4944	027742	000403	.WORD	000403
4945	027744	001001	.WORD	001001
4946	027746	001002	.WORD	001002
4947	027750	001003	.WORD	001003
4948	027752	002001	.WORD	002001
4949	027754	002002	.WORD	002002
4950	027756	002003	.WORD	002003
4951	027760	004001	.WORD	004001
4952	027762	004002	.WORD	004002
4953	027764	004003	.WORD	004003
4954	027766	010001	.WORD	010001
4955	027770	010002	.WORD	010002
4956	027772	010003	.WORD	010003
4957	027774	020001	.WORD	020001
4958	027776	020002	.WORD	020002
4959	030000	020003	.WORD	020003
4960	030002	040001	.WORD	040001
4961	030004	040002	.WORD	040002
4962	030006	040003	.WORD	040003
4963	030010	100001	.WORD	100001
4964	030012	100002	.WORD	100002
4965	030014	100003	.WORD	100003
4966	030016	177777	T12TBE: .WORD	177777

```

;+
;LOCAL TEXT MESSAGES FOR TEST
;-

```

4971	030020	104	115	101	TST12ID:	.ASCIZ	'DMA Memory Addressing'
4972	030046	103	157	156	T12GETSSR:	.ASCIZ	'Contents of TSSR Incorrect After GET STATUS'
4973	030122	103	157	156	T12WRTSSR:	.ASCIZ	'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
4974	030211	115	145	163	T12MSGBUF:	.ASCIZ	'Message Buffer Contents Incorrect After WRITE CHARACTERISTICS'
4975	030307	102	141	143	T12BKGD:	.ASCIZ	'Background Pattern Disturbed By WRITE CHARACTERISTICS'
4976	030375	105	170	160	T12NINT:	.ASCIZ	'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
4977	030466	127	162	151	T12DPR:	.ASCIZ	'Write Characteristic data in ram does not match expected'
4978	030557	124	123	123	T12NXM:	.ASCIZ	'TSSR NXM bit failed to set when non-existent memory address specifi

ed'

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

4979                                     .EVEN
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000
5001 030666
5002 030666
5003 030672 005037 027654
5004 030676 005037 027652
5005 030702 005037 027656
5006 030706 042701 170000
5007 030712 010005
5008 030714 004737 017274
5009 030720 013702 003124
5010 030724 062702 000020
5011 030730 060102
5012 030732 042702 000003
5013 030736 013703 003130
5014 030742 162703 000020
5015 030746 010237 027654
5016 030752 010237 027656
5017 030756 020203
5018 030760 101007
5019 030762 020237 003124
5020 030766 103007
5021 030770 005737 003134
5022 030774 001004
5023 030776 000424
5024 031000 162702 000020
5025 031004 000754
5026 031006
5027 031006 005737 003134
5028 031012 001420
5029 031014 005737 003132
5030 031020 001413
5031 031022 004737 017256
5032 031026 010500
5033 031030 010037 027652
5034 031034 010201
5035 031036 004737 017316

; 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
;
; OUTPUTS:
;   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
;
; T12CONVERT:
;   SAVREG
;   CLR T12LOADD
;   CLR T12HIADD
;   CLR T12PAR6
;   BIC #1C<7777>,R1
;   MOV R0,R5
;   JSR PC,KTOFF
;   MOV FREE,R2
;   ADD #16.,R2
;   ADD R1,R2
;   BIC #3,R2
;   25$: MOV FREEHI,R3
;   SUB #16.,R3
;   MOV R2,T12LOADD
;   MOV R2,T12PAR6
;   CMP R2,R3
;   BHI 35$
;   CMP R2,FREE
;   BHI 50$
;   TST KTENABLE
;   BNE 50$
;   BR 90$
;   35$: SUB #16.,R2
;   BR 25$
;   50$: TST KTENABLE
;   BEQ 100$
;   TST KTFLG
;   BEQ 90$
;   JSR PC,KTON
;   MOV R5,R0
;   MOV R0,T12HIADD
;   MOV R2,R1
;   JSR PC,SETMAP
;
;   ;SAVE R1-R5 UNTIL NEXT RETURN
;   ;CLEAR LOW ADDRESS
;   ;CLEAR HIGH ADDRESS
;   ;CLEAR PAR6 BIASED ADDRESS
;   ;FORCE TO LOWER 12 BITS OF ADDRESS
;   ;SAVE HIGH ORDER ADDRESS BITS
;   ;SHUTOFF MEMORY MANAGEMENT
;   ;GET FIRST FREE ADDRESS
;   ;IN CASE TEST PATTERN=0
;   ;ADD IN TEST PATTERN
;   ;MAKE IT MODULO-4
;   ;GET LAST FREE ADDRESS
;   ;SAVE AT LEAST 8 WORDS (IN CASE MESSAGE BUFFER)
;   ;SAVE POSSIBLE LOW ADDRESS
;   ;SAVE IT IN PAR6 BIASED TOO
;   ;IS THIS ADDRESS ABOVE FREE SPACE?
;   ;BR IF YES
;   ;IS IT IN FREE SPACE?
;   ;BR IF YES- ITS GOOD
;   ;TESTING ABOVE 28K?
;   ;BR IF YES
;   ;BR IF NOT IN FREE SPACE
;   ;FORCE FIT THE TEST PATTERN
;   ;TRY THIS TEST PATTERN ADDRESS
;   ;TESTING ABOVE 28K?
;   ;BR IF NO
;   ;ANY MEMORY ABOVE 28K?
;   ;BR IF NO
;   ;TURN ON MEMORY MANAGEMENT
;   ;GET HIGH ORDER ADDRESS
;   ;SAVE POSSIBLE HIGH ADDRESS
;   ;GET COMPUTED LOW ORDER ADDRESS
;   ;RETURN PAR6 BIASED ADDRESS IN R0

```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

5036 031042 010037 027656      MOV      R0,T12PAR6      ;COPY PAR6 BIASED ADDRESS
5037 031046 103403              BCS      105$      ;BR IF VALID ADDRESS
5038 031050 000241      90$:    CLC              ;CLR C BIT FOR FAILURE
5039 031052 000401              BR      105$      ;
5040 031054 000261      100$:   SEC              ;SET SUCCESS
5041 031056 000207      105$:   RTS      PC          ;RETURN
5042
5043
5044
5045
5046      ;
5047      ;ROUTINE TO READ THE FIRST 2 BYTES FROM RAM
5048      ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
5049      ;
5050      ;INPUT:
5051      ;
5052      ;      R4      ADDRESS OF THE COMMAND PACKET
5053      ;      R5      FIRST DEVICE UNIBUS ADDRESS
5054      ;
5055      ;OUTPUT:
5056      ;
5057      ;      CARRY   SET - RAM MATCHES PACKET
5058      ;              CLR - RAM DOES NOT MATCH PACKET
5059      ;
5060      ;IMPLICIT OUTPUT:
5061      ;
5062      ;      THE TABLE RAMDATA IS FILLED WITH THE
5063      ;      DATA HELD IN RAM.
5064      ;      RAMSIZ  SET TO 2 FOR PRAMPKT ROUTINE
5065      ;
5066      ;SIDE EFFECTS:
5067      ;
5068      ;      THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
5069      ;
5070      ;T12CKRAM::
5071      ;SAVREG
5072      ;MOV      @RAMDATA,R1      ;SAVE THE GENERAL REGISTERS
5073      ;MOV      @RMPKTBEGR,R2    ;ADDRESS TO SAVE THE RAM DATA
5074      ;CLR      R3              ;BYTE ADDRESS OF FIRST RAM DATA
5075      ;JSR      PC,CHKTSSR        ;CLEAR THE ERROR FLAG
5076      ;MOVB     @0,TSDB(R5)      ;WAIT FOR SSR
5077      ;JSR      PC,CHKTSSR        ;SET MAINTENANCE MODE
5078      ;MOV      R2,TSDB(R5)      ;WAIT FOR SSR TO SET
5079      ;JSR      PC,CHKTSSR        ;SELECT NEXT RAM ADDRESS
5080      ;MOVB     TSBA(R5),(R1)     ;WAIT FOR SSR TO SET
5081      ;CMPB     (R1)+,(R4)+       ;READ THE RAM DATA
5082      ;BEQ      20$              ;COMPARE TO EXPECTED
5083      ;INC      R3              ;BRANCH IF OK
5084      ;INC      R2              ;SET ERROR FLAG
5085      ;CMP      R2,@RMPKTBEGR+2   ;ADDRESS OF NEXT RAM LOCATION
5086      ;BLT      10$              ;DONE 2 BYTES?
5087      ;TST      R3              ;BR IF NO
5088      ;BEQ      30$              ;WAS AN ERROR FOUND ?
5089      ;CLC              ;BRANCH IF NOT
5090      ;BR      50$              ;CLEAR CARRY TO SHOW ERROR
5091      ;SEC              ;AND EXIT
5092      ;MOV      #2,RAMSIZ        ;SHOW GOOD COMPARE
5093      ;

```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

5093 031166 000207          RTS      PC              ;RETURN
5094
5095      ;*
5096      ;
5097      ;ROUTINE TO SETUP PACKET TO WRITE CHARACTERISTICS
5098      ;*
5099
5100      T12SWRT:
5101      SAVREG                      ;SAVE THE REGISTERS
5102      MOV      #T12PACKET,R1      ;START OF THE PACKET
5103      MOV      #100004,(R1)+      ;WRITE CHARACTERISTICS WITH ACK
5104      MOV      #T12DATA,(R1)+    ;ADDRESS OF CHAR DATA BLOCK
5105      CLR      (R1)+              ;EXTENDED ADDRESS
5106      MOV      #8,(R1)+          ;SIZE OF DATA BLOCK IN BYTES
5107      MOV      #T12BFR,(R1)+     ;ADDRESS OF MESSAGE BUFFER
5108      CLR      (R1)+
5109      MOV      #14,(R1)+          ;LENGTH OF MESSAGE BUFFER
5110      CLR      (R1)+
5111      CLR      (R1)
5112      RTS      PC              ;RETURN
5113
5114      ;*
5115      ;ROUTINE TO SETUP A GET STATUS COMMAND PACKET AT CURRENT PACKET ADDRESS
5116      ;
5117      R3      HIGH ORDER PACKET ADDRESS
5118      R4      LOW ORDER PACKET ADDRESS
5119      ;NOTE: R3 IS IGNORED IF KTENABLE FLAG CLEAR
5120      ;
5121      ;*
5122
5123      T12SETGET:
5124      SAVREG                      ;SAVE THE REGISTERS
5125      MOV      R4,R1              ;GET LOW ORDER ADDRESS
5126      TST      KTENABLE          ;TESTING ABOVE 28K?
5127      BEQ      10$              ;BR IF NO
5128      MOV      R3,R0              ;GET HIGH ORDER ADDRESS
5129      JSR      PC,SETMAP         ;RETURN ADDRESS BIASED TO PAR6 IN R0
5130      MOV      R0,R1              ;GET ADDRESS
5131      10$:    MOV      #P.GETSTATUS,R0 ;GET STATUS COMMAND CODE NO IE
5132      BIS      #P.ACK,R0          ;SET ACK
5133      MOV      R0,(R1)+           ;STORE GET STATUS IN PACKET
5134      CLR      (R1)+             ;CLEAR UNUSED WORD
5135      RTS      PC              ;RETURN
5136
5137
5138      ;*
5139      ;ROUTINE TO SETUP A CHARACTERISTIC DATA BLOCK AT A TEST ADDRESS
5140      ;
5141      ;*
5142
5143      T12CHAR:
5144      SAVREG                      ;SAVE R1-R5 UNTIL NEXT RETURN
5145      MOV      #T12DATA,R0        ;GET T12PACKET DATA POINTER
5146      MOV      T12LOAD,R1        ;ASSUME NOT ABOVE 28K
5147      TST      KTENABLE          ;TESTING ABOVE 28K?
5148      BEQ      10$              ;BR IF NO
5149      MOV      T12PAR6,R1        ;SET TEST ADDRESS ABOVE 28K

```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
5150 031326 012021          104:  MOV      (R0)+,(R1)+      ;STORE DATA WORD 1
5151 031330 012021          MOV      (R0)+,(R1)+      ;STORE DATA WORD 2
5152 031332 012021          MOV      (R0)+,(R1)+      ;STORE DATA WORD 3
5153 031334 012021          MOV      (R0)+,(R1)+      ;STORE DATA WORD 4
5154 031336 012021          MOV      (R0)+,(R1)+      ;STORE DATA WORD 5
5155 031340 000207          RTS      PC                ;RETURN
5156
5157 031342          ENDTST
```

```
L10042:  TRAP      C4ETST
```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

5159
5160
5161
5162
5163
5164
5165
5166
5167
5168 031344
      031344
5169
5170
5175 031344 005737 002214
5176 031350 001402
5177 031352 005237 003400
5178 031356 012700 034003
5179 031362 004737 016510
5180 031366 C12737 000005 002216
5181 031374
5182
5183
5184
5185
5186
5187
5188
5189
5190 031374
      031374
      031374 104402
5191 031376
      031376 012700 000000
      031402 104441
5192 031404 005737 003400
5193 031410 001402
5194 031412 000137 031674
5195 031416 004737 034022
5196 031422 004737 034074
5197 031426 004737 015774
5198 031432 103405
5202 031434 010001
5203 031436
      031436 104455
      031440 000621
      031442 003652
      031444 012034
5204 031446
5205 031446 012704 032720
5206 031452 004737 010662
5207 031456 103405
5211 031460 010001
5212 031462
      031462 104456
      031464 000622
      031466 005056
      031470 012034

      .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
      ;
      ;-
      BGNTST
      T4::
      TST TSTCNT ;CHECK FOR RUN MODE
      BEQ 10$ ;BR, IF NOT ONLY PROGRAM RUN
      INC SKIPT ;SET SKIP SW
      10$: MOV #TST15ID,R0 ;ASCII MESSAGE TO IDENTIFY TEST
      JSR PC,TSTSETUP ;DO INITIAL TEST SETUP
      MOV #5,LOOPCNT ;PERFORM 5 ITERATIONS
      T15LOOP:
      ;
      ; TEST 4, SUBTEST 1
      ;
      ; THIS SUBTEST WRITES THE ADDRESS (8 BITS) INTO THE
      ; RAM MEMORY SINGLE WORD (8 BITS) MODE
      ;
      ;-
      BGNSUB ;////////// BEGIN SUBTEST //////////
      T4.1:
      SETPRI #PRI00 ;LOWER PRIORITY TO ALLOW INTERRUPTS
      TRAP C1BSUB
      MOV #PRI00,R0
      TRAP C1SPRI
      TST SKIPT ;SHOULD WE SKIP THIS SUBTEST
      BEQ 10$ ;BR, IF NOW SKIP REQUIRED
      JMP 50$ ;SKIP SUBTEST
      10$: JSR PC,T15REST ;SET COMMAND PACKET
      JSR PC,T15RT2 ;SET UP OTHER COMMAND PACKET
      JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
      BCS 20$ ;BR IF INIT WAS OK
      MOV R0,R1 ;CONTENTS OF TSSR REGISTER
      ERDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      TRAP C1ERDF
      .WORD 401
      .WORD SFIERR
      .WORD SFIMSG
      20$: MOV #T15PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
      JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
      BCS 23$ ;BR, IF COMMAND ISSUED OK
      MOV R0,R1 ;SAVE CONTENTS OF TSSR
      ERHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      TRAP C1ERHRD
      .WORD 402
      .WORD WRTMSG
      .WORD SFIMSG

```

TEST 4: RAM EXERCISER TEST

5213	031472	012703	000400		234:	MOV	#256.,R3		;STARTING ADDRESS FOR RAM WRITE
5214	031476	112737	000001	033431		MOVB	#1,T15BS1		;SIZE OF TRANSFER
5215	031504	112737	000002	033430		MOVB	#2,T15BS0		;WRITE RAM "COMMAND"
5216	031512				254:				
5217	031512	010337	033432			MOV	R3,T15S2		;ADDRESS FOR RAM
5218	031516	012704	033420			MOV	#T15PK2,R4		;WRITE SUBSYS MEM PACKET
5219	031522	110337	033434			MOVB	R3,T15S3		;DATA FOR WRITE (ADDRESS)
5220	031526	010465	000000			MOV	R4,TSD8(R5)		;ISSUE COMMAND
5221	031532	004737	016336			JSR	PC,CHKTSSR		;WAIT FOR SSR
5222	031536	103407				BCS	304		;BR, IF NO ERROR
5223	031540	010001				MOV	R0,R1		;ERROR, SAVE TSSR
5227	031542					ERRHRD	ERRNO,T15SSR,PKTSSR		;TSSR NOT CORRECT AFTER WRITE SUB MEM
	031542	104456						TRAP	C#ERHRD
	031544	000623						.WORD	403
	031546	033436						.WORD	T15SSR
	031550	012046						.WORD	PKTSSR
5228	031552					ESCAPE	SUB		;DON'T CONTINUE IF ERROR ON WRITE
	031552	104410						TRAP	C#ESCAPE
	031554	C00122						.WORD	L10050-
5229	031556				304:	CKLOOP			;SCOPE LOOP
	031556	104406						TRAP	C#CLP1
5230									
5231									
5232	031560	005203				INC	R3		;NEXT ADDRESS
5233	031562	020327	010000			CMF	R3,#10000		;END OF RAM MEMORY CHECK
5234	031566	001351				BNE	254		;LOOP TILL ALL RAM WRITTEN
5235	031570	005002				CLR	R2		;CLEAR OUT R2 HIGH BITS
5236	031572	005303				DEC	R3		;SET BACK TO 7777
5237	031574	110337	033434		404:	MOVB	R3,T15S3		;GET DATA PATTERN BACK IN SHAPE
5238	031600	010337	033432			MOV	R3,T15S2		;ADDRESS FOR RAM READ
5239	031604	112737	000001	033430		MOVB	#1,T15BS0		;READ RAM COMMAND
5240	031612	010465	000000			MOV	R4,TSD8(R5)		;SEND OUT PACKET ADDRESS TO CONTR.
5241	031616	004737	016336			JSR	PC,CHKTSSR		;WAIT FOR READY, NON-AMBIGUOUS
5242	031622	103405				BCS	434		;BR, IF NO PROBLEM
5243	031624	010001				MOV	R0,R1		;SAVE TSSR
5247	031626					ERRDF	ERRNO,T15SSR,PKTSSR		;TSSR NOT CORRECT
	031626	104455						TRAP	C#ERDF
	031630	000624						.WORD	404
	031632	033436						.WORD	T15SSR
	031634	012046						.WORD	PKTSSR
5248	031636				434:	CKLOOP			;SCOPE LOOP
	031636	104406						TRAP	C#CLP1
5249	031640	013701	032762			MOV	T15BFR+20,R1		;GET RAM READ DATA
5250	031644	010302				MOV	R3,R2		;SET UP FOR COMPARE
5251	031646	120102				CMF	R1,R2		;CHECK WITH DATA WRITTEN
5252	031650	001404				BEQ	454		;BR IF OK, DATA IN = DATA OUT
5256	031652					ERRHRD	ERRNO,T15AM4,EXPBREC		;WRITTEN DATA NOT = TO READ
	031652	104456						TRAP	C#ERHRD
	031654	000625						.WORD	405
	031656	033715						.WORD	T15AM4
	031660	015502						.WORD	EXPBREC
5257	031662				454:	CKLOOP			;SCOPE LOOP
	031662	104406						TRAP	C#CLP1
5258	031664	005303				DEC	R3		;DROP DATA COUNTER (PATTERN)
5259	031666	020327	000377			CMF	R3,#255.		;AT BOTTOM YET
5260	031672	001340				BNE	404		;BR, IF MORE TO CHECK
5261	031674				504:	CKLOOP			;SCOPE LOOP

TEST 4: RAM EXERCISER TEST

```

5262 031674 104406          ENDSUB          ;////////////////// END SUBTEST ////////////////////
                                           L10050: TRAP      C#CLP1
031676                                     TRAP      C#ESUB
031676 104403
5263
5264
5265 031700          BGNSUB          ;////////////////// BEGIN SUBTEST ////////////////////
031700                                     T4.2: TRAP      C#BSUB
031700 104402
5266
5267
5268
5269
5270
5271
5272
5273
5274
5275 031702 C04737 034022          JSR      PC,T15REST          ;RESTORE PACKET FOR WRITE CHARA
5276 031706 004737 034074          JSR      PC,T15RT2          ;RESTORE PACKET FOR WRT SUB SYS MEM
5277 031712 004737 015774          JSR      PC,SOFINIT          ;DO INITIALIZE ON CONTROLLER
5278 031716 103405          BCS      20#          ;BR IF INIT WAS OK
5282 031720 010001          MOV      R0,R1          ;CONTENTS OF TSSR REGISTER
5283 031722          ERDIF          ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
                                           TRAP      C#ERDIF
                                           .WORD      406
                                           .WORD      SFIERR
                                           .WORD      SFIMSG
031722 104455
031724 000626
031726 003652
031730 012034
5284 031732          20#: MOV      #T15PACKET,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
5285 031732 012704 032720          JSR      PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
5286 031736 004737 010662          BCS      25#          ;BR, IF COMMAND ISSUED OK
5287 031742 103405          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
5291 031744 010001          ERHRD          ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
5292 031746          25#: TRAP      C#ERHRD
031746 104456                                           .WORD      407
031750 000627                                           .WORD      WRTMSG
031752 005056                                           .WORD      SFIMSG
031754 012034
5293 031756          25#: MOV      #1,T15S1          ;SET SIZE OF TRANSFER 1 BYTE
5294 031756 112737 000001 033431          MOV      #T15PK2,R4          ;SET NEW PACKET ADDRESS
5295 031764 012704 033420          MOV      #256,R3          ;STARTING ADDRESS IN RAM
5296 031770 012703 000400          MOV      #2,T15S0          ;WRITE RAM COMMAND
5297 031774 112737 000002 033430          CLRB          T15S3          ;SET DATA TO 000
5298 032002 105037 033434          MOV      R3,T15S2          ;ADDRESS TO PACKET DATA AREA
5299 032006 010337 033432          30#: MOV      R4,TSD8(R5)          ;SEND OUT PACKET ADDRESS
5300 032012 010465 000000          JSR      PC,CHKTSSR          ;WAIT FOR SSR
5301 032016 004737 016336          BCS      33#          ;BR, IF NO PROBLEM
5302 032022 103405          MOV      R0,R1          ;SAVE TSSR
5303 032024 010001          ERHRD          ERRNO,T15SSR,PKTSSR ;TSSR NOT CORRECT
5307 032026          33#: TRAP      C#ERHRD
032026 104456                                           .WORD      408
032030 000630                                           .WORD      T15SSR
032032 033436                                           .WORD      PKTSSR
032034 012046
5308 032036          33#: CKLOOP          ;SCOPE LOOP
032036 104406          TRAP      C#CLP1
5309

```

TEST 4: RAM EXERCISER TEST

Address	Offset	Hex	Label	Operation	Comments	Trap	Trap Data
5310	032040	005203		INC R3	;NEXT ADDRESS		
5312	032042	020327	010000	CMP R3,#10000	;END OF RAM MEMORY CHECK		
5313	032046	001357		BNE 30#	;BR, MORE RAM TO GO		
5314	032050	005303		DEC R3	;SET BACK TO 7777		
5315	032052	005002		CLR R2	;SET TO ALL ZEROS		
5316	032054	112737	000001 033430	MOVB #1,T15B50	;READ RAM COMMAND		
5317	032062	010337	033432	MOV R3,T15S2	;ADDRESS TO BE READ TO PACKET DATA		
5318	032066	010465	000000	MOV R4,TSD8(R5)	;SEND OUT PACKET ADDRESS		
5319	032072	004737	016336	JSR PC,CHKTSSR	;WAIT FOR SSR TO SET		
5320	032076	103405		BCS 41#	;BR, IF ALL IS WELL		
5321	032100	010001		MOV R0,R1	;SAVE TSSR		
5325	032102			ERRHRD	ERRNO,T15SSR,PKTSSR		
	032102	104456				TRAP	C#ERHRD
	032104	000631				.WORD	409
	032106	033436				.WORD	T15SSR
	032110	012046				.WORD	PKTSSR
5326	032112			41#:	CKLOOP		
	032112	104406					
5327	032114	013701	032762		MOV T15BFR+20,R1		
5328	032120	120102			CMPB R1,R2		
5329	032122	001404			BEQ 42#		
5333	032124				ERRHRD	ERRNO,T15AM3,EXPBREC	
	032124	104456					
	032126	000632					
	032130	033613					
	032132	015502					
5334	032134			42#:	CKLOOP		
	032134	104406					
5335	032136	012702	000377		MOV #000377,R2		
5336	032142	112737	000002 033430		MOVB #2,T15B50		
5337	032150	112737	000377 033434		MOVB #000377,T15S3		
5338	032156	010465	000000		MOV R4,TSD8(R5)		
5339	032162	004737	016336		JSR PC,CHKTSSR		
5340	032166	103405			BCS 43#		
5341	032170	010001			MOV R0,R1		
5345	032172				ERRHRD	ERRNO,T15SSR,PKTSSR	
	032172	104456					
	032174	000633					
	032176	033436					
	032200	012046					
5346	032202			43#:	CKLOOP		
	032202	104406					
5347	032204	112737	000001 033430		MOVB #1,T15B50		
5348	032212	010465	000000		MOV R4,TSD8(R5)		
5349	032216	004737	016336		JSR PC,CHKTSSR		
5350	032222	103405			BCS 44#		
5351	032224	010001			MOV R0,R1		
5355	032226				ERRDF	ERRNO,T15SSR,PKTSSR	
	032226	104455					
	032230	000634					
	032232	033436					
	032234	012046					
5356	032236	013701	032762		44#:	MOV T15BFR+20,R1	
5357	032242	120102			CMPB R1,R2		
5358	032244	001404					

TEST 4: RAM EXERCISER TEST

032246	104456				TRAP	C1ERHRD
032250	000635				.WORD	413
032252	033512				.WORD	T15AM2
032254	015502				.WORD	EXPBREC
5363	032256			451:	CKLOOP	;SCOPE LOOP
032256	104406					TRAP
5364	032260	005303			DEC	R3
5365	032262	020327	000377		CHP	R3, #255.
5366	032266	001271			BNE	401
5367						
5368	032270				ENDSUB	;////////// END SUBTEST //////////
032270						L10051:
032270	104403					TRAP
5369						C1ESUB
5370	032272				BGNSUB	;////////// BEGIN SUBTEST //////////
032272						T4.3:
032272	104402					TRAP
5371						C1BSUB
5372						
5373						
5374						
5375						
5376						
5377						
5378						
5379	032274	005737	003400		TST	SKIPT
5380	032300	001402			BEG	101
5381	032302	000137	032676		JMP	501
5382	032306	004737	034022	101:	JSR	PC.T15REST
5383	032312	004737	034074		JSR	PC.T15RT2
5384	032316	004737	015774		JSR	PC.SOFINIT
5385	032322	103405			BCS	201
5389	032324	010001			MOV	R0,R1
5390	032326				ERRDF	ERRNO,SFIERR,SFIMSG
032326	104455					
032330	000636					TRAP
032332	003652					.WORD
032334	012034					.WORD
5391	032336					.WORD
5392	032336	012704	032720	201:	MOV	@T15PACKET,R4
5393	032342	004737	010662		JSR	PC.WRTCHR
5394	032346	103405			BCS	251
5398	032350	010001			MOV	R0,R1
5399	032352				ERRHRD	ERRNO,WRTMSG,SFIMSG
032352	104456					
032354	000637					TRAP
032356	005056					.WORD
032360	012034					.WORD
5400	032362					.WORD
5401	032362	112737	000001	033431	251:	MOV
5402	032370	012704	033420			@1.T158S1
5403	032374	012703	000400			@T15PK2,R4
5404	032400	112737	000002	033430		MOV
5405	032406	112737	000377	033434		@256,R3
5406	032414	010337	033432			MOV
5407	032420	010465	000000		301:	@2.T158S0
5408	032424	004737	016336			MOV
						@377.T15S3
						MOV
						R3.T15S2
						MOV
						R4.T508(R5)
						JSR
						PC.CKTTSSR
						;SET SIZE TO 1 BYTE
						;SET NEW PACKET ADDRESS
						;STARTING ADDRESS IN RAM
						;WRITE RAM COMMAND
						;SET DATA TO 377
						;ADDRESS TO PACKET DATA AREA
						;SEND OUT PACKET ADDRESS
						;WAIT FOR SSR

TEST 4: RAM EXERCISER TEST

5409	032430	103405			BCS	33:		;BR, IF NO PROBLEM		
5410	032432	010001			MOV		R0,R1	;SAVE TSSR		
5414	032434				ERRHRD		ERRNO,T15SSR,PKTSSR	;TSSR NOT CORRECT		
	032434	104456							TRAP	C#ERHRD
	032436	000640							.WORD	416
	032440	033436							.WORD	T15SSR
	032442	012046							.WORD	PKTSSR
5415	032444				33:	CKLOOP		;SCOPE LOOP	TRAP	C#CLP1
	032444	104406								
5416										
5417										
5418	032446	005203			INC		R3	;NEXT ADDRESS		
5419	032450	020327	010000		CMP		R3,#10000	;END OF RAM MEMORY CHECK		
5420	032454	001357			BNE		30:	;BR, MORE RAM TO GO		
5421	032456	005303			DEC		R3	;SET BACK TO 7777		
5422	032460	112702	000377		35:	MOVB	#377,R2	;SET TO ALL ONES		
5423	032464	112737	000001	033430	40:	MOVB	#1,T15B50	;READ RAM COMMAND		
5424	032472	010337	033432		MOV		R3,T15S2	;ADDRESS TO BE READ TO PACKET DATA		
5425	032476	C10465	000000		MOV		R4,TSD8(R5)	;SEND OUT PACKET ADDRESS		
5426	032502	004737	016336		JSR		PC,CHKTSSR	;WAIT FOR SSR TO SET		
5427	032506	103405			BCS		41:	;BR, IF ALL IS WELL		
5428	032510	010001			MOV		R0,R1	;SAVE TSSR		
5432	032512				ERRHRD		ERRNO,T15SSR,PKTSSR	;TSSR NOT CORRECT		
	032512	104456							TRAP	C#ERHRD
	032514	000641							.WORD	417
	032516	033436							.WORD	T15SSR
	032520	012046							.WORD	PKTSSR
5433	032522				41:	CKLOOP		;SCOPE LOOP	TRAP	C#CLP1
	032522	104406								
5434	032524	013701	032762		MOV		T15BFR+20,R1	;PICK UP READ DATA		
5435	032530	120102			CMPB		R1,R2	;BOTH SHOULD BE 11111111 BINARY		
5436	032532	001404			BEQ		42:	;BR, IF DATA IS GOOD		
5440	032534				ERRHRD		ERRNO,T15AM3,EXPBREC	;CHARACTERISTICS DATA NOT CORRECT		
	032534	104456							TRAP	C#ERHRD
	032536	000642							.WORD	418
	032540	033613							.WORD	T15AM3
	032542	015502							.WORD	EXPBREC
5441	032544	012702	000377		42:	MOV	#000377,R2	;SET ALL ONES WORD		
5442	032550	012737	000002	033430	MOV		#2,T15B50	;WRITE RAM COMMAND		
5443	032556	112737	000377	033434	MOVB		#000377,T15S3	;ALL ONES PATTERN		
5444	032564	010465	000000		MOV		R4,TSD8(R5)	;PASS PACKET ADDRESS TO CONTR.		
5445	032570	004737	016336		JSR		PC,CHKTSSR	;WAIT FOR SSR		
5446	032574	103405			BCS		43:	;BR, IF OK (NO ERROR)		
5447	032576	010001			MOV		R0,R1	;SAVE TSSR		
5451	032600				ERRHRD		ERRNO,T15SSR,PKTSSR	;TSSR NOT CORRECT		
	032600	104456							TRAP	C#ERHRD
	032602	000643							.WORD	419
	032604	033436							.WORD	T15SSR
	032606	012046							.WORD	PKTSSR
5452	032610				43:	CKLOOP		;SCOPE LOOP	TRAP	C#CLP1
	032610	104406								
5453	032612	112737	000001	033430	MOVB		#1,T15B50	;SET UP FOR RAM READ		
5454	032620	010465	000000		MOV		R4,TSD8(R5)	;ISSUE RAM READ		
5455	032624	004737	016336		JSR		PC,CHKTSSR	;WAIT FOR SSR TO SET		
5456	032630	103405			BCS		44:	;BR, IF OK (NO ERROR)		
5457	032632	010001			MOV		R0,R1	;SAVE TSSR		
5461	032634				ERRHRD		ERRNO,T15SSR,PKTSSR	;TSSR NOT CORRECT		

TEST 4: RAM EXERCISER TEST

032634	104456				TRAP	C1ERMRD
032636	000644				.WORD	420
032640	033436				.WORD	T15SSR
032642	012046				.WORD	PKTSSR
5462	032644	013701	032762	441:	MOV	T15BFR+20,R1
5463	032650	120102			CMPB	R1,R2
5464	032652	001404			BEQ	451
5468	032654				ERRMRD	ERRNO,T15AM2,EXPBREC
	032654	104456				
	032656	000645			TRAP	C1ERMRD
	032660	033512			.WORD	421
	032662	015502			.WORD	T15AM2
					.WORD	EXPBREC
5469	032664			451:	CKLOOP	
	032664	104406				
5470	032666	005303			DEC	R3
5471	032670	020327	000377		CMP	R3,#255.
5472	032674	001271			BNE	401
5473						
5474	032676			501:		
5475	032676				ENDSUB	
	032676					
	032676	104403				
5476						
5477	032700	004737	016456		JSR	PC,TSTLOOP
5478	032704	103002			BCC	631
5479	032706	000137	031374		JMP	T15LOOP
5480	032712			631:	EXIT	TST
	032712	104432				
	032714	001216				
5481						
5482						
5483						
5484						
5486		032720				
5488	032720					
5489	032720	100204			.WORD	100204
5490	032722	032730			.WORD	T15DATA
5491	032724	000000			.WORD	0
5492	032726	000010			.WORD	8.
5493	032730					
5494	032730	032742			.WORD	T15BFR
5495	032732	000000			.WORD	0
5496	032734	000400			.WORD	256.
5497	032736	000000	000000		.WORD	0,0
5498	032742				T15BFR:	.BLKW 150.
5499						
5500						
5501						
5503		033420				
5505	033420					
5506	033420	100206			.WORD	100206
5507	033422	033430			.WORD	T15BF2
5508	033424	000000			.WORD	0
5509	033426	000006			.WORD	6.
5510						
5511					.EVEN	
5512	033430				T15BF2:	

;LOCAL STORAGE FOR THIS TEST
 ;
 ;*
 T15PACKET: .<..10>&177770
 ;COMMAND PACKET FOR TEST
 ;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
 ;ADDRESS OF CHARACTERISTICS BLOCK
 ;STARTING VALUE OF BLOCK SIZE
 ;CHARACTERISTICS DATA BLOCK
 ;ADDRESS OF MESSAGE BUFFER
 ;LENGTH OF MESSAGE BUFFER
 ;MESSAGE BUFFER
 ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
 ;
 T15PK2: .<..10>&177770
 ;WRITE SUB SYS MEM COMMAND, IE AND ACK
 ;ADDRESS OF SELECT BLOCK DATA
 ;SIZE OF DATA PACKET
 ;
 T15BF2:

TEST 4: RAM EXERCISER TEST

```

5513 033430      000      T15B50:  BYTE  0      ;BSEL0 AREA
5514 033431      000      T15B51:  .BYTE  0      ;BSEL1 AREA
5515 033432 000000      T15S2:   .WORD  0      ;SEL 2 AREA
5516 033434 000000      T15S3:   .WORD  0      ;DATA AREA
5517
5518
5519
5520
5521
5522
5523
5524
5525 033436      127      122      111  T15SSR:  .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
5526 033512      127      122      111  T15AM2:  .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Ones Word Read Back'
5527 033613      127      122      111  T15AM3:  .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Zeros Word Read Back'
5528 033715      127      122      111  T15AM4:  .ASCIZ  'WRITE SUBSYSTEM MEMORY COMMAND Failed On Address Test'
5529 034003      122      101      115  T15ID:   .ASCIZ  'RAM Exerciser'
5530
5531
5532
5533
5534
5535
5536
5537
5538 034022
5539 034022
5540 034026 012701 032720
5541 034032 012721 100204
5542 034036 012721 032730
5543 034042 005021
5544 034044 012721 000010
5545 034050 012721 032742
5546 034054 005021
5547 034056 012721 000400
5548 034062 005021
5549 034064 005011
5550 034066 005037 032742
5551 034072 000207
5552
5553
5554 034074
5555 034074
5556 034100 012701 033420
5557 034104 012721 100206
5558 034110 012721 033430
5559 034114 005021
5560 034116 012721 000006
5561 034122 005021
5562 034124 005021
5563 034126 005011
5564 034130 000207
5565 034132
      034132
      034132 104401

;*
;LOCAL TEXT MESSAGES FOR TEST
;-

T15REST:
      SAVREG
      MOV      @T15PACKET,R1      ;SAVE THE REGISTERS
      MOV      @100204,(R1).      ;START OF THE PACKET
      MOV      @T15DATA,(R1).      ;WRITE SUBSYSTEM MEM. WITH ACK, IE
      CLR      (R1).              ;ADDRESS OF CHARAISTICS DATA BLOCK
      MOV      @8,(R1).           ;EXTENDED ADDRESS
      MOV      @T15BFR,(R1).      ;SIZE OF DATA BLOCK IN BYTES
      CLR      (R1).              ;ADDRESS OF MESSAGE BUFFER
      MOV      @256,(R1).         ;LENGTH OF MESSAGE BUFFER
      CLR      (R1).
      CLR      (R1)
      CLR      T15BFR             ;CLEAR 1ST LOC IN MESSAGE BUFFER
      RTS      PC                 ;RETURN

T15RT2:
      SAVREG
      MOV      @T15PK2,R1         ;SAVE THE REGISTERS
      MOV      @100206,(R1).      ;START OF THE PACKET
      MOV      @T15BF2,(R1).      ;WRITE SUBSYSTEM MEM. WITH ACK, IE
      CLR      (R1).              ;ADDRESS OF DATA BLOCK
      MOV      @6,(R1).           ;EXTENDED ADDRESS
      CLR      (R1).              ;SIZE OF DATA BLOCK IN BYTES
      CLR      (R1).
      CLR      (R1)
      RTS      PC                 ;RETURN
      ENDTST

L10047:  TRAP      C#ETST

```

TEST 5: EXTENDED FEATURES SWITCH AND TIMERS A,B

```

5567 .SBTTL TEST 5: EXTENDED FEATURES SWITCH AND TIMERS A,B
5568
5569 ;**
5570 ; TEST DESCRIPTION:
5571 ;
5572 ; This test verifies the Invert Extended Features function
5573 ; can logically invert the Extended features switch and
5574 ; that the internal timers A and B operate correctly.
5575 ;
5576 ; TEST STEPS:
5577 ;
5578 ; REPEAT FOR LOOPCNT
5579 ; BEGIN
5580 ; Do Subtest 1 Verify Extended Features Switch
5581 ; Do Subtest 2 - Verify Timers A,B
5582 ; END
5583 ;--
5584
5585 034134 BGNTST
5586 034134
5590 034134 012700 036212 MOV #TST16ID,R0 T5::
5591 034140 004737 016510 JSR PC,TSTSETUP ;ASCII MESSAGE TO IDENTIFY TEST
5592 034144 012737 000012 002216 MOV #10.,LOOPCNT ;DO INITIAL TEST SETUP
5593 034152 T16LOOP: ;PERFORM 10 ITERATIONS
5594
5595 .SBTTL TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST
5596
5597 ;**
5598 ; TEST 5: SUBTEST 1:
5599 ;
5600 ; SUBTEST DESCRIPTION:
5601 ;
5602 ; This subtest verifies that the Invert Sense of Extended features
5603 ; Switch function (Write Subsystem Memory,Write Misc command)
5604 ; operates properly.
5605 ; First the state of the Extended Features switch is read in the
5606 ; message packet supplied by the write characteristics command.
5607 ; Then, the sense of the switch is logically inverted.
5608 ; A Write characteristics command is executed and it is verified
5609 ; that the Extended status register (XST4) is returned when
5610 ; in Extended mode, and not returned if not in extended mode.
5611 ; The subtest also verifies that specifying a Message Buffer
5612 ; address with any of bits 21-19 .set will cause the command to
5613 ; be rejected.
5614 ;
5615 ; TEST STEPS:
5616 ;
5617 ; BEGIN
5618 ; Write to TSSR register to soft initialize the controller
5619 ; Do WRITE CHARACTERISTICS to check for Extended Features Switch
5620 ; IF Extended Features Hardware Switch CLEAR
5621 ; THEN
5622 ; (* Verify Extended Features switch can be Inverted to SET *)
5623 ; Do Write Subsystem Write Miscellaneous to SET Extended Features.
5624 ; DO a WRITE CHARACTERISTICS with an extended characteristic word
5625 ; Compare the controller ram to the extended characteristic word
5626 ;

```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```

5627      ;      If Data word in controller ram NOT= to word sent Then Print Error
5628      ;      If Message Buffer Data Length NOT= 12. Then Print Error
5629      ;      ELSE
5630      ;      (* Verify Extended Features switch can be Inverted to CLEAR *)
5631      ;      Do Write Subsystem Write Miscellaneous to CLEAR Extended Features.
5632      ;      Do a WRITE CHARACTERISTICS without an extended characteristic word
5633      ;      If Message Buffer Data Length NOT= 10. Then Print Error
5634      ;      END-IF
5635      ;      (* Verify Function Reject when Message Buffer 21-19 are non-zero *)
5636      ;      Write to TSSR register to soft initialize the controller
5637      ;      REPEAT FOR MESSAGE BUFFER ADDRESS bits <21:19> FROM 0 TO 7
5638      ;      DO a WRITE CHARACTERISTICS with a message address bit<21:19> non-zero
5639      ;      If TSSR termination code NOT= Function Reject Then Print Error
5640      ;      END-REPEAT
5641      ;      END
5642      ;      --
5643      034152      BGNSUB                      ;////////// BEGIN SUBTEST //////////
5644      034152      104402                      T5.1:      TRAP      C#BSUB
5645
5646      034154      5#:
5647      ;      Write to TSSR register to soft initialize the controller
5648      034154      004737      015774      JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
5649      034160      103405      BCS      10#      ;BR IF SOFT INIT OKAY
5650      034162      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
5651      034164      ERRDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
5652      034164      104455      TRAP      C#ERDF
5653      034166      000764      .WORD      500
5654      034170      003652      .WORD      SFIERR
5655      034172      012034      .WORD      SFIMSG
5656
5657      ;      Do WRITE CHARACTERISTICS to check for Extended Features Switch
5658      10#:      JSR      PC,T16REST      ;RESTORE PACKET DEFAULTS
5659      034200      005037      002222      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
5660      034204      012704      037540      MOV      #T16PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
5661      034210      004737      010662      JSR      PC,WRTHR      ;DO WRITE CHARACTERISTICS COMMAND
5662      034214      FORCERROR      12#      ;GOODFORCE ERROR IF FORCER=1
5663      034230      103407      BCS      15#      ;BR IF CARRY SET (GOOD RETURN)
5664      034232      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
5665      034234      NEXT.ERRNO
5666      12#:      ERRDF      ERRNO,T16SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
5667      034234      104455      TRAP      C#ERDF
5668      034236      000765      .WORD      501
5669      034240      036262      .WORD      T16SSR
5670      034242      012046      .WORD      PKTSSR
5671      034244      005237      002222      15#:      INC      FATFLG      ;SET FATAL ERROR FLAG
5672      034250      034250      104406      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
5673      034250      104406      TRAP      C#CLP1
5674
5675      ;      If Extended Features Hardware Switch Clear then:
5676      ;      (* Verify Extended Features switch can be Inverted to SET *)
5677      ;      REPEAT FOR TEST PATTERNS IN TSTBLK TABLE
5678      034252      012701      037562      MOV      #T16BFR,R1      ;MESSAGE BUFFER ADDRESS
5679      034256      032761      000200      000012      BIT      #X2.EXTF,XST2(R1)      ;EXTENDED FEATURES SWITCH CLEAR?
5680      034264      001402      BEQ      20#      ;BR IF YES
5681      034266      000137      034636      JMP      200#      ;
5682      034272      012703      002764      20#:      MOV      #TSTBLK+10.,R3      ;START OF TEST DATA

```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```

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

```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```

5715 034532 103404      BCS      100$      ;BR IF YES
5716 034534      NEXT.ERRNO
5717 034534      92$:      ERRHRD  ERRNO,PKTRAM,RAMERR      ;REPORT THE RAM ERROR(S)
                                TRAP      C$ERHRD
                                .WORD     505
                                .WORD     PKTRAM
                                .WORD     RAMERR
5718 034544      104456
5719 034544      000771
5720 034546      012702 037562      ;
5721 034552      016201 000002      ; If Message Buffer Data Length NOT= 12. Then Print Error
5722 034556      012702 000014      ; MOV      #T16BFR,R2      ;GET MESSAGE BUFFER ADDRESS
5723 034562      020102      ; MOV      2(R2),R1      ;GET RECV DATA FIELD LENGTH
5724 034572      001404      ; MOV      #12.,R2      ;GET EXPD DATA FIELD LENGTH
5725 034574      ; FORCERROR 112$,NOTSSR      ;GOODFORCE ERROR IF FORCER=1
5726 034576      ; CMP      R1,R2      ;COMPARE EXPECTED TO RECEIVED
5727 034576      112$:      BEQ      120$      ;ERROR IF NOT EQUAL
                                ERRHRD  ERRNO,T16LEN,EXPREC      ;PRINT THE ERROR & EXPD/RECV
                                TRAP      C$ERHRD
                                .WORD     506
                                .WORD     T16LEN
                                .WORD     EXPREC
5728 034606      104406      120$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
5729 034610      004737 015774      ;
5730 034614      103405      JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
5731 034616      010001      BCS      125$      ;BR IF SOFT INIT OKAY
5732 034620      ; MOV      R0,R1      ;SAVE CONTENTS OF TSSR
5733 034620      104455      ERRDF  ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
5734 034630      104406      ;
5735 034632      000137 035016      ; TRAP      C$ERDF
5736 034636      ;
5737 034636      ; (* Verify Extended Features switch can be Inverted to CLEAR *)
5738 034636      200$:      ;
5739 034636      ; Do Write Subsystem Write Miscellaneous to CLEAR Extended Features.
5740 034636      004737 037520      JSR      PC,T16SEXT      ;SETUP PACKET FOR WRITE MISC INVERT
5741 034642      012704 037610      MOV      #T16PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
5742 034646      010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
5743 034652      004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
5744 034656      103407      FORCERROR 232$      ;GOODFORCE ERROR IF FORCER=1
5745 034672      010001      BCS      240$      ;BR IF CARRY SET (GOOD RETURN)
5746 034674      ; MOV      R0,R1      ;SAVE CONTENTS OF TSSR
5747 034676      ; NEXT.ERRNO
5748 034676      232$:      ERRDF  ERRNO,T162SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
5749 034706      005237 002222      ;
5750 034712      104406      ; TRAP      C$ERDF
5751 034712      ;
5751 034712      240$:      INC      FATFLG      ;SET FATAL ERROR FLAG
                                CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1

```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```

5752      ; DO a WRITE CHARACTERISTICS without an extended characteristic word
5753 034714 012704 037540      MOV    #T16PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
5754 034720 012764 000016 000006  MOV    #14.,PKBCNT(R4)      ;STORE MESSAGE PACKET SIZE
5755 034726 004737 010662      JSR     PC,WRCHR          ;DO WRITE CHARACTERISTICS COMMAND
5756 034732      FORCERROR 242$      ;DO FORCE ERROR IF FORCER=1
5757 034746 103407      BCS     250$      ;BR IF CARRY SET (GOOD RETURN)
5758 034750 010001      MOV     R0,R1      ;SAVE CONTENTS OF TSSR
5759 034752      NEXT.ERRNO
5760 034752 242$: ERRDF  ERRNO,T16SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      ; TRAP C#ERDF
      ; .WORD 508
      ; .WORD T16SSR
      ; .WORD PKTSSR
      ;
5761 034762 005237 002222      INC     FATFLG      ;SET FATAL ERROR FLAG
5762 034766 104406 250$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      ; TRAP C#CLP1
      ;
5763      ; If Message Buffer Data Length NOT= 10. Then Print Error
5764 034770 013701 037564      MOV     T16BFR+2,R1      ;GET RECV DATA FIELD LENGTH
5765 034774 012702 000012      MOV     #10.,R2      ;GET EXPD DATA FIELD LENGTH
5766 035000 020102      CMP      R1,R2      ;COMPARE EXPECTED TO RECEIVED
5767 035002 001404      BEQ      270$      ;ERROR IF NOT EQUAL
5768 035004      NEXT.ERRNO
5769 035004 262$: ERRHRD  ERRNO,T16LEN,EXPREC      ;PRINT THE ERROR & EXPD/RECV
      ; TRAP C#ERHRD
      ; .WORD 509
      ; .WORD T16LEN
      ; .WORD EXPREC
      ;
5770 035014 104406 270$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      ; TRAP C#CLP1
      ;
5771
5772
5773      ; (* Verify Function Reject when Message Buffer 21-19 are non-zero *)
5774      ; Write to TSSR register to soft initialize the controller
5775 035016 300$: REPEAT FOR MESSAGE BUFFER ADDRESS bits <21:19> FROM 0 TO 7
5776 035016 012737 000001 002312 320$: MOV     #1,DATA      ;START AT BITS<21:19>=001
5777      ; DO a WRITE CHARACTERISTICS with a message address bit<21:19> non-zero
5778 035024 325$: MOV     #T16PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
5779 035024 012704 037540      MOV     #14.,PKBCNT(R4)      ;STORE MESSAGE PACKET SIZE
5780 035030 012764 000016 000006  MOV     DATA,R0      ;GET TEST DATA
5781 035036 013700 002312      .REPT    3
5782      ; ASL      R0      ;SHIFT INTO BITS 21:19
5783      ; .ENDR
5784      ; MOV     R0,T16DATA+2      ;STORE BUFFER ADDRESS BITS 21:19
5785      ; MOV     R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
5786 035050 010037 037552      JSR     PC,WAITF      ;WAIT FOR SSR
5787 035054 010465 000000      FORCERROR 342$      ;DO FORCE ERROR IF FORCER=1
5788 035060 004737 016250      BCS     350$      ;BR IF CARRY SET (GOOD RETURN)
5789 035064      MOV     R0,R1      ;SAVE CONTENTS OF TSSR
5790 035100 103407
5791 035102 010001      NEXT.ERRNO
5792 035104 342$: ERRDF  ERRNO,T16SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      ; TRAP C#ERDF
      ; .WORD 510
      ; .WORD T16SSR
      ; .WORD PKTSSR
      ;
5793 035104 104455
5794 035106 000776
5795 035110 036262
5796 035112 012046
5797 035114 005237 002222      INC     FATFLG      ;SET FATAL ERROR FLAG

```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```

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

```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

5844      ;      (* Verify Timers A,B after RESET TIMER with 0 microsecond delay *)
5845      ;      Do a Write Control RESET TIMER with 1 microsecond delay
5846      ;      Do a Write Subsystem READ STATUS
5847      ;      If Timer A NOT= 0 Then Print Error
5848      ;      If Timer B NOT= 0 Then Print Error
5849      ;      (* Verify Timers A,B after RESET TIMER with 28 microsecond delay *)
5850      ;      Do a Write Control RESET TIMER with 28 microsecond delay
5851      ;      If Timer A NOT= 1 Then Print Error
5852      ;      If Timer B NOT= 1 Then Print Error
5853      ;      Do a Write Control RESET TIMER with 53 microsecond delay
5854      ;      If Timer A NOT= 0 Then Print Error
5855      ;      If Timer B NOT= 1 Then Print Error
5856      ;      Do a Write Control RESET TIMER with 78 microsecond delay
5857      ;      If Timer A NOT= 1 Then Print Error
5858      ;      If Timer B NOT= 0 Then Print Error
5859      ;--
5860 035232      BGNSUB                      ;////////// BEGIN SUBTEST ///////////
      035232      T5.2:                      TRAP      C#BSUB
      035232 104402
5861      ;      Write to TSSR register to soft initialize the controller
5862 035234      5#:
5863 035234 004737 015774      JSR      PC,SOFINIT          ;WRITE TO TSSR TO SOFT INITIALIZE
5864 035240 103405      BCS      10#                      ;BR IF SOFT INIT OKAY
5865 035242 010001      MOV      R0,R1                      ;SAVE CONTENTS OF TSSR
5866 035244      ERRDF      ERRNO,SFIERR,SFIMSG          ;DEVICE FATAL DURING INIT
      035244 104455      TRAP      C#ERDF
      035246 000777      .WORD      511
      035250 003652      .WORD      SFIERR
      035252 012034      .WORD      SFIMSG
5867      ;      Do WRITE CHARACTERISTICS to setup a Message Buffer
5868 035254 004737 037360      10#:      JSR      PC,T16REST          ;RESTORE PACKET DEFAULTS
5869 035260 005037 002222      CLR      FATFLG          ;CLEAR FATAL ERROR FLAG
5870 035264 012704 037540      MOV      #T16PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
5871 035270 012764 000010 000006      MOV      #8.,PKBCNT(R4) ;MESSAGE PACKET SIZE NO EXTEND
5872 035276 004737 010662      JSR      PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
5873 035302      FORCERROR      12#                      ;BDDFORCE ERROR IF FORCER=1
5874 035316 103407      BCS      15#                      ;BR IF CARRY SET (GOOD RETURN)
5875 035320 010001      MOV      R0,R1                      ;SAVE CONTENTS OF TSSR
5876 035322      NEXT.ERRNO
5877 035322      12#:      ERRDF      ERRNO,T16SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      035322 104455      TRAP      C#ERDF
      035324 001000      .WORD      312
      035326 036262      .WORD      T16SSR
      035330 012046      .WORD      PKTSSR
5878 035332 005237 002222      15#:      INC      FATFLG          ;SET FATAL ERROR FLAG
5879 035336      CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      035336 104406      TRAP      C#CLP1
5880      ;
5881      ;      (* Verify Timers A,B after RESET TIMER with 1 microsecond delay *)
5882      ;      Do a Write Control RESET TIMER with 1 microsecond delay
5883 035340 012700 000001      MOV      #MS.RSD,R0          ;RESET TIMER COMMAND
5884 035344 013701 036202      MOV      T16U01,R1          ;1 MICROSECOND DELAY
5885 035350 004737 037472      JSR      PC,T16WMISC          ;SETUP T16PK2 COMMAND PACKET
5886 035354 012704 037610      MOV      #T16PK2,R4          ;GET WRITE SUBSYSTEM COMMAND PACKET
5887 035360 010465 000000      MOV      R4,TSDB(R5)          ;SET THE PACKET ADDRESS TO EXECUTE
5888 035364 004737 016336      JSR      PC,CHKTSSR          ;WAIT FOR SSR TO SET
5889 035370      FORCERROR      32#                      ;BDDFORCE ERROR IF FORCER=1

```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

5890 035404 103407      BCS      40$      ;BR IF CARRY SET (GOOD RETURN)
5891 035406 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
5892 035410              NEXT.ERRNO
5893 035410              ERRDF   ERRNO,T162SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     513
                                .WORD     T162SSR
                                .WORD     PKTSSR
5894 035420 005237 002222 40$: INC      FATFLG      ;SET FATAL ERROR FLAG
5895 035424 104406      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
5896              ; If Timer A NOT= 0 Then Print Error
5897              ; If Timer B NOT= 0 Then Print Error
5898 035426 005002      CLR      R2      ;INIT EXPD
5899 035430 042702 000010 BIC      #S2.ATIM,R2      ;TIMER A EXPD=0
5900 035434 042702 000004 BIC      #S2.BTIM,R2      ;TIMER B EXPD=0
5901 035440 012700 037602 MOV      #T16BFSTA,R0      ;GET RECV READ STATUS
5902 035444 016001 000002 MOV      2(R0),R1      ;GET RECV BYTE 2
5903 035450 C42701 177763 BIC      #C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
5904 035454          FORCERROR 72$,NOTSSR ;880
5905 035464 020201      CMP      R2,R1      ;EXPD EQUAL RECV?
5906 035466 001404      BEQ      80$      ;BR IF YES
5907 035470          NEXT.ERRNO
5908 035470              72$: ERRHRD   ERRNO,T16T01,TIMEXP ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD     514
                                .WORD     T16T01
                                .WORD     TIMEXP
5909 035500              80$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
5910              ; Do a Write Control RESET TIMER with 28 microsecond delay
5911              MOV      #MS.RSD,R0      ;RESET TIMER COMMAND
5912 035502 012700 000001 MOV      T16D28,R1      ;28 MICROSECOND DELAY
5913 035506 013701 036204 JSR      PC,T16WHISC      ;SETUP T16PK2 COMMAND PACKET
5914 035512 004737 037472 MOV      #T16PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
5915 035516 012704 037610 MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
5916 035522 010465 000000 JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
5917 035526 004737 016336 FORCERROR 112$ ;880FORCE ERROR IF FORCER=1
5918 035532          BCS      120$      ;BR IF CARRY SET (GOOD RETURN)
5919 035546 103407      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
5920 035550 010001      NEXT.ERRNO
5921 035552          112$: ERRDF   ERRNO,T162SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     515
                                .WORD     T162SSR
                                .WORD     PKTSSR
5922 035552              120$: INC      FATFLG      ;SET FATAL ERROR FLAG
5923 035562 005237 002222 CKLOOP      ;LOOP ON ERROR, IF FLAG SET
5924 035566 104406      TRAP      C$CLP1
5925              ; If Timer A NOT= 1 Then Print Error
5926              ; If Timer B NOT= 1 Then Print Error
5927 035570 005002      CLR      R2      ;INIT EXPD
5928 035572 052702 000010 BIS      #S2.ATIM,R2      ;TIMER A EXPD=1
5929 035576 052702 000004 BIS      #S2.BTIM,R2      ;TIMER B EXPD=1
5930 035602 012700 037602 MOV      #T16BFSTA,R0      ;GET RECV READ STATUS
5931 035606 016001 000002 MOV      2(R0),R1      ;GET RECV BYTE 2

```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

5932 035612 042701 177763      BIC      #C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
5933 035616                      FORCERROR      1724,NOTSSR ;000
5934 035626 020201      CMP      R2,R1 ;EXPD EQUAL RECV?
5935 035630 001404      BEQ      1804 ;BR IF YES
5936 035632                      NEXT.ERRNO
5937 035632 1724:      ERRHRD  ERRNO,T16T28,TIMEXP ;REPORT ERROR
                                TRAP      CIERHRD
                                .WORD     516
                                .WORD     T16T28
                                .WORD     TIMEXP
                                035632 104456
                                035634 001004
                                035636 037060
                                035640 015552
5938 035642 1804:      CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                035642 104406                      TRAP      CCLP1
5939
5940      ; Do a Write Control RESET TIMER with 53 microsecond delay
5941 035644 012700 000001      MOV      #MS.RSD,R0 ;RESET TIMER COMMAND
5942 035650 013701 036206      MOV      T16D53,R1 ;53 MICROSECOND DELAY
5943 035654 004737 037472      JSR      PC,T16WMISC ;SETUP T16PK2 COMMAND PACKET
5944 035660 012704 037610      MOV      #T16PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
5945 035664 C10465 000000      MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
5946 035670 004737 016336      JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
5947 035674                      FORCERROR      2124 ;000FORCE ERROR IF FORCER=1
5948 035710 103407      BCS      2204 ;BR IF CARRY SET (GOOD RETURN)
5949 035712 010001      MOV      R0,R1 ;SAVE CONTENTS OF TSSR
5950 035714                      NEXT.ERRNO
5951 035714 2124:      ERRDF  ERRNO,T162SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      CIERDF
                                .WORD     517
                                .WORD     T162SSR
                                .WORD     PKTSSR
                                035714 104455
                                035716 001005
                                035720 036317
                                035722 012046
5952 035724 005237 002222      INC      FATFLG ;SET FATAL ERROR FLAG
5953 035730 2204:      CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                035730 104406                      TRAP      CCLP1
5954      ; If Timer A NOT= 0 Then Print Error
5955      ; If Timer B NOT= 1 Then Print Error
5956 035732 005002      CLR      R2 ;INIT EXPD
5957 035734 042702 000010      BIC      #S2.ATIM,R2 ;TIMER A EXPD=0
5958 035740 052702 000004      BIS      #S2.BTIM,R2 ;TIMER B EXPD=1
5959 035744 012700 037602      MOV      #T16BFSTA,R0 ;GET RECV READ STATUS
5960 035750 016001 000002      MOV      2(R0),R1 ;GET RECV BYTE 2
5961 035754 042701 177763      BIC      #C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
5962 035760                      FORCERROR      2724,NOTSSR ;000
5963 035770 020201      CMP      R2,R1 ;EXPD EQUAL RECV?
5964 035772 001404      BEQ      2804 ;BR IF YES
5965 035774                      NEXT.ERRNO
5966 035774 2724:      ERRHRD  ERRNO,T16T53,TIMEXP ;REPORT ERROR
                                TRAP      LIERHRD
                                .WORD     518
                                .WORD     T16T53
                                .WORD     TIMEXP
                                035774 104456
                                035776 001006
                                036000 037160
                                036002 015552
5967 036004 2804:      CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                036004 104406                      TRAP      CCLP1
5968      ; Do a Write Control RESET TIMER with 78 microsecond delay
5969 036006 012700 000001      MOV      #MS.RSD,R0 ;RESET TIMER COMMAND
5970 036012 013701 036210      MOV      T16D78,R1 ;78 MICROSECOND DELAY
5971 036016 004737 037472      JSR      PC,T16WMISC ;SETUP T16PK2 COMMAND PACKET
5972 036022 012704 037610      MOV      #T16PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
5973 036026 010465 000000      MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE

```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

5974 036032 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
5975 036036      FORCERROR 312;      ;GOODFORCE ERROR IF FORCER=1
5976 036052 103407      BCS      320;      ;BR IF CARRY SET (GOOD RETURN)
5977 036054 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
5978 036056      NEXT.ERRNO
5979 036056      312;:  ERRDF  ERRNO,T162SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      036056 104455      TRAP      C#ERRDF
      036060 001007      .WORD    519
      036062 036317      .WORD    T162SSR
      036064 012046      .WORD    PKTSSR
5980 036066 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
5981 036072 104406      320;:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      036072      TRAP      C#CLP1
5982      ;
5983      ; If Timer A NOT= 1 Then Print Error
5984 036074 005002      CLR      R2      ;INIT EXPD
5985 036076 052702 000010      BIS      #S2.ATIM,R2      ;TIMER A EXPD=1
5986 036102 042702 000004      BIC      #S2.BTIM,R2      ;TIMER B EXPD=0
5987 036106 C12700 037602      MOV      #T16BFSTA,R0      ;GET RECV READ STATUS
5988 036112 016001 000002      MOV      2(R0),R1      ;GET RECV BYTE 2
5989 036116 042701 177763      BIC      #C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
5990 036122      FORCERROR 372;:NOTSSR ;BBD
5991 036132 020201      CMP      R2,R1      ;EXPD EQUAL RECV?
5992 036134 001404      BEQ      380;      ;BR IF YES
5993 036136      NEXT.ERRNO
5994 036136      372;:  ERRMRD  ERRNO,T16T78,TIMEXP ;REPORT ERROR
      036136 104456      TRAP      C#ERRMRD
      036140 001010      .WORD    520
      036142 037260      .WORD    T16T78
      036144 015552      .WORD    TIMEXP
5995 036146 1;4406      380;:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      036146      TRAP      C#CLP1
5996      ;
5997 036150      ENDSUB      ;////////// END SUBTEST //////////
      036150      L10055:
      036150 104403      TRAP      C#ESUB
5998      ;
5999 036152 005737 002222      TST      FATFLG      ;ANY FATAL ERRORS ?
6000 036156 001402      BEQ      460;      ;BRANCH IF NOT
6001 036160 004737 017202      JSR      PC,CKDROP
6002 036164 004737 016456      460;:  JSR      PC,TSTLOOP
6003 036170 103002      BCC      465;      ;SHOULD WE DO ITERATIONS?
6004 036172 000137 034152      JMP      T16LOOP      ;BR IF NO
6005 036176      465;:
6006      ;
6007      ;
6008 036176      EXIT      TST      ;////////// EXIT TEST //////////
      036176 104432      TRAP      C#EXIT
      036200 001524      .WORD    L10053-.
6009      ;
6010      ;
6011      ;
6012      ; LOCAL STORAGE FOR THIS TEST
6013      ;
6014 036202 000001      T16D01:      .WORD    1      ;1 MICROSECOND DELAY (ACTUALLY .8 MIC)
6015 036204 000040      T16D28:      .WORD    40      ;28 MICROSECOND DELAY (.8 MICROS PER)
6016 036206 000076      T16D53:      .WORD    76      ;53 MICROSECOND

```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

6017 036210 000142      T16D78:      .WORD    142      ;78 MICROSECOND
6018
6019                    ;*
6020                    ;LOCAL TEXT MESSAGES FOR TEST
6021                    ;-
6022 036212      105      170      164  TST16ID:      .ASCIZ  'Extended Features Switch and Timers A,B'
6023 036262      127      122      111  T16SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
6024 036317      127      122      111  T162SSR: .ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
6025 036363      127      122      111  T163SSR: .ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
6026 036430      102      165      163  T16TSBA: .ASCIZ  'Bus Address Register (TSBA) Incorrect after Write Characteristics'
6027 036532      104      141      164  T16LEN: .ASCIZ  'Data Field Length in Message Buffer Incorrect after Write Characteristics'
6028 036644      124      123      123  T16REJ: .ASCIZ  'TSSR Function Reject Not Returned When Non-Existent Buffer Address Specified'
d' 6029 036761      124      151      155  T16T01: .ASCIZ  'Timer A,B Incorrect after Reset Timer with 1 microsecond Delay'
6030 037060      124      151      155  T16T28: .ASCIZ  'Timer A,B Incorrect after Reset Timer with 28 microsecond Delay'
6031 037160      124      151      155  T16T53: .ASCIZ  'Timer A,B Incorrect after Reset Timer with 53 microsecond Delay'
6032 037260      124      151      155  T16T78: .ASCIZ  'Timer A,B Incorrect after Reset Timer with 78 microsecond Delay'
6033
6034
6035                    ;*
6036                    ;SET DEFAULT PACKET
6037                    ;-
6038 037360
6039 037360      012700      037540      T16REST:      MOV      #T16PACKET,R0      ;PACKET ADDRESS
6040 037364      012720      100004      MOV      #100004,(R0)+      ;WRITE CHARACTERISTICS WITH ACK
6041 037370      012720      037550      MOV      #T16DATA,(R0)+      ;ADDRESS OF CHAR DATA BLOCK
6042 037374      005020      CLR      (R0)+      ;EXTENDED ADDRESS
6043 037376      012720      000012      MOV      #10,(R0)+      ;SIZE OF MESSAGE PACKET
6044 037402      012720      037562      MOV      #T16BFR,(R0)+      ;MESSAGE BUFFER ADDRESS
6045 037406      005020      CLR      (R0)+      ;CLEAR EXTENDED BUFFER ADDRESS
6046 037410      012720      000024      MOV      #20,(R0)+      ;LENGTH OF MESSAGE BUFFER
6047 037414      005020      CLR      (R0)+      ;CLEAR ESS,ENB,EAI,ERI
6048 037416      005010      CLR      (R0)      ;CLEAR EXTENDED FEATURES WORD
6049 037420      005037      037562      CLR      T16BFR      ;CLEAR 1ST LOCATION IN MESSAGE BUFFER
6050 037424      000207      RTS      PC
6051
6052
6053                    ;*
6054                    ;CLEAR MESSAGE BUFFER
6055                    ;-
6056 037426
6057 037426      012701      037562      T16CLRBUF:      SAVREG      ;SAVE R1-R5 UNTIL NEXT RETURN
6058 037436      012702      000026      MOV      #T16BFR,R1      ;GET MESSAGE BUFFER ADDRESS
6059 037442      105021      MOV      #T16BEND-T16BFR,R2      ;SIZE OF MESSAGE BUFFER IN BYTES
6060 037444      005302      101:      CLRB      (R1)+      ;CLEAR A BYTE
6061 037446      003375      DEC      R2      ;DONE?
6062 037450      000207      BGT      101      ;BR IF NO
6063
6064
6065                    ;*
6066                    ;SETUP T16PK2 PACKET FOR READ STATUS
6067                    ;-
6068 037452      004737      037426      T16SRD:      JSR      PC,T16CLRBUF      ;CLEAR MESSAGE BUFFER
6069 037456      012700      037620      MOV      #T16D*2,R0      ;WRITE SUBSYSTEM DATA BUFFER
6070 037462      112720      000005      MOV      #PW.RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
6071 037466      105010      CLRB      (R0)      ;CLEAR BSEL1
6072 037470      000207      RTS      PC      ;RETURN
6073

```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

6074
6075
6076      ;* SETUP T16PK2 PACKET FOR WRITE MISC.
6077      ;
6078      ; INPUT:
6079      ;      R0      CONTAINS WRITE MISC FUNCTION CODE (BSEL1)
6080      ;      R1      CONTAINS DELAY (TIMES 800 NS) FOR BSEL2
6081      ;
6082      T16WMISC:
6083      SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
6084      JSR      PC,T16CLRBUF                ;CLEAR MESSAGE BUFFER
6085      MOV      #T16DT2,R2                 ;WRITE SUBSYSTEM DATA BUFFER
6086      MOV      #PW.WMISC,(R2)+            ;STORE WRITE MISCELLANEOUS IN BSEL0
6087      MOV      R0,(R2)+                   ;STORE WRITE MISC CODE IN BSEL1
6088      MOV      R1,(R2)+                   ;STORE DELAY (RESET TIMER) IN BSEL2
6089      RTS      PC                         ;RETURN
6090
6091      ;* SETUP T16PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
6092      ;
6093      T16SEXT:
6094      MOV      #T16DT2,R0                 ;WRITE SUBSYSTEM DATA BUFFER
6095      MOV      #PW.WMISC,(R0)+            ;STORE WRITE MISCELLANEOUS IN BSEL0
6096      MOV      #MS.EXT,(R0)+             ;STORE INVERT EXTENDED FEATURES IN BSEL1
6097      RTS      PC                         ;RETURN
6098
6099
6100
6101
6102      .=<..10>E177770
6103      ;
6104      ;WRITE CHARACTERISTICS COMMAND PACKET
6105      ;
6106      T16PACKET:
6107      .WORD     100004                     ;COMMAND PACKET FOR TEST
6108      .WORD     T16DATA                   ;WRITE CHARACTERISTICS COMMAND, WITH ACK
6109      .WORD     0                          ;ADDRESS OF CHARACTERISTICS BLOCK
6110      .WORD     10.                       ;MESSAGE PACKET 'LE
6111
6112      T16DATA:
6113      .WORD     T16BFR                     ;CHARACTERISTICS DATA BLOCK
6114      .WORD     0                          ;ADDRESS OF MESSAGE BUFFER
6115      .WORD     20.                       ;LENGTH OF MESSAGE BUFFER
6116      .WORD     0                          ;ESS,ENB,EAI,ERI
6117      .WORD     0                          ;EXTENDED FEATURES WORD
6118
6119      ;MESSAGE BUFFER
6120
6121      T16BFR:
6122      .WORD     0                          ;BEGIN MESSAGE BUFFER
6123      .WORD     0                          ;MESSAGE TYPE
6124      .WORD     0                          ;DATA FIELD LENGTH
6125      .WORD     0                          ;R8PCR
6126      .WORD     0                          ;XST0
6127      .WORD     0                          ;XST1
6128      .WORD     0                          ;XST2
6129      .WORD     0                          ;XST3
6130      .WORD     0                          ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
6131
6132

```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

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

L10053: TRAP C:ETST

TEST 6: FIFO EXERCISER

```

6156 .SBTTL TEST 6: FIFO EXERCISER
6157 ;**
6158 ; TEST DESCRIPTION:
6159 ;
6160 ; This test uses the Write Subsystem Memory command to
6161 ; verify the controller's FIFO and associated status and
6162 ; control logic.
6163 ;
6164 ; TEST STEPS:
6165 ;
6166 ; REPEAT FOR LOOPCNT
6167 ; BEGIN
6168 ; Do Subtest 1 - FIFO Initialize status test
6169 ; Do Subtest 2 - FIFO Write Single Byte test
6170 ; Do Subtest 3 - FIFO Write Multiple Bytes test
6171 ; Do Subtest 4 - FIFO Verify ILW Status test
6172 ; Do Subtest 5 - FIFO Input Ready test
6173 ; Do Subtest 6 - FIFO Verify Reset FIFO test
6174 ; END
6175 ;--
6176
6177
6178 037726 BGNTST
6179 037726
6183 037726 012700 046156 MOV #TST17ID,RO ;ASCII MESSAGE TO IDENTIFY TEST
6184 037732 004737 016510 JSR PC,TSTSETUP ;DO INITIAL TEST SETUP
6185 037736 012737 000012 MOV #10,LOOPCNT ;PERFORM 10 ITERATIONS
6186 037744 004737 017274 JSR PC,KTOFF ;SHUT OFF MEMORY MANAGEMENT
6187 037750 005037 003134 CLR KTENABLE ;REALLY SHUT DOWN KT-11
6188 037754 T17LOOP:
6189
6190
6191 .SBTTL TEST 6: SUBTEST 1: FIFO INITIALIZE STATUS TEST
6192 ;**
6193 ; TEST 6: SUBTEST 1:
6194 ;
6195 ; SUBTEST DESCRIPTION:
6196 ;
6197 ; This test verifies, by using the Read Status select code,
6198 ; that the FIFO status is in the correct initial state after
6199 ; the controller is initialized (Input Ready TRUE,
6200 ; Output Ready and Data In Miss FALSE). These status
6201 ; signals are checked by the controller's self-test
6202 ; sequence, so this subtest is actually more of a partial
6203 ; check of the Read Status function than the FIFO status.
6204 ;
6205 ; TEST STEPS:
6206 ;
6207 ; BEGIN
6208 ; Write to TSSR to soft initialize
6209 ; Do a WRITE CHARACTERISTICS to setup a message buffer
6210 ; Do a WRITE SUBSYSTEM Read Status
6211 ; If Input Ready NOT=1 Then Print Error
6212 ; If Output Ready NOT=0 Then Print Error
6213 ; If Data In Miss NOT=0 Then Print Error
6214 ;
6215 ; END

```

TEST 6: SUBTEST 1: FIFO INITIALIZE STATUS TEST

```

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

```

TEST 6: SUBTEST 1: FIFO INITIALIZE STATUS TEST

```

6257 040156 042711 000200      BIC      #S2.DIM,(R1)      ;SET EXP DATA IN MISS = FALSE
6258                          ; If Input Ready NOT=1 then Print Error
6259                          ; If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
6260 040162 005000      CLR      R0      ;HIGH RECV ADDRESS FOR CKMSG2
6261 040164 012701 047572      MOV      #T17BFR,R1      ;LOW RECV ADDRESS FOR CKMSG2
6262 040170 012702 045732      MOV      #T17EXP,R2      ;EXPL ADDRESS
6263 040174 012703 000024      MOV      #20.,R3      ;NUMBER OF BYTES TO COMPARE
6264 040200 004737 011500      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
6265 040204      FORCERROR      82$,NOTSSR      ;880
6266 040214 103404      BCS      90$      ;BR IF YES
6267 040216      NEXT.ERRNO
6268      82$:      ERRHRD      ERRNO,T171CMP,MSGSTAT      ;REPORT ERROR
                          TRAP      C$ERHRD
                          .WORD      603
                          .WORD      T171CMP
                          .WORD      MSGSTAT
6269 040226      90$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                          TRAP      C$CLP1
6270      040226 104406
6271 040230      ENDSUB      ;////////// END SUBTEST //////////
                          L10057:
                          TRAP      C$ESUB
6272      040230 104403
6273 040232 005737 002222      TST      FATFLG      ;ANY FATAL ERRORS ?
6274 040236 001402      BEQ      160$      ;BRANCH IF NOT
6275 040240 004737 017202      JSR      PC,CKDROP      ;TRY TO DROP THE UNIT
6276 040244      160$:

```

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

```

6277      ;**
6278      ; TEST 6: SUBTEST 2:
6279      ;
6280      ; SUBTEST DESCRIPTION:
6281      ;
6282      ; This subtest verifies the ability of the FIFO to correctly
6283      ; pass a single data byte from input to output. For each
6284      ; of 256 data values (0-377 octal) the following is done:
6285      ; 1. Initial FIFO status is checked
6286      ; 2. The Write FIFO function, specifying a count of
6287      ;    one byte to be written is executed.
6288      ; 3. Read Status is executed and FIFO status is checked.
6289      ; 4. Read FIFO is executed and the data and final status
6290      ;    is checked.
6291      ;
6292      ; TEST STEPS:
6293      ;
6294      ; BEGIN
6295      ; Write to TSSR to soft initialize
6296      ; Do a WRITE CHARACTERISTICS to setup a message buffer
6297      ; Do a Write Subsystem READ STATUS
6298      ; If Input Ready NOT=1 Then Print Error
6299      ; If Output Ready NOT=0 Then Print Error
6300      ; If Data In Miss NOT=0 Then Print Error
6301      ;
6302      ; REPEAT FOR DATA FROM 0 TO 377 OCTAL
6303      ; BEGIN
6304      ;
6305      ;
6306      ;

```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```

6307      ; Do a Write Subsystem WRITE NPR to set tape direction out
6308      ; Do a Write Subsystem WRITE FIFO with byte count equal to 1
6309      ; Do a Write Subsystem READ STATUS
6310      ; If Input Ready NOT=1 Then Print Error
6311      ; If Output Ready NOT=1 Then Print Error
6312      ; If Data In Miss NOT=0 Then Print Error
6313      ; Do Write Subsystem READ FIFO with byte count equal to 1
6314      ; If Data read from FIFO NOT= to Data sent Then Print Error
6315      ; Do a Write Subsystem READ STATUS
6316      ; If Input Ready NOT=1 Then Print Error
6317      ; If Output Ready NOT=0 Then Print Error
6318      ; If Data In Miss NOT=0 Then Print Error
6319      ; END
6320      ; END
6321      ; --
6322      040244      BGNSUB                      ;////////// BEGIN SUBTEST //////////
        040244      T6.2:                      TRAP      C#BSUB
        040244      104402

6323      ;
6324      5$: Write to TSSR register to soft initialize the controller
6325      040246      JSR      PC,SOFINIT          ;WRITE TO TSSR TO SOFT INITIALIZE
6326      040246      BCS      10$                ;BR IF SOFT INIT OKAY
6327      040252      MOV      R0,R1              ;SAVE CONTENTS OF TSSR
6328      040254      ERRDF   ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
6329      040256      TRAP      C#ERDF
        040256      104455      .WORD      603
        040260      001133      .WORD      SFIERR
        040262      003652      .WORD      SFIMSG
        040264      012034

6330      ; Do a WRITE CHARACTERISTICS to setup a message buffer
6331      040266      10$: CLR      FATFLG          ;CLEAR FATAL ERROR FLAG
6332      040272      MOV      #T17PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
6333      040276      JSR      PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
6334      040302      FORCERROR 42$              ;BDFORCE ERROR IF FORCER=1
6335      040316      BCS      50$                ;BR IF CARRY SET (GOOD RETURN)
6336      040320      MOV      R0,R1              ;SAVE CONTENTS OF TSSR
6337      040322      NEXT.ERRNO
6338      040322      42$: ERRDF   ERRNO,T17SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
        040322      104455      TRAP      C#ERDF
        040324      001134      .WORD      604
        040326      046175      .WORD      T17SSR
        040330      012046      .WORD      PKTSSR

6339      040332      005237      10$: INC      FATFLG          ;SET FATAL ERROR FLAG
6340      040336      50$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
        040336      104406      TRAP      C#CLP1

6341      ; Do a Write Subsystem READ STATUS
6342      040340      JSR      PC,T17SRD          ;SETUP PACKET FOR READ STATUS
6343      040344      MOV      #T17PK2,R4        ;GET WRITE SUBSYSTEM COMMAND PACKET
6344      040350      MOV      R4,TSD8(R5)        ;SET THE PACKET ADDRESS TO EXECUTE
6345      040354      JSR      PC,CHKTSSR        ;WAIT FOR SSR TO SET
6346      040360      FORCERROR 62$              ;BDFORCE ERROR IF FORCER=1
6347      040374      BCS      70$                ;BR IF CARRY SET (GOOD RETURN)
6348      040376      MOV      R0,R1              ;SAVE CONTENTS OF TSSR
6349      040400      NEXT.ERRNO
6350      040400      62$: ERRDF   ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
        040400      104455      TRAP      C#ERDF
        040402      001135      .WORD      605

```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

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

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```

6395 040624 012704 047720      MOV    #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
6396 040630 010465 000000      MOV    R4,TSD8(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
6397 040634 004737 016336      JSR     PC,CHKTSSR    ;WAIT FOR SSR TO SET
6398 040640          FORCERROR 107#    ;B0DFORCE ERROR IF FORCER=1
6399 040654 103407          BCS     11C#    ;BR IF CARRY SET (GOOD RETURN)
6400 040656 010001          MOV     R0,R1    ;SAVE CONTENTS OF TSSR
6401 040660          NEXT.ERRNO
6402          107#:  ERRDF  ERRNO,T175SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C#ERDF
                                .WORD     608
                                .WORD     T175SSR
                                .WORD     PKTSSR
6403 040670 005237 002222      110#:  INC     FATFLG    ;SET FATAL ERROR FLAG
6404 040674          CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
6405 040674 104406
6406          ;      Do a Write Subsystem READ STATUS
6407 040676 004737 047334      JSR     PC,T17SRD    ;SETUP PACKET FOR READ STATUS
6408 040702 C12704 047720      MOV    #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
6409 040706 010465 000000      MOV    R4,TSD8(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
6410 040712 004737 016336      JSR     PC,CHKTSSR    ;WAIT FOR SSR TO SET
6411 040716          FORCERROR 112#    ;B0DFORCE ERROR IF FORCER=1
6412 040732 103407          BCS     120#    ;BR IF CARRY SET (GOOD RETURN)
6413 040734 010001          MOV     R0,R1    ;SAVE CONTENTS OF TSSR
6414 040736          NEXT.ERRNO
6415          112#:  ERRDF  ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C#ERDF
                                .WORD     609
                                .WORD     T173SSR
                                .WORD     PKTSSR
6416 040746 005237 002222      120#:  INC     FATFLG    ;SET FATAL ERROR FLAG
6417 040752          CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
6418 040752 104406
6419          ;      Set WORDS 0-7 of expd message buffer = to recv since not testing
6420 040754 004737 047516      JSR     PC,T17SETEXP ;SET WORDS 0-7 EXPD=RCV
6421 040760 012701 045752      MOV    #T17EXSTA,R1 ;GET EXPECTED READ STATUS
6422 040764 012702 047612      MOV    #T17BFSTA,R2 ;GET RCV READ STATUS
6423 040770 012221          MOV    (R2)+,(R1)+ ;SET EXPD WORD #8 = RCV TEMP
6424 040772 011211          MOV    (R2),(R1) ;SET EXPD WORD #9 = RCV TEMP
6425 040774 052711 000020      BIS     #S2.INRDY,(R1) ;SET EXP INPUT READY= 1
6426 041000 052711 000040      BIS     #S2.OUTRDY,(R1) ;SET EXP OUTPUT READY= 1
6427 041004 042711 000200      BIC     #S2.DIM,(R1) ;SET EXP DATA IN MISS = 0
6428          ;      If Input Ready NOT=1 then Print Error
6429          ;      If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
6430 041010 005000          CLR     R0 ;HIGH RCV ADDRESS FOR CKMSG2
6431 041012 012701 047572      MOV    #T17BFR,R1 ;LOW RCV ADDRESS FOR CKMSG2
6432 041016 012702 045732      MOV    #T17EXP,R2 ;EXPD ADDRESS
6433 041022 012703 000024      MOV    #20,,R3 ;NUMBER OF BYTES TO COMPARE
6434 041026 004737 011500      JSR     PC,CKMSG2 ;EXPD EQUAL RCV?
6435 041032          FORCERROR 132# ,NOTSSR ;B0D
6436 041042 103404          BCS     140#    ;BR IF YES
6437 041044          NEXT.ERRNO
                                132#:  ERRHRD  ERRNO,T173CMP,MSGSTAT ;REPORT ERROR
                                TRAP      C#ERHRD
                                .WORD     610
                                .WORD     T173CMP
                                .WORD     MSGSTAT
6438 041044 104456
6439 041046 001142
6440 041050 046673
6441 041052 012350

```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```

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

```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```

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

```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```

6529      and byte counts from 2 to 64.
6530      1. Initial FIFO status is checked
6531      2. The Write FIFO function.
6532      3. Read Status is executed and FIFO status is checked.
6533      4. Read FIFO is executed and the data and final status
6534      is checked.
6535
6536      TEST STEPS:
6537
6538      BEGIN
6539      Write to TSSR to soft initialize
6540      Do a WRITE CHARACTERISTICS to setup a message buffer
6541      Do a Write Subsystem READ STATUS
6542      If Input Ready NOT=1 Then Print Error
6543      If Output Ready NOT=0 Then Print Error
6544      If Data In Miss NOT=0 Then Print Error
6545      If Last Word NOT=0 Then Print Error
6546      REPEAT FOR DATA 0 TO 377, 377 TO 0, FLOATING 1'S,0'S AND ALL 1'S/0'S
6547      REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
6548      BEGIN
6549      Do a Write Subsystem WRITE NPR to set tape direction out
6550      Do a Write Subsystem WRITE FIFO
6551      Do a Write Subsystem READ STATUS
6552      If Input Ready NOT=1 Then Print Error
6553      If Output Ready NOT=1 Then Print Error
6554      If Data In Miss NOT=0 Then Print Error
6555      If Last Word NOT=0 Then Print Error
6556      Do Write Subsystem READ FIFO
6557      If Data read from FIFO NOT= to Data sent Then Print Error
6558      Do a Write Subsystem READ STATUS
6559      If Input Ready NOT=1 Then Print Error
6560      If Output Ready NOT=0 Then Print Error
6561      If Data In Miss NOT=0 Then Print Error
6562      If Last Word NOT=0 Then Print Error
6563      END
6564      END
6565
6566      BGNSUB      ;//////////////// BEGIN SUBTEST //////////////////
6567      041454      T6.3:      TRAP      C#BSUB
6568      041454      104402
6569
6570      Write to TSSR register to soft initialize the controller
6571      54:
6572      JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
6573      BCS      104      ;BR IF SOFT INIT OKAY
6574      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
6575      ERDIF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
6576      041466      104455      TRAP      C#ERDF
6577      041470      001146      .WORD      614
6578      041472      003652      .WORD      SFIERR
6579      041474      012034      .WORD      SFIMSG
6580
6581      Do a WRITE CHARACTERISTICS to setup a message buffer
6582      104:
6583      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
6584      MOV      @T17PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
6585      JSR      PC,WRCHRR      ;DO WRITE CHARACTERISTICS COMMAND
6586      FORCERROR      424      ;GOODFORCE ERROR IF FORCER=1
6587      BCS      504      ;BR IF CARRY SET (GOOD RETURN)

```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```

6580 041530 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
6581 041532      NEXT.ERRNO
6582 041532      ERRDF  ERRNO,T17SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C1ERDF
                                .WORD    615
                                .WORD    T17SSR
                                .WORD    PKTSSR
                                421:
                                INC      FATFLG      ;SET FATAL ERROR FLAG
                                CKLOOP    ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C1CLP1
; Do a Write Subsystem READ STATUS
6586 041550 004737 047334 JSR      PC,T17SRD      ;SETUP PACKET FOR READ STATUS
6587 041554 012704 047720 MOV      @T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
6588 041560 010465 000000 MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
6589 041564 004737 016336 JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
6590 041570      FORCERROR 621      ;BDDFORCE ERROR IF FORCER=1
6591 041604 103407      BCS      701      ;BR IF CARRY SET (GOOD RETURN)
6592 041606 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
6593 041610      NEXT.ERRNO
6594 041610      ERRDF  ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C1ERDF
                                .WORD    616
                                .WORD    T173SSR
                                .WORD    PKTSSR
                                621:
                                INC      FATFLG      ;SET FATAL ERROR FLAG
                                CKLOOP    ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C1CLP1
; Set WORDS 0-7 of expd message buffer = to recv since not testing
6598 041626 004737 047516 JSR      PC,T17SETEXP    ;SET WORDS 0-7 EXPD=RCV
6599 041632 012701 045752 MOV      @T17EXSTA,R1    ;GET EXPECTED READ STATUS
6600 041636 012702 047612 MOV      @T17BFSTA,R2    ;GET RCV READ STATUS
6601 041642 012221      MOV      (R2), (R1)      ;SET EXPD WORD 08 = RCV TEMP
6602 041644 011211      MOV      (R2), (R1)      ;SET EXPD WORD 09 = RCV TEMP
6603 041646 052711 000020 BIS      @S2.INRDY,(R1)    ;SFT EXP INPUT READY= 1
6604 041652 042711 000040 BIC      @S2.OTRDY,(R1)    ;SET EXP OUTPUT READY= 0
6605 041656 042711 000200 BIC      @S2.DIM,(R1)      ;SET EXP DATA IN MISS = 0
6606 041662 042711 000100 BIC      @S2.ILW,(R1)      ;SET EXP LAST WORD (ILW)=0
6607      ; If Input Ready NOT=1 then Print Error
6608      ; If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
6609      ; If Last Word NOT=0 Then Print Error
6610 041666 005000      CLR      R0      ;HIGH RCV ADDRESS FOR CKMSG2
6611 041670 012701 047572 MOV      @T17BFR,R1      ;LOW RCV ADDRESS FOR CKMSG2
6612 041674 012702 045732 MOV      @T17EXP,R2      ;EXPD ADDRESS
6613 041700 012703 000024 MOV      @20.,R3      ;NUMBER OF BYTES TO COMPARE
6614 041704 004737 011500 JSR      PC,CKMSG2      ;EXPD EQUAL RCV?
6615 041710      FORCERROR 821,NOTSSR ;BDD
6616 041720 103404      BCS      901      ;BR IF YES
6617 041722      NEXT.ERRNO
6618 041722      ERRMRD  ERRNO,T171CMP,MSGSTAT ;REPORT ERROR
                                TRAP      C1ERRRD
                                .WORD    617
                                .WORD    T171CMP
                                .WORD    MSGSTAT
                                821:
                                CKLOOP    ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C1CLP1
6619 041732 104406      901:      CKLOOP
6620
6621

```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```

6622
6623      ; REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
6624      ; TSTFLAG =1 FOR INCREMENT TEST PATTERN
6625      ;           =2 FOR DECREMENT TEST PATTERN
6626      ;           =3 FOR TSTBLK TABLE PATTERN
6627 041734 012737 000001 002314      95#: MOV     #1,TSTFLAG      ;TEST PATTERN FLAG
6628 041742
6629 041742 012737 000002 002310      MOV     #2,COUNT      ;GET FIRST BYTE COUNT
6630 041750      100#:
6631      ; Do a Write Subsystem WRITE NPR to set tape direction out
6632 041750 012700 000100      MOV     #NP.OUT,R0      ;SET TAPE DIRECTION OUT
6633 041754 004737 047376      JSR     PC,T17SNPR      ;SETUP T17PK2 FOR WRITE NPR
6634 041760 012704 047720      MOV     #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
6635 041764 010465 000000      MOV     R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
6636 041770 004737 016336      JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
6637 041774      FORCERROR 102#      ;BR IF CARRY SET (GOOD RETURN)
6638 042010 103407      BCS     105#      ;SAVE CONTENTS OF TSSR
6639 042012 010001      MOV     R0,R1
6640 042014      NEXT.ERRNO
6641 042014      102#: ERRDF  ERRNO,T174SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
6642      042014 104455      TRAP     C:ERDF
6643      042016 001152      .WORD    618
6644      042020 046343      .WORD    T174SSR
6645      042022 012046      .WORD    PKTSSR
6646 042024 005237 002222      105#: INC     FATFLG      ;SET FATAL ERROR FLAG
6647 042030      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
6648 042030 104406      TRAP     C:CLP1
6649      ; Do a Write Subsystem WRITE FIFO
6650 042032 004737 047476      JSR     PC,T17CLEXP      ;CLEAR EXPD BUFFER
6651 042036 012701 046054      MOV     #T17WFDATA,R1      ;EXPD WRITE FIFO DATA BUFFER
6652 042042 013702 002310      MOV     COUNT,R2      ;TEST PATTERN SIZE
6653 042046 022737 000001 002314      CMP     #1,TSTFLAG      ;INCREMENT PATTERN THIS TIME THRU?
6654 042054 001005      BNE     115#      ;BR IF NO
6655 042056 005000      CLR     R0      ;INCREMENT TEST PATTERN
6656 042060 110021      110#: MOVB    R0,(R1)+      ;STORE INCREMENT TEST BYTE
6657 042062 005200      INC     R0      ;SET NEXT PATTERN
6658 042064 005302      DEC     R2      ;DONE?
6659 042066 003374      BGT     110#      ;BR IF NO
6660 042070 022737 000002 002314      115#: CMP     #2,TSTFLAG      ;DECREMENT PATTERN THIS TIME THRU?
6661 042076 001006      BNE     125#      ;BR IF NO
6662 042100 012700 000377      MOV     #377,R0      ;DECREMENT TEST PATTERN
6663 042104 110021      120#: MOVB    R0,(R1)+      ;STORE DECREMENT TEST BYTE
6664 042106 005300      DEC     R0      ;SET NEXT PATTERN
6665 042110 005302      DEC     R2      ;DONE?
6666 042112 003374      BGT     120#      ;BR IF NO
6667 042114 022737 000003 002314      125#: CMP     #3,TSTFLAG      ;TSTBLK PATTERNS THIS TIME THRU?
6668 042122 001005      BNE     135#      ;BR IF NO
6669 042124 012700 002752      MOV     #TSTBLK,R0      ;FLOAT 1'S/0'S ETC. TEST TABLE
6670 042130 112021      130#: MOVB    (R0)+,(R1)+      ;STORE A TSTBLK BYTE
6671 042132 005302      DEC     R2      ;DONE?
6672 042134 003375      BGT     130#      ;BR IF NO
6673 042136      135#: MOV     COUNT,R0      ;FIFO BYTE COUNT
6674 042142 012701 046054      MOV     #T17WFDATA,R1      ;FIFO WRITE DATA ADDRESS
6675 042146 004737 047422      JSR     PC,T17WIF      ;SETUP T17PK2 FOR WRITE FIFO
6676 042152 012704 047720      MOV     #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
6677 042156 010465 000000      MOV     R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE

```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```

6674 042162 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
6675 042166                     FORCERROR      142$      ;BDFORCE ERROR IF FORCER=1
6676 042202 103407             BCS      150$      ;BR IF CARRY SET (GOOD RETURN)
6677 042204 010001             MOV      R0,R1      ;SAVE CONTENTS OF TSSR
6678 042206                     NEXT.ERRNO
6679 042206 142$:             ERRDF      ERRNO,T175SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     619
                                .WORD     T175SSR
                                .WORD     PKTSSR
                                042206 104455
                                042210 001153
                                042212 046406
                                042214 012046
6680 042216 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
6681 042222 150$:             CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                042222 104406
6682
6683 ; Do a Write Subsystem READ STATUS
6684 042224 004737 047334      JSR      PC,T17SRD      ;SETUP PACKET FOR READ STATUS
6685 042230 012704 047720      MOV      @T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
6686 042234 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
6687 042240 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
6688 042244                     FORCERROR      157$      ;BDFORCE ERROR IF FORCER=1
6689 042260 103407             BCS      160$      ;BR IF CARRY SET (GOOD RETURN)
6690 042262 010001             MOV      R0,R1      ;SAVE CONTENTS OF TSSR
6691 042264                     NEXT.ERRNO
6692 042264 157$:             ERRDF      ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     620
                                .WORD     T173SSR
                                .WORD     PKTSSR
                                042264 104455
                                042266 001154
                                042270 046276
                                042272 012046
6693 042274 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
6694 042300 160$:             CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                042300 104406
6695
6696 ; Set WORDS 0-7 of expd message buffer = to recv since not testing
6697 042302 004737 047516      JSR      PC,T17SETEXP      ;SET WORDS 0-7 EXPD=RCV
6698 042306 012701 045752      MOV      @T17EXSTA,R1      ;GET EXPECTED READ STATUS
6699 042312 012702 047612      MOV      @T17BFSTA,R2      ;GET RCV READ STATUS
6700 042316 012221             MOV      (R2),R1      ;SET EXPD WORD #8 = RCV TEMP
6701 042320 011211             MOV      (R2),R1      ;SET EXPD WORD #9 = RCV TEMP
6702 042322 052711 000020      BIS      @S2.INPDY,(R1)      ;SET EXP INPUT READY= 1
6703 042326 052711 000040      BIS      @S2.OTRDY,(R1)      ;SET EXP OUTPUT READY= 1
6704 042332 042711 000200      BIC      @S2.DIM,(R1)      ;SET EXP DATA IN MISS = 0
6705 042336 042711 000100      BIC      @S2.ILW,(R1)      ;SET EXP LAST WORD (ILW)=0
6706
6707 ; If Input Ready NOT=1 then Print Error
6708 042342 005000             CLR      R0
6709 042344 012701 047572      MOV      @T17BFR,R1      ;HIGH RCV ADDRESS FOR CKMSG2
6710 042350 012702 045732      MOV      @T17EXP,R2      ;LOW RCV ADDRESS FOR CKMSG2
6711 042354 012703 000024      MOV      @20,R3      ;EXPD ADDRESS
6712 042360 004737 011500      JSR      PC,CKMSG2      ;NUMBER OF BYTES TO COMPARE
6713 042364                     FORCERROR      162$,NOTSSR ;EXPD EQUAL RCV?
6714 042374 103404             BCS      170$      ;BDF
6715 042376                     NEXT.ERRNO
6716 042376 162$:             ERRMRD      ERRNO,T173CMP,MSGSTAT ;REPORT ERROR
                                TRAP      C$ERRMRD
                                .WORD     621
                                .WORD     T173CMP
                                .WORD     MSGSTAT
                                042376 104456
                                042400 001155
                                042402 046673
                                042404 012350

```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```

6717 042406      104406      170$: CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
6718 042406
6719
6720 042410 013700 002310      ; Do Write Subsystem READ FIFO
                                MOV      COUNT,R0                      ;SET READ BYTE COUNT
6721 042414 004737 047456      JSR      PC,T17RFIF                    ;SETUP T17PK2 FOR READ FIFO
6722 042420 012704 047720      MOV      @T17PK2,R4                    ;GET WRITE SUBSYSTEM COMMAND PACKET
6723 042424 010465 000000      MOV      R4,TSD8(R5)                  ;SET THE PACKET ADDRESS TO EXECUTE
6724 042430 004737 016336      JSR      PC,CHKTSSR                    ;WAIT FOR SSR TO SET
6725 042434                                FORCERROR 172$              ;BDDFORCE ERROR IF FORCER=1
6726 042450 103407      BCS      180$                                ;BR IF CARRY SET (GOOD RETURN)
6727 042452 010001      MOV      R0,R1                                ;SAVE CONTENTS OF TSSR
6728 042454                                NEXT.ERRNO
6729 042454      172$: ERRDF  ERRNO,T176SSR,PKTSSR                ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD    622
                                .WORD    T176SSR
                                .WORD    PKTSSR
6730 042464 005237 002222      INC      FATFLG                      ;SET FATAL ERROR FLAG
6731 042470      180$: CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
6732
6733      ; If Data read from FIFO NOT= to Data sent Then Print Error
6734 042472 005000      CLR      R0                                ;HIGH RECV ADDRESS FOR CKMSG2
6735 042474 012702 046054      MOV      @T17WFDATA,R2                ;GET EXPECTED ADDRESS FOR CKMSG2
6736 042500 012701 047612      MOV      @T17BFSTA,R1                ;GET RECEIVED ADDRESS FOR CKMSG2
6737 042504 013703 002310      MOV      COUNT,R3                    ;NUMBER OF BYTES TO COMPARE
6738 042510 004737 011500      JSR      PC,CKMSG2                    ;EXPD EQUAL RECV?
6739 042514                                FORCERROR 192$,NOTSSR      ;BDD
6740 042524 103406      BCS      200$                                ;BR IF YES
6741 042526                                NEXT.ERRNO
6742 042526 013701 002310      192$: MOV      COUNT,R1                ;GET BYTE COUNT
6743 042532      ERRMRD  ERRNO,T175CMP,FIFEXP                    ;REPORT ERROR
                                TRAP      C$ERMRD
                                .WORD    623
                                .WORD    T175CMP
                                .WORD    FIFEXP
6744 042542      200$: CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
6745
6746      ; Do a Write Subsystem READ STATUS
6747 042544 004737 047334      JSR      PC,T17SRD                    ;SETUP PACKET FOR READ STATUS
6748 042550 012704 047720      MOV      @T17PK2,R4                    ;GET WRITE SUBSYSTEM COMMAND PACKET
6749 042554 010465 000000      MOV      R4,TSD8(R5)                  ;SET THE PACKET ADDRESS TO EXECUTE
6750 042560 004737 016336      JSR      PC,CHKTSSR                    ;WAIT FOR SSR TO SET
6751 042564                                FORCERROR 212$              ;BDDFORCE ERROR IF FORCER=1
6752 042600 103407      BCS      220$                                ;BR IF CARRY SET (GOOD RETURN)
6753 042602 010001      MOV      R0,R1                                ;SAVE CONTENTS OF TSSR
6754 042604                                NEXT.ERRNO
6755 042604      212$: ERRDF  ERRNO,T173SSR,PKTSSR                ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD    624
                                .WORD    T173SSR
                                .WORD    PKTSSR
6756 042614 005237 002222      INC      FATFLG                      ;SET FATAL ERROR FLAG
6757 042620      220$: CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1

```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```

6758      ; Set WORDS 0-7 of expd message buffer = to recv since not testing
6759 042622 004737 047516      JSR PC,T17SETEXP      ;SET WORDS 0-7 EXPD=RCV
6760 042626 012701 045752      MOV #T17EXSTA,R1      ;GET EXPECTED READ STATUS
6761 042632 012702 047612      MOV #T17BFSTA,R2      ;GET RCV READ STATUS
6762 042636 012221              MOV (R2),.(R1)+      ;SET EXPD WORD #8 = RCV TEMP
6763 042640 011211              MOV (R2),.(R1)      ;SET EXPD WORD #9 = RCV TEMP
6764 042642 052711 000020      BIS #S2.INRDY,(R1)      ;SET EXP INPUT READY= 1
6765 042646 042711 000040      BIC #S2.OTRDY,(R1)      ;SET EXP OUTPUT READY= 0
6766 042652 042711 000200      BIC #S2.DIM,(R1)      ;SET EXP DATA IN MISS = 0
6767 042656 042711 000100      BIC #S2.ILW,(R1)      ;SET EXP LAST WORD (ILW)=0
6768      ; If Input Ready NOT=1 then Print Error
6769      ; If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
6770 042662 005000              CLR R0              ;HIGH RCV ADDRESS FOR CKMSG2
6771 042664 012701 047572      MOV #T17BFR,R1      ;LOW RCV ADDRESS FOR CKMSG2
6772 042670 012702 045732      MOV #T17EXP,R2      ;EXPD ADDRESS
6773 042674 012703 000024      MOV #20.,R3      ;NUMBER OF BYTES TO COMPARE
6774 042700 004737 011500      JSR PC,CKMSG2      ;EXPD EQUAL RCV?
6775 042704              FORCERROR 232$,NOTSSR      ;@80
6776 042714 103404              BCS 240$          ;BR IF YES
6777 042716              NEXT.ERRNO
6778 042716 232$: ERRHRD ERRNO,T174CMP,MSGSTAT      ;REPORT ERROR
6779 042716 104456              TRAP C$ERHRD
6780 042720 001161              .WORD 625
6781 042722 046757              .WORD T174CMP
6782 042724 012350              .WORD MSGSTAT
6783 042726 104406              240$: CKLOOP          ;LOOP ON ERROR. IF FLAG SET
6784 042730              FORCEEXIT 250$              ;@80
6785 042740 005237 002310      INC COUNT              ;GET NEXT BYTE COUNT
6786 042744 023727 002310 000077      CMP COUNT,#77      ;DONE 0 TO 77
6787 042752 101002              BHI 250$            ;BR IF YES
6788 042754 000137 041750      JMP 100$            ;DO ANOTHER BYTE COUNT
6789 042760 005237 002314      250$: INC TSTFLAG      ;GET NEXT TEST PATTERN CODE
6790 042764 023727 002314 000003      CMP TSTFLAG,#3      ;DONE INC,DEC,TSTBLK PATTERNS?
6791 042772 101002              BHI 255$            ;BR IF YES
6792 042774 000137 041742      JMP 95$             ;DO ANOTHER TEST PATTERN
6793 043000 255$: ENDSUB          ;//////////////// END SUBTEST //////////////////
6794 043000              L10061:
6795 043000 104403              TRAP C$ESUB
6796 043002 005737 002222      TST FATFLG          ;ANY FATAL ERRORS ?
6797 043006 001402              BEQ 260$            ;BRANCH IF NOT
6798 043010 004737 017202      JSR PC,CKDROP        ;TRY TO DROP THE UNIT
6799 043014 260$:
6800      .SBTTL TEST 6: SUBTEST 4: FIFO Verify ILW Status
6801      ;**
6802      ; TEST 6: SUBTEST 4:
6803      ; SUBTEST DESCRIPTION:
6804      ;
6805      ; This subtest verifies that reading the FIFO when it is
6806      ; empty causes the Last Word (ILW) status to assert.
6807

```

TEST 6: SUBTEST 4: FIFO Verify ILW Status

```

6808      ;
6809      ; TEST STEPS:
6810      ;
6811      ; BEGIN
6812      ;       Write to TSSR to soft initialize
6813      ;       Do Write Subsystem READ FIFO with byte count equal to 1
6814      ;       Do a Write Subsystem READ STATUS
6815      ;       If Input Ready NOT=1 Then Print Error
6816      ;       If Output Ready NOT=0 Then Print Error
6817      ;       If Data In Miss NOT=0 Then Print Error
6818      ;       If Last Word (ILW) NOT=1 Then Print Error
6819      ;
6820      ;       END
6821      ;--
6821      043014      BGNSUB                      ;////////// BEGIN SUBTEST //////////
6821      043014      T6.4:                      TRAP      C#BSUB
6821      043014      104402
6822      ;
6823      ;       Write to TSSR register to soft initialize the controller
6824      043016      5$:
6825      043016      JSR      PC,SOFINIT          ;WRITE TO TSSR TO SOFT INITIALIZE
6826      043022      BCS      10$                ;BR IF SOFT INIT OKAY
6827      043024      MOV      R0,R1              ;SAVE CONTENTS OF TSSR
6828      043026      ERDF     ERRNO,SFIERR,SFMSG  ;DEVICE FATAL DURING INIT
6828      043026      104455                      TRAP      C#ERDF
6828      043030      001161                      .WORD     625
6828      043032      003652                      .WORD     SFIERR
6828      043034      012034                      .WORD     SFMSG
6829      ;
6830      043036      10$:      Do a WRITE CHARACTERISTICS to setup a message buffer
6831      043042      012704      CLR      FATFLG          ;CLEAR FATAL ERROR FLAG
6832      043046      004737      MOV      @T17PACKET,R4    ;GET THE ADDRESS OF COMMAND PACKET
6833      043052      103407      JSR      PC,WRTPCHR        ;DO WRITE CHARACTERISTICS COMMAND
6834      043066      010001      FORCERROR 42$            ;GOODFORCE ERROR IF FORCER=1
6835      043070      010001      BCS      50$                ;BR IF CARRY SET (GOOD RETURN)
6836      043072      010001      MOV      R0,R1              ;SAVE CONTENTS OF TSSR
6837      043072      42$:      ERDF     ERRNO,T17SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
6837      043072      104455                      TRAP      C#ERDF
6837      043074      001162                      .WORD     626
6837      043076      046175                      .WORD     T17SSR
6837      043100      012046                      .WORD     PKTSSR
6838      043102      005237      002222      INC      FATFLG          ;SET FATAL ERROR FLAG
6839      043106      043106      50$:      CKLOOP          ;LOOP ON ERROR, IF FLAG SET
6839      043106      104406                      TRAP      C#CLP1
6840      ;
6841      ;       Do Write Subsystem READ FIFO with byte count equal to 1
6842      043110      012700      MOV      @1,R0              ;SET READ BYTE COUNT
6843      043114      004737      JSR      PC,T17RFIF        ;SETUP T17PK2 FOR READ FIFO
6844      043120      012704      MOV      @T17PK2,R4        ;GET WRITE SUBSYSTEM COMMAND PACKET
6845      043124      010465      000000      MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
6846      043130      004737      016336      JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
6847      043134      103407      FORCERROR 142$            ;GOODFORCE ERROR IF FORCER=1
6848      043150      010001      BCS      150$                ;BR IF CARRY SET (GOOD RETURN)
6849      043152      010001      MOV      R0,R1              ;SAVE CONTENTS OF TSSR
6850      043154      142$:      ERDF     ERRNO,T176SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
6851      043154      104455                      TRAP      C#ERDF
6851      043156      001163                      .WORD     627

```

TEST 6: SUBTEST 4: FIFO Verify ILW Status

```

043160 046452
043162 012046
6852 043164 005237 002222
6853 043170 104406
6854
6855
6856 043172 004737 047334
6857 043176 012704 047720
6858 043202 010465 000000
6859 043206 004737 016336
6860 043212
6861 043226 103407
6862 043230 010001
6863 043232
6864 043232
043232 104455
043234 001164
043236 046276
043240 012046
6865 043242 005237 002222
6866 043246 104406
6867
6868 043250 004737 047516
6869 043254 012701 045752
6870 043260 012702 047612
6871 043264 012221
6872 043266 011211
6873 043270 052711 000020
6874 043274 042711 000040
6875 043300 042711 000200
6876 043304 052711 000100
6877
6878
6879
6880 043310 005000
6881 043312 012701 047572
6882 043316 012702 045732
6883 043322 012703 000024
6884 043326 004737 011500
6885 043332
6886 043342 103404
6887 043344
6888 043344
043344 104456
043346 001165
043350 047116
043352 012350
6889 043354 104406
6890
6891 043356
043356
043356 104403
6892
6893 043360 005737 002222

```

150\$: INC FATFLG ;SET FATAL ERROR FLAG
CKLOOP ;LOOP ON ERROR, IF FLAG SET TRAP C#CLP1

Do a Write Subsystem READ STATUS
JSR PC,T17SRD ;SETUP PACKET FOR READ STATUS
MOV #T17PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
MOV R4,TSDR(R5) ;SET THE PACKET ADDRESS TO EXECUTE
JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
FORCERROR 162\$;B0DFORCE ERROR IF FORCER=1
BCS 170\$;BR IF CARRY SET (GOOD RETURN)
MOV R0,R1 ;SAVE CONTENTS OF TSSR
NEXT.ERRNO
162\$: ERRDF ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
TRAP C#ERDF
WORD 628
WORD T173SSR
WORD PKTSSR

170\$: INC FATFLG ;SET FATAL ERROR FLAG
CKLOOP ;LOOP ON ERROR, IF FLAG SET TRAP C#CLP1

Set WORDS 0-7 of expd message buffer = to recv since not testing
JSR PC,T17SETEXP ;SET WORDS 0-7 EXPD=RCV
MOV #T17EXSTA,R1 ;GET EXPECTED READ STATUS
MOV #T17BFSTA,R2 ;GET RCV READ STATUS
MOV (R2)+,(R1)+ ;SET EXPD WORD #8 = RCV TEMP
MOV (R2),(R1) ;SET EXPD WORD #9 = RCV TEMP
BIS #S2.INRDY,(R1) ;SET EXP INPUT READY= 1
BIC #S2.OTRDY,(R1) ;SET EXP OUTPUT READY= 0
BIC #S2.DIM,(R1) ;SET EXP DATA IN MISS = 0
BIS #S2.ILW,(R1) ;SET EXP LAST WORD (ILW)=1
; If Input Ready NOT=1 then Print Error
; If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
; If Last Word (ILW) NOT=1 Then Print Error
CLR R0 ;HIGH RCV ADDRESS FOR CKMSG2
MOV #T17BFR,R1 ;LOW RCV ADDRESS FOR CKMSG2
MOV #T17EXP,R2 ;EXPD ADDRESS
MOV #20,R3 ;NUMBER OF BYTES TO COMPARE
JSR PC,CKMSG2 ;EXPD EQUAL RCV?
FORCERROR 172\$,NOTSSR ;B0D
BCS 180\$;BR IF YES
NEXT.ERRNO
172\$: ERRHRD ERRNO,T176CMP,MSGSTAT ;REPORT ERROR
TRAP C#ERHRD
WORD 629
WORD T176CMP
WORD MSGSTAT

180\$: CKLOOP ;LOOP ON ERROR, IF FLAG SET TRAP C#CLP1

ENDSUB ;////////// END SUBTEST ///////////
L10062: TRAP C#ESUB

TST FATFLG ;ANY FATAL ERRORS ?

TEST 6: SUBTEST 4: FIFO Verify ILW Status

```

6894 043364 001402          BEQ      260$          ;BRANCH IF NOT
6895 043366 004737 017202    JSR      PC,CKDROP    ;TRY TO DROP THE UNIT
6896 043372          260$:
6897
6898
6899          .SBTTL TEST 6: SUBTEST 5: FIFO Verify Input Ready
6900
6901      ;**
6902      ; TEST 6: SUBTEST 5:
6903      ;
6904      ; SUBTEST DESCRIPTION:
6905      ;
6906      ;     This subtest verifies that writing 64. bytes into the FIFO
6907      ;     without reading any out causes the Input Ready status to
6908      ;     negate. The Subtest then verifies that writing a 65th byte
6909      ;     into the FIFO causes the Data In Miss status to assert.
6910      ;     Next it is verified that the original 64 bytes can be read
6911      ;     out correctly and that the data has not been corrupted.
6912      ;
6913      ; TEST STEPS:
6914      ;
6915      ; BEGIN
6916      ;     Write to TSSR to soft initialize
6917      ;     Do a WRITE CHARACTERISTICS to setup a message buffer
6918      ;     Do a Write Subsystem WRITE NPR to set tape direction out
6919      ;     Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
6920      ;     Do a Write Subsystem READ STATUS
6921      ;     If Input Ready NOT=0 Then Print Error
6922      ;     If Output Ready NOT=1 Then Print Error
6923      ;     If Data In Miss NOT=0 Then Print Error
6924      ;     Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
6925      ;     Do a Write Subsystem READ STATUS
6926      ;     If Input Ready NOT=0 Then Print Error
6927      ;     If Output Ready NOT=1 Then Print Error
6928      ;     If Data In Miss NOT=1 Then Print Error
6929      ;     Do Write Subsystem READ FIFO
6930      ;     If Data read from FIFO NOT= to Data sent Then Print Error
6931      ;     Do a Write Subsystem READ STATUS
6932      ;     If Input Ready NOT=1 Then Print Error
6933      ;     If Output Ready NOT=0 Then Print Error
6934      ;     If Data In Miss NOT=1 Then Print Error
6935      ; END
6936      ; --
6937 043372          BGNSUB          ;////////// BEGIN SUBTEST ///////////
6938 043372          T6.5:          TRAP      C$BSUB
6939 043372 104402
6940
6941      ;
6942      ; 5$: Write to TSSR register to soft initialize the controller
6943      ;
6944      ; JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
6945      ; BCS      10$            ;BR IF SOFT INIT OKAY
6946      ; MOV      R0,R1          ;SAVE CONTENTS OF TSSR
6947      ; ERDF     ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
6948      ;
6949      ; TRAP     C$ERDF
6950      ; .WORD    629
6951      ; .WORD    SFIERR
6952      ; .WORD    SFIMSG

```

TEST 6: SUBTEST 5: FIFO Verify Input Ready

```

6945      ; Do a WRITE CHARACTERISTICS to setup a message buffer
6946 043414 005037 002222      10$: CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
6947 043420 012704 047550      MOV      #T17PK2,R4      ;GET THE ADDRESS OF COMMAND PACKET
6948 043424 004737 010662      JSR      PC,WRTCHR      ;DO WRITE CHARACTERISTICS COMMAND
6949 043430      FORCERROR      42$      ;DO FORCE ERROR IF FORCER=1
6950 043444 103407      BCS      50$      ;BR IF CARRY SET (GOOD RETURN)
6951 043446 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
6952 043450      NEXT.ERRNO
6953 043450      42$: ERRDF      ERRNO,T17SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      630
                                .WORD      T17SSR
                                .WORD      PKTSSR
6954 043460 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
6955 043464      50$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
6956
6957      ; Do a Write Subsystem WRITE NPR to set tape direction out
6958 043466 C12700 000100      100$: MOV      #NP.OUT,R0      ;SET TAPE DIRECTION OUT
6959 043472 004737 047376      JSR      PC,T17SNPR      ;SETUP T17PK2 FOR WRITE NPR
6960 043476 012704 047720      MOV      #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
6961 043502 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
6962 043506 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
6963 043512      FORCERROR      102$      ;DO FORCE ERROR IF FORCER=1
6964 043526 103407      BCS      105$      ;BR IF CARRY SET (GOOD RETURN)
6965 043530 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
6966 043532      NEXT.ERRNO
6967 043532      102$: ERRDF      ERRNO,T174SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      631
                                .WORD      T174SSR
                                .WORD      PKTSSR
6968 043542 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
6969 043546      105$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
6970
6971      ; Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
6972 043550 012737 000100 002310      MOV      #64.,COUNT      ;WRITE 64 BYTES
6973 043556 012701 046054      MOV      #T17WFDATA,R1      ;EXPD WRITE FIFO DATA BUFFER
6974 043562 012702 000100      MOV      #64.,R2      ;TEST PATTERN SIZE
6975 043566 005000      CLR      R0      ;INCREMENT TEST PATTERN
6976 043570 110021      110$: MOV      R0,(R1)+      ;STORE INCREMENT TEST BYTE
6977 043572 005200      INC      R0      ;SET NEXT PATTERN
6978 043574 005302      DEC      R2      ;DONE?
6979 043576 003374      BGT      110$      ;BR IF NO
6980 043600 013700 002310      MOV      COUNT,R0      ;FIFO BYTE COUNT
6981 043604 012701 046054      MOV      #T17WFDATA,R1      ;FIFO WRITE DATA ADDRESS
6982 043610 004737 047422      JSR      PC,T17WFI      ;SETUP T17PK2 FOR WRITE FIFO
6983 043614 012704 047720      MOV      #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
6984 043620 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
6985 043624 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
6986 043630      FORCERROR      142$      ;DO FORCE ERROR IF FORCER=1
6987 043644 103407      BCS      150$      ;BR IF CARRY SET (GOOD RETURN)
6988 043646 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
6989 043650      NEXT.ERRNO
6990 043650      142$: ERRDF      ERRNO,T175SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      630
                                .WORD      T175SSR
                                .WORD      PKTSSR

```

TEST 6: SUBTEST 5: FIFO Verify Input Ready

043652	001170					.WORD	632
043654	046406					.WORD	T175SSR
043656	012046					.WORD	PKTSSR
6991	043660	005237	002222				
6992	043664			150#:	INC FATFLG		;SET FATAL ERROR FLAG
	043664	104406			CKLOOP		;LOOP ON ERROR, IF FLAG SET
						TRAP	C#CLP1
6993							
6994				:	Do a Write Subsystem READ STATUS		
6995				:	If Input Ready NOT=0 Then Print Error		
6996				:	If Output Ready NOT=1 Then Print Error		
6997				:	If Data In Miss NOT=0 Then Print Error		
6998	043666	004737	047334		JSR PC,T17SRD		;SETUP PACKET FOR READ STATUS
6999	043672	012704	047720		MOV #T17PK2,R4		;GET WRITE SUBSYSTEM COMMAND PACKET
7000	043676	010465	000000		MOV R4,TSD8(R5)		;SET THE PACKET ADDRESS TO EXECUTE
7001	043702	004737	016336		JSR PC,CHKTSSR		;WAIT FOR SSR TO SET
7002	043706				FORCERROR 157#		;BR IF CARRY SET (GOOD RETURN)
7003	043722	103407			BCS 160#		;SAVE CONTENTS OF TSSR
7004	043724	010001			MOV R0,R1		
7005	043726				NEXT.ERRNO		
7006	043726			157#:	ERRDF ERRNO,T173SSR,PKTSSR		;DEVICE FATAL SSR FAILED TO SET
	043726	104455				TRAP	C#ERDF
	043730	001171				.WORD	633
	043732	046276				.WORD	T173SSR
	043734	012046				.WORD	PKTSSR
7007	043736	005237	002222		INC FATFLG		;SET FATAL ERROR FLAG
7008	043742			160#:	CKLOOP		;LOOP ON ERROR, IF FLAG SET
	043742	104406				TRAP	C#CLP1
7009				:	Set WORDS 0-7 of expd message buffer = to recv since not testing		
7010	043744	004737	047516		JSR PC,T17SETEXP		;SET WORDS 0-7 EXPD=RCV
7011	043750	012701	045752		MOV #T17EXSTA,R1		;GET EXPECTED READ STATUS
7012	043754	012702	047612		MOV #T17BFSTA,R2		;GET RCV READ STATUS
7013	043760	012221			MOV (R2), (R1)		;SET EXPD WORD #8 = RCV TEMP
7014	043762	011211			MOV (R2), (R1)		;SET EXPD WORD #9 = RCV TEMP
7015	043764	042711	000020		BIC #S2.INRDY,(R1)		;SET EXP INPUT READY= 0
7016	043770	052711	000040		BIS #S2.OTRDY,(R1)		;SET EXP OUTPUT READY= 1
7017	043774	042711	000200		BIC #S2.DIM,(R1)		;SET EXP DATA IN MISS = 0
7018	044000	005000			CLR R0		;HIGH RCV ADDRESS FOR CKMSG2
7019	044002	012701	047572		MOV #T17BFR,R1		;LOW RCV ADDRESS FOR CKMSG2
7020	044006	012702	045732		MOV #T17EXP,R2		;EXPD ADDRESS
7021	044012	012703	000024		MOV #20,R3		;NUMBER OF BYTES TO COMPARE
7022	044016	004737	011500		JSR PC,CKMSG2		;EXPD EQUAL RCV?
7023	044022				FORCERROR 162#,NOTSSR		;BR
7024	044032	103404			BCS 170#		;BR IF YES
7025	044034				NEXT.ERRNO		
7026	044034			162#:	ERRHRD ERRNO,T173CMP,MSGSTAT		;REPORT ERROR
	044034	104456				TRAP	C#ERHRD
	044036	001172				.WORD	634
	044040	046673				.WORD	T173CMP
	044042	012350				.WORD	MSGSTAT
7027	044044			170#:	CKLOOP		;LOOP ON ERROR, IF FLAG SET
	044044	104406				TRAP	C#CLP1
7028							
7029							
7030				:	Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written		
7031	044046	012700	000001		MOV #1,R0		;FIFO BYTE COUNT
7032	044052	012701	046054		MOV #T17WFDATA,R1		;FIFO WRITE DATA ADDRESS
7033	044056	004737	047 22		JSR PC,T17WFIF		;SETUP T17PK2 FOR WRITE FIFO

TEST 6: SUBTEST 5: FIFO Verify Input Ready

```

7034 044062 012704 047720      MOV      #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
7035 044066 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
7036 044072 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
7037 044076      FORCERROR      172$      ;SSDFORCE ERROR IF FORCER=1
7038 044112 103407      BCS      180$      ;BR IF CARRY SET (GOOD RETURN)
7039 044114 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
7040 044116      NEXT.ERRNO
7041 044116      172$: ERRDF      ERRNO,T175SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      635
                                .WORD      T175SSR
                                .WORD      PKTSSR
                                7042 044126 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
                                7043 044132      180$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
7044      ;
7045      ; Do a Write Subsystem READ STATUS
7046      ; If Input Ready NOT=0 Then Print Error
7047      ; If Output Ready NOT=1 Then Print Error
7048      ; If Data In Miss NOT=1 Then Print Error
7049 044134 004737 047334      JSR      PC,T17SRD      ;SETUP PACKET FOR READ STATUS
7050 044140 012704 047720      MOV      #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
7051 044144 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
7052 044150 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
7053 044154      FORCERROR      187$      ;SSDFORCE ERROR IF FORCER=1
7054 044170 103407      BCS      190$      ;BR IF CARRY SET (GOOD RETURN)
7055 044172 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
7056 044174      NEXT.ERRNO
7057 044174      187$: ERRDF      ERRNO,T173SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      636
                                .WORD      T173SSR
                                .WORD      PKTSSR
                                7058 044204 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
                                7059 044210      190$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
7060      ; Set WORDS 0-7 of expd message buffer = to recv since not testing
7061 044212 004737 047516      JSR      PC,T17SETEXP      ;SET WORDS 0-7 EXPD=RCV
7062 044216 012701 045752      MOV      #T17EXSTA,R1      ;GET EXPECTED READ STATUS
7063 044222 012702 047612      MOV      #T17BFSTA,R2      ;GET RECV READ STATUS
7064 044226 012221      MOV      (R2)+,(R1)+      ;SET EXPD WORD #8 = RECV TEMP
7065 044230 011211      MOV      (R2),(R1)      ;SET EXPD WORD #9 = RECV TEMP
7066 044232 042711 000020      BIC      #S2.INRDY,(R1)      ;SET EXP INPUT READY= 0
7067 044236 052711 000040      BIS      #S2.OTRDY,(R1)      ;SET EXP OUTPUT READY= 1
7068 044242 052711 000200      BIS      #S2.DIM,(R1)      ;SET EXP DATA IN MISS = 1
7069 044246 005000      CLR      R0      ;HIGH RECV ADDRESS FOR CKMSG2
7070 044250 012701 047572      MOV      #T17BFR,R1      ;LOW RECV ADDRESS FOR CKMSG2
7071 044254 012702 045732      MOV      #T17EXP,R2      ;EXPD ADDRESS
7072 044260 012703 000021      MOV      #20,R3      ;NUMBER OF BYTES TO COMPARE
7073 044264 004737 011501      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
7074 044270      FORCERROR      192$,NOTSSR      ;SSD
7075 044300 103404      BCS      200$      ;BR IF YES
7076 044302      NEXT.ERRNO
7077 044302      192$: ERRMRD      ERRNO,T173CMP,MSGSTAT      ;REPORT ERROR
                                TRAP      C$ERMRD
                                .WORD      637
                                .WORD      T173CMP
                                044302 104456
                                044304 001175
                                044306 046673

```

TEST 6: SUBTEST 5: FIFO Verify Input Ready

7078	044310	012350	200#:	CKLOOP	;LOOP ON ERROR, IF FLAG	SET	MSGSTAT
	044312					TRAP	C#CLP1
7079	044312	104406	:	Do Write Subsystem READ FIFO			
7080	044314	013700		MOV COUNT,R0	;SET READ BYTE COUNT		
7081	044320	004737		JSR PC,T17RFIF	;SETUP T17PK2 FOR READ FIFO		
7082	044324	012704		MOV #T17PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET		
7083	044330	010465		MOV R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE		
7084	044334	004737		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET		
7085	044340			FORCERROR 212#	;BDFORCE ERROR IF FORCER=1		
7086	044354	103407		BCS 220#	;BR IF CARRY SET (GOOD RETURN)		
7087	044356	010001		MOV R0,R1	;SAVE CONTENTS OF TSSR		
7088	044360			NEXT.ERRNO			
7089	044360		212#:	ERRDF ERRNO,T176SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET		
	044360	104455				TRAP	C#ERDF
	044362	001176				.WORD	638
	044364	046452				.WORD	T176SSR
	044366	012046				.WORD	PKTSSR
7090	044370	C05237		INC FATFLG	;SET FATAL ERROR FLAG		
7091	044374		220#:	CKLOOP	;LOOP ON ERROR, IF FLAG	SET	
	044374	104406				TRAP	C#CLP1
7092			:	If Data read from FIFO NOT= to Data sent Then Print Error			
7093				CLR R0	;HIGH RECV ADDRESS FOR CKMSG2		
7094	044376	005000		MOV #T17WFDATA,R2	;GET EXPECTED ADDRESS FOR CKMSG2		
7095	044400	012702		MOV #T17BFSTA,R1	;GET RECEIVED ADDRESS FOR CKMSG2		
7096	044404	012701		MOV COUNT,R3	;NUMBER OF BYTES TO COMPARE		
7097	044410	013703		JSR PC,CKMSG2	;EXPD EQUAL RECV?		
7098	044414	004737		FORCERROR 232#,NOTSSR	;BDF		
7099	044420			BCS 240#	;BR IF YES		
7100	044430	103406		NEXT.ERRNO			
7101	044432			MOV COUNT,R1	;GET BYTE COUNT		
7102	044432	013701	232#:	ERRHRD ERRNO,T175CMP,FIFEXP	;REPORT ERROR		
7103	044436					TRAP	C#ERHRD
	044436	104456				.WORD	639
	044440	001177				.WORD	T175CMP
	044442	047042				.WORD	FIFEXP
	044444	012170					
7104	044446		240#:	CKLOOP	;LOOP ON ERROR, IF FLAG	SET	
	044446	104406				TRAP	C#CLP1
7105			:	Do a Write Subsystem READ STATUS			
7106			:	If Input Ready NOT=1 Then Print Error			
7107			:	If Output Ready NOT=0 Then Print Error			
7108			:	If Data In Miss NOT=1 Then Print Error			
7109			:				
7110	044450	004737		JSR PC,T17SRD	;SETUP PACKET FOR READ STATUS		
7111	044454	012704		MOV #T17PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET		
7112	044460	010465		MOV R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE		
7113	044464	004737		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET		
7114	044470			FORCERROR 252#	;BDFORCE ERROR IF FORCER=1		
7115	044504	103407		BCS 260#	;BR IF CARRY SET (GOOD RETURN)		
7116	044506	010001		MOV R0,R1	;SAVE CONTENTS OF TSSR		
7117	044510			NEXT.ERRNO			
7118	044510		252#:	ERRDF ERRNO,T173SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET		
	044510	104455				TRAP	C#ERDF
	044512	001200				.WORD	640
	044514	046276				.WORD	T173SSR
	044516	012046				.WORD	PKTSSR

TEST 6: SUBTEST 5: FIFO Verify Input Ready

```

7119 044520 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
7120 044524      2604:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
7121      ;      Set WORDS 0-7 of expd message buffer = to rcv since not testing
7122 044526 004737 047516      JSR      PC,T17SETEXP ;SET WORDS 0-7 EXPD=RCV
7123 044532 012701 045752      MOV      #T17EXSTA,R1 ;GET EXPECTED READ STATUS
7124 044536 012702 047612      MOV      #T17BFSTA,R2 ;GET RCV READ STATUS
7125 044542 012221      MOV      (R2), (R1) ;SET EXPD WORD #0 = RCV TEMP
7126 044544 011211      MOV      (R2), (R1) ;SET EXPD WORD #9 = RCV TEMP
7127 044546 052711 000020      BIS      #S2.INRDY,(R1) ;SET EXP INPUT READY= 1
7128 044552 042711 000040      BIC      #S2.OUTRDY,(R1) ;SET EXP OUTPUT READY= 0
7129 044556 052711 000200      BIS      #S2.DIM,(R1) ;SET EXP DATA IN MISS = 1
7130 044562 005000      CLR      R0 ;HIGH RCV ADDRESS FOR CKMSG2
7131 044564 012701 047572      MOV      #T17BFR,R1 ;LOW RCV ADDRESS FOR CKMSG2
7132 044570 012702 045732      MOV      #T17EXP,R2 ;EXPD ADDRESS
7133 044574 012703 000024      MOV      #20,,R3 ;NUMBER OF BYTES TO COMPARE
7134 044600 004737 011500      JSR      PC,CKMSG2 ;EXPD EQUAL RCV?
7135 044604      FORCERROR 2724,NOTSSR ;000
7136 044614 103404      BCS      2804 ;BR IF YES
7137 044616      NEXT.ERRNO
7138 044616      2724:      ERRHRD ERRNO,T174CMP,MSGSTAT ;REPORT ERROR
                                TRAP      C#ERHRD
                                .WORD      641
                                .WORD      T174CMP
                                .WORD      MSGSTAT
7139 044626      2804:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
7140 044626 104406
7141 044630      ENDSUB      ;///////////////// END SUBTEST ///////////////////
                                L10063:
                                TRAP      C#ESUB
7142 044630 104403
7143 044632 005737 002222      TST      FATFLG      ;ANY FATAL ERRORS ?
7144 044636 001402      BEQ      3004 ;BRANCH IF NOT
7145 044640 004737 017202      JSR      PC,CKDROP ;TRY TO DROP THE UNIT
7146 044644      3004:

```

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

```

; **
; TEST 6: SUBTEST 6:

```

; SUBTEST DESCRIPTION:

```

; This subtest verifies that the Reset FIFO function within
; the Write Miscellaneous Control 1 function initializes
; the FIFO to correct initial status. The following steps
; are performed:
; 1. Reset an already initialized FIFO and check for
;    proper status.
; 2. Write a varying number of bytes (1-65.) into the
;    FIFO and verify that after each block of bytes is
;    written the FIFO can be reset to it's initial
;    state.

```

```

7168      ; TEST STEPS:
7169      ;
7170      ; BEGIN
7171      ;       Write to TSSR to soft initialize
7172      ;       Do a WRITE CHARACTERISTICS to setup a message buffer
7173      ;       Do a Write Subsystem Write Misc to Reset FIFO
7174      ;       Do a Write Subsystem READ STATUS
7175      ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
7176      ;       signals NOT=0 Then Print Error
7177      ;       Do a Write Subsystem WRITE NPR to set tape direction out
7178      ;
7179      ; REPEAT FOR BYTE COUNT 1 TO 65.
7180      ; BEGIN
7181      ;       Do a Write Subsystem WRITE FIFO with the current byte count
7182      ;       Do a Write Subsystem Write Misc to Reset FIFO
7183      ;       Do a Write Subsystem READ STATUS
7184      ;       If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
7185      ;       signals NOT=0 Then Print Error
7186      ; END
7187      ;--
7188      044644      BGNSUB                      ;//////////////// BEGIN SUBTEST //////////////////
7189      044644      T6.6:                      TRAP      C0BSUB
7190      044644      104402
7191      ;       Write to TSSR register to soft initialize the controller
7192      50:      JSR      PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
7193      044646      004737      015774      BCS      100      ;BR IF SOFT INIT OKAY
7194      044652      103405      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
7195      044654      010001      ERDIF      ERNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
7196      044656      104455      TRAP      C0ERDF
7197      044660      001201      .WORD      641
7198      044662      003652      .WORD      SFIERR
7199      044664      012034      .WORD      SFIMSG
7200      ;       Do a WRITE CHARACTERISTICS to setup a message buffer
7201      100:      CLR      FATFLG              ;CLEAR FATAL ERROR FLAG
7202      044666      005037      002222      MOV      @T17PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
7203      044672      012704      047550      JSR      PC,WRITCMD ;DO WRITE CHARACTERISTICS COMMAND
7204      044676      004737      010662      FORCERROR 420 ;BDFORCE ERROR IF FORCER=1
7205      044702      103407      BCS      500      ;BR IF CARRY SET (GOOD RETURN)
7206      044716      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
7207      044722      044722      NEXT.ERRNO
7208      044724      104455      420:      ERDIF      ERNO,T17SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
7209      044726      001202      TRAP      C0ERDF
7210      044728      046175      .WORD      642
7211      044730      012046      .WORD      T17SSR
7212      044732      005237      002222      .WORD      PKTSSR
7213      044736      104406      500:      INC      FATFLG              ;SET FATAL ERROR FLAG
7214      044738      004737      002222      CKLOOP              ;LOOP ON ERROR, IF FLAG SET
7215      044740      004737      002222      TRAP      C0CLP1
7216      ;       Do a Write Subsystem Write Misc to Reset FIFO
7217      7207      JSR      PC,T17R5FIF          ;SETUP PNT FOR WRITE MISC RESET FIFO
7218      7208      044740      004737      047354      MOV      @T17PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
7219      7209      044744      012704      047720      MOV      R4,TSD0(R5) ;SET THE PACKET ADDRESS TO EXECUTE
7220      7210      044750      010465      000000      JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
7221      7211      044754      004737      016336      FORCERROR 620 ;BDFORCE ERROR IF FORCER=1
7222      7212      044760      103407      BCS      700      ;BR IF CARRY SET (GOOD RETURN)
7223      7213      044774      103407

```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```

7214 044776 010001      MOV      RO,R1      ;SAVE CONTENTS OF TSSR
7215 045000      NEXT.ERRNO
7216 045000      ( 1:  ERRDF  ERRNO,T172SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C:ERDF
                                .WORD    643
                                .WORD    T172SSR
                                .WORD    PKTSSR
7217 045010 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
7218 045014      70$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C:CLP1
7219 045014 104406
7220      ;      Do a Write Subsystem READ STATUS
7221      ;      If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
7222      ;      signals NOT=0 Then Print Error
7223 045016 004737 047334      JSR      PC,T17SRD      ;SETUP PACKET FOR READ STATUS
7224 045022 012704 047720      MOV      #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
7225 045026 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
7226 045032 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
7227 045036      FORCERROR 7$      ;B0DFORCE ERROR IF FORCER=1
7228 045052 103407      BCS      80$      ;BR IF CARRY SET (GOOD RETURN)
7229 045054 010001      MOV      RO,R1      ;SAVE CONTENTS OF TSSR
7230 045056      NEXT.ERRNO
7231 045056      77$:  ERRDF  ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C:ERDF
                                .WORD    644
                                .WORD    T173SSR
                                .WORD    PKTSSR
7232 045066 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
7233 045072      80$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C:CLP1
7234 045074 004737 047516      JSR      PC,T17SETEXP      ;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
7235 045100 012701 045752      MOV      #T17EXSTA,R1      ;GET EXPECTED READ STATUS
7236 045104 012702 047612      MOV      #T17BFSTA,R2      ;GET RCV READ STATUS
7237 045110 011211      MOV      (R2),(R1)      ;SET EXPD WORD #8 = RCV TEMP
7238 045112 042711 002000      BIC      #S1.ICER,(R1)      ;SET EXPD ICER =0
7239 045116 042711 001000      BIC      #S1.IFMK,(R1)      ;SET EXPD IFMK =0
7240 045122 042711 000400      BIC      #S1.IHER,(R1)      ;SET EXPD IHER =0
7241 045126 016261 000002 000002      MOV      2(R2),2(R1)      ;SET EXPD WORD #9 = RCV (NOT TESTING)
7242 045134 005000      CLR      RO      ;HIGH: RCV ADDRESS FOR CKMSG2
7243 045136 012701 047572      MOV      #T17BFR,R1      ;LOW RCV ADDRESS FOR CKMSG2
7244 045142 012702 045732      MOV      #T17EXP,R2      ;EXPD ADDRESS
7245 045146 012703 000024      MOV      #20,R3      ;NUMBER OF BYTES TO COMPARE
7246 045152 004737 011500      JSR      PC,CKMSG2      ;EXPD EQUAL RCV?
7247 045156      FORCERROR 92$,NOTSSR      ;B0D
7248 045166 103404      BCS      100$      ;BR IF YES
7249 045170      NEXT.ERRNO
7250 045170      92$:  ERRHRD  ERRNO,T177CMP,MSGSTAT ;REPORT ERROR
                                TRAP      C:ERHRD
                                .WORD    645
                                .WORD    T177CMP
                                .WORD    MSGSTAT
7251 045200      100$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C:CLP1
7252 045200 104406
7253      ;      Do a Write Subsystem WRITE NPR to set tape direction out
7254 045202 012700 000100      MOV      #NPR.OUT,RO      ;SET TAPE DIRECTION OUT
7255 045206 004737 047376      JSR      PC,T17SNPR      ;SETUP T17PK2 FOR WRITE NPR

```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```

7256 045212 012704 047720      MOV    #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
7257 045216 010465 000000      MOV    R4,TSD8(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
7258 045222 004737 016336      JSR     PC,CHKTSSR    ;WAIT FOR SSR TO SET
7259 045226      FORCERROR 112$      ;BDFORCE ERROR IF FORCER=1
7260 045242 103407      BCS     120$      ;BR IF CARRY SET (GOOD RETURN)
7261 045244 010001      MOV     R0,R1      ;SAVE CONTENTS OF TSSR
7262 045246      NEXT.ERRNO
7263      112$: ERRDF  ERRNO,T174SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP    C$ERDF
                                .WORD   646
                                .WORD   T174SSR
                                .WORD   PKTSSR
                                TRAP    C$CLP1
                                7264 045256 005237 002222      INC     FATFLG      ;SET FATAL ERROR FLAG
7265 045262      120$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP    C$CLP1
                                7266 045262 104406
7266      ; Setup incrementing pattern in FIFO data buffer
7267      MOV    #T17EXSTA,R1      ;EXPD WRITE FIFO DATA BUFFER
7268 045264 012701 045752      MOV    #64.,R2      ;TEST PATTERN SIZE
7269 045270 C12702 000100      CLR     R0      ;INCREMENT TEST PATTERN
7270 045274 005000      130$: MOVB   R0,(R1).      ;STORE INCREMENT TEST BYTE
7271 045276 110021      INC     R0      ;SET NEXT PATTERN
7272 045300 005200      DEC     R2      ;DONE?
7273 045302 005302      BGT     130$      ;BR IF NO
7274 045304 003374
7275      ; REPEAT FOR BYTE COUNT 1 TO 65.
7276      MOV    #1,COUNT      ;GET FIRST BYTE COUNT
7277 045306 012737 000001 002310 ; Do a Write Subsystem WRITE FIFO with the current byte count
7278      150$: MOV    COUNT,R0      ;REPEAT LOOP LABEL
7279 045314      MOV    #T17EXSTA,R1      ;FIFO BYTE COUNT
7280 045314 013700 002310      MOV    PC,T17WFIF      ;FIFO WRITE DATA ADDRESS
7281 045320 012701 045752      JSR     PC,T17PK2,R4      ;SETUP T17PK2 FOR WRITE FIFO
7282 045324 004737 047422      MOV    R4,TSD8(R5)    ;GET WRITE SUBSYSTEM COMMAND PACKET
7283 045330 012704 047720      MOV    R4,TSD8(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
7284 045334 010465 000000      JSR     PC,CHKTSSR    ;WAIT FOR SSR TO SET
7285 045340 004737 016336      FORCERROR 152$      ;BDFORCE ERROR IF FORCER=1
7286 045344      BCS     160$      ;BR IF CARRY SET (GOOD RETURN)
7287 045360 103407      MOV     R0,R1      ;SAVE CONTENTS OF TSSR
7288 045362 010001      NEXT.ERRNO
7289 045364      152$: ERRDF  ERRNO,T175SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
7290 045364      104455      TRAP    C$ERDF
7291 045374 005237 002222      .WORD   647
7292 045400      160$: INC     FATFLG      ;SET FATAL ERROR FLAG
7293 045400 104406      .WORD   T175SSR
7294      160$: CKLOOP      .WORD   PKTSSR
7295      ; Do a Write Subsystem Write Misc to Reset FIFO
7296 045402 004737 047354      TRAP    C$CLP1
7297 045406 012704 047720      JSR     PC,T17R5FIF      ;SETUP PKT FOR WRITE MISC RESET FIFO
7298 045412 010465 000000      MOV    #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
7299 045422 004737 016336      MOV    R4,TSD8(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
7300 045436 103407      JSR     PC,CHKTSSR    ;WAIT FOR SSR TO SET
7301 045440 010001      FORCERROR 162$      ;BDFORCE ERROR IF FORCER=1
7302 045442      BCS     170$      ;BR IF CARRY SET (GOOD RETURN)
                                MOV     R0,R1      ;SAVE CONTENTS OF TSSR
                                NEXT.ERRNO

```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```

7303 045442      162$: ERRDF  ERRNO,T172SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
      045442      104455                                TRAP      C#ERDF
      045444      001210                                .WORD      648
      045446      046232                                .WORD      T172SSR
      045450      012046                                .WORD      PKTSSR
7304 045452      005237 002222      170$: INC      FATFLG      ;SET FATAL ERROR FLAG
7305 045456      104406      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      045456      104406                                TRAP      C#CLP1

7306
7307      ;      Do a Write Subsystem READ STATUS
7308      ;      If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
7309      ;      signals NOT=0 Then Print Error
7310 045460      004737 047334      JSR      PC,T17SRD      ;SETUP PACKET FOR READ STATUS
7311 045464      012704 047720      MOV      #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
7312 045470      010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
7313 045474      004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
7314 045500      FORCERROR      177$      ;DO FORCE ERROR IF FORCER=1
7315 045514      103407      BCS      180$      ;BR IF CARRY SET (GOOD RETURN)
7316 045516      C10001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
7317 045520      NEXT.ERRNO
7318      177$: ERRDF  ERRNO,T173SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
      045520      104455                                TRAP      C#ERDF
      045522      001211                                .WORD      649
      045524      046276                                .WORD      T173SSR
      045526      012046                                .WORD      PKTSSR
7319 045530      005237 002222      180$: INC      FATFLG      ;SET FATAL ERROR FLAG
7320 045534      104406      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      045534      104406                                TRAP      C#CLP1
7321 045536      004737 047516      JSR      PC,T17SETEXP      ;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
7322 045542      012701 045752      MOV      #T17EXSTA,R1      ;GET EXPECTED READ STATUS
7323 045546      012702 047612      MOV      #T17BFSTA,R2      ;GET RCV READ STATUS
7324 045552      011211      MOV      (R2),(R1)      ;SET EXPD WORD #8 = RCV TEMP
7325 045554      042711 002000      BIC      #S1.ICER,(R1)      ;SET EXPD ICER =0
7326 045560      042711 001000      BIC      #S1.IFMK,(R1)      ;SET EXPD IFMK =0
7327 045564      042711 000400      BIC      #S1.IHER,(R1)      ;SET EXPD IHER =0
7328 045570      016261 000002 000002      MOV      2(R2),2(R1)      ;SET EXPD WORD #9 = RCV (NOT TESTING)
7329 045576      005000      CLR      R0      ;HIGH RCV ADDRESS FOR CKMSG2
7330 045600      012701 047572      MOV      #T17BFR,R1      ;LOW RCV ADDRESS FOR CKMSG2
7331 045604      012702 045732      MOV      #T17EXP,R2      ;EXPD ADDRESS
7332 045610      012703 000024      MOV      #20,R3      ;NUMBER OF BYTES TO COMPARE
7333 045614      004737 011500      JSR      PC,CKMSG2      ;EXPD EQUAL RCV?
7334 045620      FORCERROR      192$,NOTSSR      ;DO
7335 045630      103404      BCS      200$      ;BR IF YES
7336 045632      NEXT.ERRNO
7337      192$: ERRHRD  ERRNO,T177CMP,MSGSTAT  ;REPORT ERROR
      045632      104456                                TRAP      L#ERHRD
      045634      001212                                .WORD      650
      045636      047224                                .WORD      T177CMP
      045640      012350                                .WORD      MSGSTAT
7338 045642      200$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      045642      104406                                TRAP      C#CLP1

7339
7340      250$:
7341 045644      FORCEEXIT      260$
7342 045644      INC      COUNT
7343 045654      005237 002310      CMP      COUNT,#65.      ;GET NEXT BYTE COUNT
7344 045660      023727 002310 000101      ;DONE ALL BYTES?

```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```

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

```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```

7398 046452      127      122      111 T176SSR:.ASCIZ 'WRITE SUBSYSTEM (Read FIFO) Failed'
7399 046515      106      111      106 T171CMP:.ASCIZ 'FIFO Status in WORD #9 Incorrect after Initialize'
7400 046577      122      145      141 T172CMP:.ASCIZ 'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
7401 046673      106      111      106 T173CMP:.ASCIZ 'FIFO Status (In WORD #9) Incorrect after WRITE FIFO'
7402 046757      106      111      106 T174CMP:.ASCIZ 'FIFO Status (In WORD #9) Incorrect after READ FIFO'
7403 047042      122      145      141 T175CMP:.ASCIZ 'Read FIFO Data not equal to Write FIFO Data'
7404 047116      106      111      106 T176CMP:.ASCIZ 'FIFO Status (In WORD #9) Incorrect after READ FIFO from an Empty FIFO'
7405 047224      106      111      106 T177CMP:.ASCIZ 'FIFO Status (In WORD #9) Incorrect after RESET FIFO'
7406                                     .EVEN
7407
7408
7409      ;* CLEAR MESSAGE BUFFER
7410      ;~
7411      T17CLRBUF:
7412      SAVREG
7413      MOV      #T17BFR,R1
7414      MOV      #T17BEND-T17BFR,R2
7415      10+:    CLRB      (R1)+
7416      DEC      R2
7417      BGT      10+
7418      RTS      PC
7419
7420      ;* SETUP T17PK2 PACKET FOR READ STATUS
7421      ;~
7422      T17SRD:
7423      JSR      PC,T17CLRBUF
7424      MOV      #T17DT2,R0
7425      MOV      #PW.RDSTATUS,(R0)+
7426      CLRB      (R0)
7427      RTS      PC
7428
7429      ;* SETUP T17PK2 PACKET FOR WRITE MISC RESET FIFO
7430      ;~
7431      T17RSFIF:
7432      JSR      PC,T17CLRBUF
7433      MOV      #T17DT2,R0
7434      MOV      #PW.WMISC,(R0)+
7435      MOV      #MS.RSFIF!MS.RSTAP,(R0)
7436      RTS      PC
7437
7438      ;* SETUP T17PK2 PACKET FOR WRITE NPR
7439      ;~
7440      INPUT:
7441      ; RO CONTAINS BSEL1 NPR DATA
7442      ;
7443      ; SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
7444      ;~
7445      T17SNPR:
7446      JSR      PC,T17CLRBUF
7447      MOV      #T17DT2,R1
7448      MOV      #PW.WNPR,(R1)+
7449      BIS      #NP.WRP,R0
7450      MOV      R0,(R1)
7451      RTS      PC
7452
7453
7454

```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```

7455
7456
7457      ;*
7458      ; SETUP T17PK2 PACKET FOR WRITE FIFO
7459      ;
7460      ; INPUT:
7461      ;       R0 CONTAINS BYTE COUNT
7462      ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
7463      ;*
7464      ; T17WFIF:
7465      ; SAVREG      ;SAVE R1-R5 UNTIL NEXT RETURN
7466      ; JSR        PC,T17CLRBUF      ;CLEAR MESSAGE BUFFER
7467      ; MOV        #T17DT2,R2      ;WRITE SUBSYSTEM DATA BUFFER
7468      ; MOVB       #PW.WFIFO,(R2)+ ;STORE WRITE FIFO IN BSEL0
7469      ; MOVB       R0,(R2)+        ;STORE BYTE COUNT IN BSEL1
7470      ; CLR        (R2)+          ;CLEAR SEL2 (UNUSED)
7471      ; MOVB       (R1)+,(R2)+    ;STORE DATA PATTERN BYTE
7472      ; DEC        R0              ;DONE ALL BYTES?
7473      ; BGT        10$            ;BR IF NO
7474      ; RTS         PC             ;RETURN
7475
7476      ;*
7477      ; SETUP T17PK2 PACKET FOR READ FIFO
7478      ;
7479      ; INPUT:
7480      ;       R0 CONTAINS SEL2 BYTE COUNT
7481      ;*
7482      ; T17RFIF:
7483      ; JSR        PC,T17CLRBUF      ;CLEAR MESSAGE BUFFER
7484      ; MOV        #T17DT2,R1      ;WRITE SUBSYSTEM DATA BUFFER
7485      ; MOVB       #PW.RFIFO,(R1)+ ;STORE READ FIFO IN BSEL0
7486      ; MOVB       R0,(R1)+        ;STORE BYTE COUNT IN BSEL1
7487      ; RTS         PC             ;RETURN
7488
7489      ;*
7490      ; CLEAR EXPECTED DATA MESSAGE BUFFER
7491      ;*
7492      ; T17CLEXP:
7493      ; MOV        #T17EXP,R1        ;GET EXPD ADDRESS
7494      ; MOV        #T17EXEND-T17EXP,R0 ;GET EXPD SIZE
7495      ; CLR        (R1)+            ;CLEAR A BYTE
7496      ; DEC        R0              ;DONE?
7497      ; BGT        10$            ;BR IF NO
7498      ; RTS         PC             ;RETURN
7499
7500      ;*
7501      ; Set WORDS 0-7 of expd message buffer = to recv since not testing
7502      ;*
7503      ; T17SETEXP:
7504      ; MOV        #T17EXP,R2        ;GET EXPD
7505      ; MOV        #T17BFR,R3      ;GET READ STATUS RECV BUFFER
7506      ; MOV        #8,R0           ;SET WORDS 0-7 EXP=RECV
7507      ; MOVB       (R3)+,(R2)+    ;SET EXPD=RECV
7508      ; DEC        R0              ;DONE WORDS 0-7 WORDS?
7509      ; BGT        5$             ;BR IF NO
7510      ; RTS         PC             ;RETURN
7511      ;
7512      ; .=<. +10>&177770
7513      ;

```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```

7514      ;WRITE CHARACTERISTICS COMMAND PACKET
7515      ;
7516 047550      T17PACKET:      ;COMMAND PACKET FOR TEST
7517 047550 100004      .WORD 100004      ;WRITE CHARACTERISTICS COMMAND, WITH ACK
7518 047552 047560      .WORD T17DATA      ;ADDRESS OF CHARACTERISTICS BLOCK
7519 047554 000000      .WORD 0
7520 047556 000012      .WORD 10.      ;MINIMUM MESSAGE PACKET SIZE
7521
7522 047560      T17DATA:      ;CHARACTERISTICS DATA BLOCK
7523 047560 047572      .WORD T17BFR      ;ADDRESS OF MESSAGE BUFFER
7524 047562 000000      .WORD 0
7525 047564 000024      .WORD 20.      ;LENGTH OF MESSAGE BUFFER
7526 047566 000000      .WORD 0      ;ESS,ENB,EAI,ERI
7527 047570 000000      .WORD 0      ;EXTENDED FEATURES UNIT NO. ETC.
7528
7529
7530      ;MESSAGE BUFFER FOR ALL TEST 6 COMMANDS
7531
7532 047572      T17BFR:      ;BEGIN MESSAGE BUFFER
7533 047572 000000      .WORD 0      ;MESSAGE TYPE
7534 047574 000000      .WORD 0      ;DATA FIELD LENGTH
7535 047576 000000      .WORD 0      ;RBPGR
7536 047600 000000      .WORD 0      ;XST0
7537 047602 000000      .WORD 0      ;XST1
7538 047604 000000      .WORD 0      ;XST2
7539 047606 000000      .WORD 0      ;XST3
7540 047610 000000      .WORD 0      ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
7541 047612      T17BFSTA: .BLKB 64.      ;READ STATUS AND WRITE FIFO BUFFER
7542 047712      T17BEND:      ;END OF MESSAGE BUFFER
7543
7544      ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
7545      ;
7546      .=<.>10>&177770
7547 047720      T17PK2:      ;WRITE SUBSYSTEM WITH ACK
7548 047720 100006      .WORD P.WRTSUB!P.ACK      ;LOW ADDRESS OF DATA BLOCK
7549 047722 047730      .WORD T17DT2      ;HIGH ADDRESS OF DATA BLOCK
7550 047724 000000      .WORD 0      ;MINIMUM MESSAGE PACKET SIZE
7551 047726 000012      .WORD 10.
7552
7553 047730      T17DT2:      ;DATA BLOCK
7554 047730 000      .BYTE 0      ;BSEL0
7555 047731 000      .BYTE 0      ;BSEL1
7556 047732 000000      .WORD 0      ;SEL2
7557 047734      .BLKB 66.      ;WRITE FIFO DATA OUTPUT BUFFER
7558
7559
7560      ENDTST
7561 050036
7562 050036      L10056:      TRAP      C#ETST
7563 050036 104401
7564
7565      .SBTTL TEST 7: STATIC TRANSPORT BUS INTERFACE TEST
7566      ;*
7567      ; TEST DESCRIPTION:
7568      ;
7569      ; TEST STEPS:
7570      ; REPEAT FOR LOOPCNT

```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```

7571      ; BEGIN
7572      ; Write to TSSR register to soft initialize the controller
7573      ; Do WRITE CHARACTERISTICS to check for Extended Features Switch
7574      ; If Extended Features Hardware Switch Clear then:
7575      ;     Do Write Subsystem Write Miscellaneous to Set Extended Features.
7576      ; Do WRITE CHARACTERISTICS to select reserved unit 7
7577      ; Do a Write Subsystem READ STATUS
7578      ; If any transport interface signals are asserted then Print Error
7579      ; END
7580      ; --
7581
7582
7583      050040      BGNTST
7584      050040
7588      050040      012700      050546      MOV      #TST18ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
7589      050044      004737      016510      JSR      PC,TSTSETUP      ;DO INITIAL TEST SETUP
7590      050050      012737      000012      002216      MOV      #10.,LOOPCNT      ;PERFORM 10 ITERATIONS
7591      050056      T18LOOP:
7592      ; Write to TSSR register to soft initialize the controller
7593      5$:
7594      050056      004737      015774      JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
7595      050062      103405      BCS      10$      ;BR IF SOFT INIT OKAY
7596      050064      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
7597      050066      ERRDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
7598      050066      104455      TRAP      C#ERDF
7599      050070      001274      .WORD      700
7600      050072      003652      .WORD      SFIERR
7601      050074      012034      .WORD      SFIMSG
7602
7603      ; Do WRITE CHARACTERISTICS to check for Extended Features Switch
7604      10$:
7605      050076      005037      002222      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
7606      050102      012704      051230      MOV      #T18PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
7607      050106      004737      010662      JSR      PC,WRTCHR      ;DO WRITE CHARACTERISTICS COMMAND
7608      050112      FORCERROR      12$      ;GOODFORCE ERROR IF FORCER=1
7609      050126      103407      BCS      15$      ;BR IF CARRY SET (GOOD RETURN)
7610      050130      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
7611      050132      NEXT.ERRNO
7612      050132      12$:
7613      050132      104455      ERRDF      ERRNO,T18SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
7614      050134      001275      TRAP      C#ERDF
7615      050136      050605      .WORD      701
7616      050140      012046      .WORD      T18SSR
7617      050142      005237      002222      .WORD      PKTSSR
7618      050146      104406      15$:
7619      050146      INC      FATFLG      ;SET FATAL ERROR FLAG
7620      050146      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
7621      050146      TRAP      C#CLP1
7622
7623      ; If Extended Features Hardware Switch Clear then:
7624      ; Do Write Subsystem Write Miscellaneous to Set Extended Features.
7625      050150      012701      051252      MOV      #T18BFR,R1      ;MESSAGE BUFFER ADDRESS
7626      050154      032761      000200      000012      BIT      #X2.EXTF,XST2(R1)      ;EXTENDED FEATURES SWITCH SET?
7627      050162      001026      BNE      30$      ;BR IF YES
7628      050164      004737      051076      JSR      PC,T18SMISC      ;SETUP PACKET FOR WRITE MISCELLANEOUS
7629      050170      012704      051300      MOV      #T18PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
7630      050174      010465      000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
7631      050200      004737      016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
7632      050204      FORCERROR      22$      ;GOODFORCE ERROR IF FORCER=1
7633      050220      103407      BCS      30$      ;BR IF CARRY SET (GOOD RETURN)

```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```

7622 050222 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
7623 050224      NEXT.ERRNO
7624 050224      22$:  ERRDF  ERRNO,T182SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
                                TRAP  C$ERDF
                                .WORD 702
                                .WORD T182SSR
                                .WORD PKTSSR
                                7625 050234 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
                                7626 050240      30$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                                TRAP  C$CLP1
7627
7628
7629      ;      Do WRITE CHARACTERISTICS to select reserved unit 7
7630 050242 005037 002222      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
7631 050246 012704 051230      MOV      #T18PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
7632 050252 004737 010662      JSR      PC,WRTCHR      ;DO WRITE CHARACTERISTICS COMMAND
7633 050256      FORCERROR 42$      ;B0DFORCE ERROR IF FORCER=1
7634 050272 103407      BCS      50$      ;BR IF CARRY SET (GOOD RETURN)
7635 050274 C10001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
7636 050276      NEXT.ERRNO
7637 050276      42$:  ERRDF  ERRNO,T18SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
                                TRAP  C$ERDF
                                .WORD 703
                                .WORD T18SSR
                                .WORD PKTSSR
                                7638 050306 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
                                7639 050312      50$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                                TRAP  C$CLP1
7640
7641      ;      Clear message buffer
7642 050314 012701 051252      MOV      #T18BFR,R1      ;GET MESSAGE BUFFER ADDRESS
7643 050320 013700 051244      MOV      T18DATA+4,R0      ;SIZE OF MESSAGE BUFFER IN BYTES
7644 050324 105021      60$:  CLRB      (R1)+      ;CLEAR A BYTE
7645 050326 005300      DEC      R0      ;DONE?
7646 050330 003375      BGT      60$      ;BR IF NO
7647      ;      Do a Write Subsystem READ STATUS
7648 050332 004737 051056      JSR      PC,T18SRD      ;SETUP PACKET FOR READ STATUS
7649 050336 012704 051300      MOV      #T18PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
7650 050342 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
7651 050346 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
7652 050352      FORCERROR 62$      ;B0DFORCE ERROR IF FORCER=1
7653 050366 103407      BCS      70$      ;BR IF CARRY SET (GOOD RETURN)
7654 050370 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
7655 050372      NEXT.ERRNO
7656 050372      62$:  ERRDF  ERRNO,T183SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
                                TRAP  C$ERDF
                                .WORD 704
                                .WORD T183SSR
                                .WORD PKTSSR
                                7657 050402 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
                                7658 050406      70$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                                TRAP  C$CLP1
7659
7660
7661      ;      Set first 8 words of expd message buffer = to recv since not testing
7662      ;      Set unused bits in Read Status expd equal rcvd
7663 050410 004737 051120      JSR      PC,T18SETEXP      ;SET SOME EXPD TO RECV

```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```

7664      ;      If any transport interface signals are asserted then Print Error
7665 050414 005000      CLR      R0      ;HIGH RECV ADDRESS FOR CKMSG2
7666 050416 012701 051252      MOV      #T18BFR,R1      ;LOW RECV ADDRESS FOR CKMSG2
7667 050422 012702 050516      MOV      #T18EXP,R2      ;EXPD ADDRESS
7668 050426 012703 000012      MOV      #10.,R3      ;NUMBER OF WORDS TO COMPARE
7669 050432 004737 011500      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
7670 050436      FORCERROR      82$,NOTSSR      ;880
7671 050446 103404      BCS      90$      ;BR IF YES
7672 050450      NEXT.ERRNO
7673 050450      82$:      ERRHRD      ERRNO,T18CMP,MSGSTAT      ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD      705
                                .WORD      T18CMP
                                .WORD      MSGSTAT
7674 050460      90$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
7675 050460 104406
7676 050462 005737 002222      TST      FATFLG      ;ANY FATAL ERRORS ?
7677 050466 C01402      BEQ      160$      ;BRANCH IF NOT
7678 050470 004737 017202      JSR      PC,CKDROP      ;TRY TO DROP THE UNIT
7679 050474 004737 016456      160$:      JSR      PC,TSTLOOP      ;DO ITERATIONS?
7680 050500 103002      BCC      165$      ;BR IF NO
7681 050502 000137 050056      165$:      JMP      T18LOOP      ;LOOP UNTIL ITERATIONS DONE
7682 050506      EXIT      TST
7683 050506 104432      TRAP      C$EXIT
7684 050510 000606      .WORD      L10065-.
7685
7686      ;*
7687      ;LOCAL STORAGE FOR THIS TEST
7688      ;-
7689
7690 050512      T18MSK:      ;MASK OF UNUSED BITS IN READ STATUS BYTES
7691 050512      .BYTE      +C<000>      ;BYTE 0 MASK
7692 050513      .BYTE      +C<340>      ;BYTE 1
7693 050514      .BYTE      +C<277>      ;BYTE 2
7694 050515      .BYTE      0      ;MAKE IT EVEN
7695
7696 050516      T18EXP:      ;EXPECTED DATA BUFFER
7697 050516      .WORD      0      ;MESSAGE TYPE
7698 050520      .WORD      0      ;DATA FIELD LENGTH
7699 050522      .WORD      0      ;RBPCR
7700 050524      .WORD      0      ;XST0
7701 050526      .WORD      0      ;XST1
7702 050530      .WORD      0      ;XST2
7703 050532      .WORD      0      ;XST3
7704 050534      .WORD      0      ;XST4 (ALWAYS PRESENT FOR WRITE SUB)
7705 050536      .WORD      0      ;READ STATUS BYTE 1/0
7706 050540      .WORD      0      ;READ STATUS BYTE 2
7707
7708 050542      377      020      T18XS:      .BYTE      377,020      ;READ STATUS BYTE 0/1 EXPECTED BASE
7709 050544      000000      .WORD      0      ;READ STATUS BYTE 2 EXPECTED BASE
7710
7711      ;*
7712      ;LOCAL TEXT MESSAGES FOR TEST
7713      ;-

```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```

7714
7715 050546      123      164      141 TST18ID:      .ASCIZ 'Static Transport Bus Interface'
7716 050605      127      122      111 T18SSR: .ASCIZ 'WRITE CHARACTERISTICS Failed'
7717 050642      127      122      111 T182SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Misc) Failed'
7718 050706      127      122      111 T183SSR: .ASCIZ 'WRITE SUBSYSTEM (Read Status) Failed'
7719 050753      124      162      141 T18CMP: .ASCIZ 'Transport Bus Interface Signals NOT Negated After Unit 7 Selected'
7720                      .EVEN
7721
7722
7723      ;*
7724      ; SETUP T18PK2 PACKET FOR READ STATUS
7725      ;-
7725 051056      T18SRD:
7726 051056                      SAVREG                      ;SAVE R1-R5 UNTIL NEXT RETURN
7727 051062      012700      051310      MOV      #T18DT2,R0      ;WRITE SUBSYSTEM DATA BUFFER
7728 051066      112720      000005      MOVW     #PW.RDSTATUS,(R0)+ ;STORE READ STATUS COMMAND IN BSEL0
7729 051072      105010      CLRW      (R0)      ;CLEAR BSEL1
7730 051074      000207      RTS      PC      ;RETURN
7731
7732
7733      ;*
7734      ; SETUP T18PK2 PACKET FOR WRITE MISC.
7735      ;-
7735 051076      T18SMISC:
7736 051076                      SAVREG                      ;SAVE R1-R5 UNTIL NEXT RETURN
7737 051102      012700      051310      MOV      #T18DT2,R0      ;WRITE SUBSYSTEM DATA BUFFER
7738 051106      112720      000010      MOVW     #PW.WMISC,(R0)+ ;STORE WRITE MISCELLANEOUS IN BSEL0
7739 051112      112710      000200      MOVW     #MS.EXT,(R0)    ;STORE INVERT EXTENDED FEATURES IN BSEL1
7740 051116      000207      RTS      PC      ;RETURN
7741
7742
7743      ;*
7744      ;Set first 8 words of expd message buffer = to rcv since not testing
7745      ; Set unused bits in Read Status expd equal rcvd
7746      ;-
7746 051120      T18SETEXP:
7747 051120      012702      050516      MOV      #T18EXP,R2      ;GET EXPD
7748 051124      012703      051252      MOV      #T18BFR,R3      ;GET READ STATUS RECV BUFFER
7749 051130      012700      000010      MOV      #8,R0      ;SET FIRST 8 WORDS EXP=RCV
7750 051134      012322      S4:      MOV      (R3)+,(R2)+      ;SET EXPD=RCV
7751 051136      005300      DEC      R0      ;DONE FIRST 8 WORDS?
7752 051140      003375      BGT      S4      ;BR IF NO
7753 051142      012701      050512      MOV      #T18MSK,R1      ;GET UNUSED BIT MASK
7754 051146      013712      050542      MOV      T18XS,(R2)      ;SETUP BASE EXPECTED BYTE 1/0
7755 051152      013762      050544      000002      MOV      T18XS+2,2(R2) ;SETUP BASE EXPECTED BYTE 2
7756 051160      011300      MOV      (R3),R0      ;GET RECV BYTE 1 AND BYTE 0
7757 051162      041100      BIC      (R1),R0      ;CLEAR ALL BUT UNUSED
7758 051164      040012      BIC      R0,(R2)      ;CLEAR UNUSED IN EXP
7759 051166      050012      BIS      R0,(R2)      ;SET UNUSED EXPD=RCV FOR COMPARE
7760 051170      016300      000002      MOV      2(R3),R0      ;GET RECV BYTE 2
7761 051174      046100      000002      BIC      2(R1),R0      ;CLEAR ALL BUT UNUSED
7762 051200      040062      000002      BIC      R0,2(R2)      ;CLEAR UNUSED IN EXPD
7763 051204      050062      000002      BIS      R0,2(R2)      ;SET UNUSED EXPD=RCV FOR COMPARE
7764 051210      105062      000003      CLRB     3(R2)      ;CLEAR EXPD BYTE 3 (UNUSED)
7765 051214      105063      000003      CLRB     3(R3)      ;CLEAR RECV BYTE 3 (UNUSED)
7766 051220      000207      RTS      PC      ;RETURN
7767
7768
7769      .=<<.10>&177770
7770
7771      ;
7772      ;WRITE CHARATERISTICS COMMAND PACKET

```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```

7773
7774 051230      T18PACKET:
7775 051230      .WORD 100004      ;COMMAND PACKET FOR TEST
7776 051232 051240      .WORD T18DATA ;WRITE CHARACTERISTICS COMMAND, WITH ACK
7777 051234 000000      .WORD 0      ;ADDRESS OF CHARACTERISTICS BLOCK
7778 051236 000012      .WORD 10.    ;MESSAGE PACKET MINIMUM SIZE
7779
7780 051240      T18DATA:
7781 051240 051252      .WORD T18BFR  ;CHARACTERISTICS DATA BLOCK
7782 051242 000000      .WORD 0      ;ADDRESS OF MESSAGE BUFFER
7783 051244 000024      .WORD 20.    ;LENGTH OF MESSAGE BUFFER
7784 051246 000000      .WORD 0      ;ESS,ENB,EAI,ERI
7785 051250 000007      .WORD 7      ;SELECT RESERVED UNIT 7
7786
7787
7788 051252      T18BFR:
7789 051252 000000      .WORD 0      ;MESSAGE BUFFER
7790 051254 000000      .WORD 0      ;MESSAGE TYPE
7791 051256 000000      .WORD 0      ;DATA FIELD LENGTH
7792 051260 000000      .WORD 0      ;RBPGR
7793 051262 000000      .WORD 0      ;XST0
7794 051264 000000      .WORD 0      ;XST1
7795 051266 000000      .WORD 0      ;XST2
7796 051270 000000      .WORD 0      ;XST3
7797 051272 000000      .WORD 0      ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
7798 051274 000000      .WORD 0      ;READ STATUS BYTE 1/0 RETURNED
7799
7800      ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
7801
7803      .=<..+10>&177770
7805 051300      T18PK2:
7806 051300 100006      .WORD P.WRTSUB:P.ACK ;WRITE SUBSYSTEM WITH ACK
7807 051302 051310      .WORD T18DT2      ;LOW ADDRESS OF DATA BLOCK
7808 051304 000000      .WORD 0      ;HIGH ADDRESS OF DATA BLOCK
7809 051306 000010      .WORD 8.    ;BUFFER EXTENT
7810
7811 051310      T18DT2:
7812 051310 000      .BYTE 0      ;DATA BLOCK
7813 051311 000      .BYTE 0      ;BSELO
7814 051312 000000      .WORD 0      ;BSEL1
7815 051314 000000      .WORD 0      ;SEL2
7816
7817
7818 051316      ENDTST
7819 051316 104401
7820
7821      .SBTTL TEST 8: TRANSPORT BUS INTERFACE LOOPBACK TEST
7822
7823      ;**
7824      ;TEST DESCRIPTION:
7825      ;
7826      ;This test verifies the controller's Transport Bus
7827      ;drivers, receivers, and signal loopback logic. Note
7828      ;that the Static Transport Bus test must have run
7829      ;correctly for this test to provide meaningful results.
7830
7831      ;TEST STEPS:
7832      ;

```

L10065: TRAP L4ETST

TEST 8: TRANSPORT BUS INTERFACE LOOPBACK TEST

```

7830      ; REPEAT FOR LOOPCNT
7831      ; BEGIN
7832      ;   Do Subtest 1      - Loopback Control signals test
7833      ;   Do Subtest 2      - Loopback Read/Write signals test
7834      ;   Do Subtest 3      - Loopback Write Strobe test
7835      ;   Do Subtest 4      - Loopback Read Strobe test
7836      ; END
7837      ; --
7838
7839
7840 051320      BGNTST
7841 051320
7845 051320 012700 057532      MOV      #TST19ID,R0      T8::
7846 051324 004737 016510      JSR      PC,TSTSETUP      ;ASCII MESSAGE TO IDENTIFY TEST
7847 051330 012737 000012 002216  MOV      #10.,LOOPCNT      ;DO INITIAL TEST SETUP
7848 051336      T19LOOP:      ;PERFORM 10 ITERATIONS
7849
7850      .SBTTL TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST
7851
7852      ;**
7853      ; TEST 8: SUBTEST 1:
7854      ;
7855      ; SUBTEST DESCRIPTION:
7856      ;
7857      ;   This subtest verifies the Transport Control loopback
7858      ;   path can transmit and receive correctly. The
7859      ;   control signals are all loopback signals other
7860      ;   than the read/write data (IW<7:0> and IR<7:0>).
7861      ;
7862      ; TEST STEPS:
7863      ;
7864      ;   The loopback signals IFAD,ITAD0,ITAD1 are the tape unit select
7865      ;   lines. Since reserved unit 7 must remain selected these signals
7866      ;   are always set low. This further means the signals they drive
7867      ;   (ISPEED,IRDY,IONL) are only tested in the low state.
7868      ;
7869      ; BEGIN
7870      ;   Write to TSSR register to soft initialize the controller
7871      ;   Do WRITE CHARACTERISTICS to check for Extended Features Switch
7872      ;   If Extended Features Hardware Switch Clear then:
7873      ;       Do Write Subsystem Write Miscellaneous to Set Extended Features.
7874      ;   Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
7875      ;   Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
7876      ;   Do Write Subsystem Write Control to CLEAR loopback signals group 1.
7877      ;   Do Write Subsystem Write Format to CLEAR loopback signals group 2.
7878      ;   (the loopback signals have to be cleared here due to the flip-flops
7879      ;   that are set on a 1 to 0 transition (IHER,IFMK,ICER))
7880      ;   Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
7881      ;   Do a Write Subsystem READ STATUS
7882      ;   If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
7883      ;   signals NOT=0 Then Print Error
7884      ;
7885      ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
7886      ; BEGIN
7887      ;   Do Write Subsystem Write Control to Drive loopback signals group 1.
7888      ;   Do Write Subsystem Write Format to Drive loopback signals group 2.
7889      ;   Do a Write Subsystem READ STATUS

```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

7890      ;      If loopback data NOT= data sent Then Print Error
7891      ;      Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
7892      ;      Do a Write Subsystem READ STATUS
7893      ;      If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
7894      ;      signals NOT=0 Then Print Error
7895      ;      END
7896      ;--
7897 051336      BGNSUB                      ;//////////////// BEGIN SUBTEST //////////////////
      051336                      T8.1:
      051336 104402                      TRAP      C#BSUB

7898
7899      ;      Write to TSSR register to soft initialize the controller
7900 051340      5#:
7901 051340 004737 015774      JSR      PC,SO,INIT          ;WRITE TO TSSR TO SOFT INITIALIZE
7902 051344 103405      BCS      10#                      ;BR IF SOFT INIT OKAY
7903 051346 010001      MOV      R0,R1                      ;SAVE CONTENTS OF TSSR
7904 051350      ERRDF      ERRNO,SFIERR,SFIMSG          ;DEVICE FATAL DURING INIT
      051350 104455                      TRAP      C#ERDF
      051352 001440                      .WORD      800
      051354 003652                      .WORD      SFIERR
      051356 012034                      .WORD      SFIMSG

7905      ;      Do WRITE CHARACTERISTICS to check for Extended Features Switch
7906 051360 005037 002222      10#:      CLR      FATFLG          ;CLEAR FATAL ERROR FLAG
7907 051364 012704 061660      MOV      #T19PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
7908 051370 004737 010662      JSR      PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
7909 051374      FORCERROR 12#          ;#00FORCE ERROR IF FORCER=1
7910 051410 103407      BCS      15#          ;BR IF CARRY SET (GOOD RETURN)
7911 051412 010001      MOV      R0,R1          ;SAVE CONTENTS OF TSSR
7912 051414      NEXT.ERRNO
7913 051414      12#:      ERRDF      ERRNO,T19SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      051414 104455                      TRAP      C#ERDF
      051416 001441                      .WORD      801
      051420 057573                      .WORD      T19SSR
      051422 012046                      .WORD      PKTSSR

7914 051424 005237 002222      15#:      INC      FATFLG          ;SET FATAL ERROR FLAG
7915 051430      CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      051430 104406                      TRAP      C#CLP1

7916      ;      If Extended Features Hardware Switch Clear then:
7917      ;      Do Write Subsystem Write Miscellaneous to Set Extended Features.
7918 051432 012701 061702      MOV      #T19BFR,R1          ;MESSAGE BUFFER ADDRESS
7919 051436 032761 000200 000012      BIT      #X2.EXTF,XST2(R1)      ;EXTENDED FEATURES SWITCH SET?
7920 051444 001026      BNE      30#          ;BR IF YES
7921 051446 004737 061532      JSR      PC,T19SEXT          ;SETUP PACKET FOR WRITE MISC INVERT
7922 051452 012704 062030      MOV      #T19PK2,R4          ;GET WRITE SUBSYSTEM COMMAND PACKET
7923 051456 010465 000000      MOV      R4,TSD8(R5)          ;SET THE PACKET ADDRESS TO EXECUTE
7924 051462 004737 016336      JSR      PC,CHKTSSR          ;WAIT FOR SSR TO SET
7925 051466      FORCERROR 22#          ;#00FORCE ERROR IF FORCER=1
7926 051502 103407      BCS      30#          ;BR IF CARRY SET (GOOD RETURN)
7927 051504 010001      MOV      R0,R1          ;SAVE CONTENTS OF TSSR
7928 051506      NEXT.ERRNO
7929 051506      22#:      ERRDF      ERRNO,T192SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      051506 104455                      TRAP      C#ERDF
      051510 001442                      .WORD      802
      051512 057630                      .WORD      T192SSR
      051514 012046                      .WORD      PKTSSR

7930 051516 005237 002222      30#:      INC      FATFLG          ;SET FATAL ERROR FLAG
7931 051522      CKLOOP          ;LOOP ON ERROR, IF FLAG SET

```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

051522 104406                                TRAP    C4CLP1
7932      ; Do WRITE CHARACTERISTICS to select reserved unit 7
7933 051524 005037 002222      CLR    FATFLG      ;CLEAR FATAL ERROR FLAG
7934 051530 012704 061660      MOV    #T19PACKET,R4    ;GET THE ADDRESS OF COMMAND PACKET
7935 051534 004737 010662      JSR    PC,WRTCHR      ;DO WRITE CHARACTERISTICS COMMAND
7936 051540      FORCERROR      42$      ;DO FORCE ERROR IF FORCER=1
7937 051554 103407      BCS    50$      ;BR IF CARRY SET (GOOD RETURN)
7938 051556 010001      MOV    R0,R1      ;SAVE CONTENTS OF TSSR
7939 051560      NEXT.ERRNO
7940 051560 42$: ERRDF    ERRNO,T19SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP    C4ERDF
                                .WORD   803
                                .WORD   T19SSR
                                .WORD   PKTSSR
                                051560 104455
                                051562 001443
                                051564 057573
                                051566 012046
7941 051570 005237 002222      INC    FATFLG      ;SET FATAL ERROR FLAG
7942 051574 50$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP    C4CLP1
7943      ; Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
7944 051576 C12700 000100      MOV    #NP.OUT,R0      ;SET TAPE DIRECTION OUT
7945 051602 052700 000040      BIS    #NP.LOOP,R0      ;SET LOOPBACK ENABLE
7946 051606 004737 061372      JSR    PC,T19SNPR      ;SETUP T19PK2 FOR WRITE NPR
7947 051612 012704 062030      MOV    #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
7948 051616 010465 000000      MOV    R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
7949 051622 004737 016336      JSR    PC,CHKTSSR      ;WAIT FOR SSR TO SET
7950 051626      FORCERROR      62$      ;DO FORCE ERROR IF FORCER=1
7951 051642 103407      BCS    70$      ;BR IF CARRY SET (GOOD RETURN)
7952 051644 010001      MOV    R0,R1      ;SAVE CONTENTS OF TSSR
7953 051646      NEXT.ERRNO
7954 051646 62$: ERRDF    ERRNO,T194SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP    C4ERDF
                                .WORD   804
                                .WORD   T194SSR
                                .WORD   PKTSSR
                                051646 104455
                                051650 001444
                                051652 057741
                                051654 012046
7955 051656 005237 002222      INC    FATFLG      ;SET FATAL ERROR FLAG
7956 051662 70$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP    C4CLP1
7957      ; Do Write Subsystem Write Control to CLEAR loopback signals group 1.
7958      ; Do Write Subsystem Write Format to CLEAR loopback signals group 2.
7959      ; (the loopback signals have to be cleared here due to the flip-flops
7960      ; that are set on a 1 to 0 transition (IHER,IFMK,ICER))
7961 051664 005000      CLR    R0      ;WRITE 0'S
7962 051666 042700 000200      BIC    #WC.IFAD,R0      ;IFAD MUST ALWAYS =0
7963 051672 042700 000100      BIC    #WC.IOTAD,R0      ;ITADO MUST ALWAYS =0
7964 051676 042700 000040      BIC    #WC.IITAD,R0      ;ITAD1 MUST ALWAYS =0
7965 051702 004737 061472      JSR    PC,T19WCTL      ;SETUP PACKET FOR WRITE CONTROL
7966 051706 012704 062030      MOV    #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
7967 051712 010465 000000      MOV    R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
7968 051716 004737 016336      JSR    PC,CHKTSSR      ;WAIT FOR SSR TO SET
7969 051722      FORCERROR      82$      ;DO FORCE ERROR IF FORCER=1
7970 051736 103407      BCS    90$      ;BR IF CARRY SET (GOOD RETURN)
7971 051740 010001      MOV    R0,R1      ;SAVE CONTENTS OF TSSR
7972 051742      NEXT.ERRNO
7973 051742 82$: ERRDF    ERRNO,T197SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP    C4ERDF
                                .WORD   805
                                .WORD   T197SSR
                                .WORD   PKTSSR
                                051742 104455
                                051744 001445
                                051746 060113
                                051750 012046

```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

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

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

8015 052200 012701 057432      MOV      #T19EXSTA,R1      ;GET EXPECTED READ STATUS
8016 052204 012702 061722      MOV      #T198FSTA,R2      ;GET RECV READ STATUS
8017 052210 011211              MOV      (R2),(R1)      ;SET EXPD WORD #8 = RECV TEMP
8018 052212 042711 002000      BIC      #S1.ICER,(R1)      ;SET EXPD ICER =0
8019 052216 042711 001000      BIC      #S1.IFMK,(R1)      ;SET EXPD IFMK =0
8020 052222 042711 000400      BIC      #S1.IHER,(R1)      ;SET EXPD IHER =0
8021 052226 016261 000002 000002 MOV      2(R2),2(R1)      ;SET EXPD WORD #9 = RECV (NOT TESTING)
8022 052234 005000              CLR      R0      ;HIGH RECV ADDRESS FOR CKMSG2
8023 052236 012701 061702      MOV      #T198FR,R1      ;LOW RECV ADDRESS FOR CKMSG2
8024 052242 012702 057412      MOV      #T19EXP,R2      ;EXPD ADDRESS
8025 052246 012703 000024      MOV      #20,R3      ;NUMBER OF BYTES TO COMPARE
8026 052252 004737 011500      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
8027 052256              FORCERROR 152$,NOTSSR      ;880
8028 052266 103404              BCS      160$      ;BR IF YES
8029 052270              NEXT.ERRNO
8030 052270 103404 152$:      ERRHRD  ERRNO,T197CMP,MSGLOOP ;REPORT ERROR
                                TRAP      C#ERHRD
                                .WORD     809
                                .WORD     T197CMP
                                .WORD     MSGLOOP
8031 052300 104456 160$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
                                ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
8032 052302 005037 057344      CLR      T19PREV      ;INIT 1-0 TRANSITION FLAG
8033 052306 012703 002752      MOV      #TSTBLK,R3      ;GET FIRST PATTERN ADDRESS
8034 052312 012300 200$:      MOV      (R3)+,R0      ;GET A TEST PATTERN
8035 052314 010337 002316      MOV      R3,TSTPTR      ;SAVE POINTER INTO TSTBLK
8036 052320 042700 000200      BIC      #WC.IFAD,R0      ;IFAD MUST ALWAYS =0
8037 052324 042700 000100      BIC      #WC.ITAD,R0      ;ITAD0 MUST ALWAYS =0
8038 052330 042700 000040      BIC      #WC.ITAD,R0      ;ITAD1 MUST ALWAYS =0
8039 052334 010037 002312      MOV      R0,DATA      ;SET DATA PATTERN
8040 052334 010037 002312      ; Do Write Subsystem Write Control to Drive loopback signals group 1.
8041 052334 010037 002312      ;880 CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
8042 052340 013700 002312      MOV      DATA,R0      ;GET TEST PATTERN
8043 052344 004737 061614      JSR      PC,T19CNVT      ;CONVERT PATTERN TO CONTROL DRIVE MASK
8044 052344 004737 061614      ;R0 CONTAINS WRITE CONTROL DATA HERE
8045 052350 004737 061472      JSR      PC,T19WCTL      ;SETUP PACKET FOR WRITE CONTROL
8046 052354 012704 062030      MOV      #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
8047 052360 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
8048 052364 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
8049 052370 010407 010001      FORCERROR 212$      ;880FORCE ERROR IF FORCER=1
8050 052370 010407 010001      BCS      220$      ;BR IF CARRY SET (GOOD RETURN)
8051 052404 010001 212$:      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
8052 052406 010001 212$:      NEXT.ERRNO
8053 052410 010455 212$:      ERRDF  ERRNO,T197SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
8054 052410 010455 212$:      TRAP      L#ERDF
8055 052410 010455 212$:      .WORD     810
8056 052410 010455 212$:      .WORD     T197SSR
8057 052410 010455 212$:      .WORD     PKTSSR
8058 052420 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
8059 052424 104406 220$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
8060 052426 013700 002312      TRAP      C#CLP1
8061 052432 004737 061614      ; Do Write Subsystem Write Format to Drive loopback signals group 2.
8062 052432 004737 061614      ;880 CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
8063 052432 004737 061614      MOV      DATA,R0      ;GET TEST PATTERN
8064 052432 004737 061614      JSR      PC,T19CNVT      ;CONVERT PATTERN TO FORMAT DRIVE MASK

```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

8062	052436	000300		SWAB	R0		;WRITE FORMAT DATA RETURNED IN HIGH BYTE
8063	052440	004737	061512	JSR	PC,T19WFM		;SETUP PACKET FOR WRITE FORMAT
8064	052444	012704	062030	MOV	#T19PK2,R4		;GET WRITE SUBSYSTEM COMMAND PACKET
8065	052450	010465	000000	MOV	R4,TSD8(R5)		;SET THE PACKET ADDRESS TO EXECUTE
8066	052454	004737	016336	JSR	PC,CHKTSSR		;WAIT FOR SSR TO SET
8067	052460			FORCERROR	232#		;B0DFORCE ERROR IF FORCER=1
8068	052474	103407		BCS	240#		;BR IF CARRY SET (GOOD RETURN)
8069	052476	010001		MOV	R0,R1		;SAVE CONTENTS OF TSSR
8070	052500			NEXT.ERRNO			
8071	052500			232#:	ERRDF	ERRNO,T198SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	052500	104455					TRAP C#ERDF
	052502	001453					.WORD 811
	052504	060162					.WORD T198SSR
	052506	012046					.WORD PKTSSR
8072	052510	005237	002222	INC	FATFLG		;SET FATAL ERROR FLAG
8073	052514			240#:	CKLOOP		;LOOP ON ERROR, IF FLAG SET
	052514	104406					TRAP C#CLP1
8074				:	Do a Write Subsystem READ STATUS		
8075	052516	004737	061330	JSR	PC,T19SRD		;SETUP PACKET FOR READ STATUS
8076	052522	012704	062030	MOV	#T19PK2,R4		;GET WRITE SUBSYSTEM COMMAND PACKET
8077	052526	010465	000000	MOV	R4,TSD8(R5)		;SET THE PACKET ADDRESS TO EXECUTE
8078	052532	004737	016336	JSR	PC,CHKTSSR		;WAIT FOR SSR TO SET
8079	052536			FORCERROR	252#		;B0DFORCE ERROR IF FORCER=1
8080	052552	103407		BCS	260#		;BR IF CARRY SET (GOOD RETURN)
8081	052554	010001		MOV	R0,R1		;SAVE CONTENTS OF TSSR
8082	052556			NEXT.ERRNO			
8083	052556			252#:	ERRDF	ERRNO,T193SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	052556	104455					TRAP C#ERDF
	052560	001454					.WORD 812
	052562	057674					.WORD T193SSR
	052564	012046					.WORD PKTSSR
8084	052566	005237	002222	INC	FATFLG		;SET FATAL ERROR FLAG
8085	052572			260#:	CKLOOP		;LOOP ON ERROR, IF FLAG SET
	052572	104406					TRAP C#CLP1
8086				:	If loopback data NOT= data sent	Then Print Error	
8087	052574	004737	061570	JSR	PC,T19SETEXP		;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
8088	052600	012701	057432	MOV	#T19EXSTA,R1		;GET EXPECTED READ STATUS
8089	052604	012702	061722	MOV	#T198FSTA,R2		;GET RECV READ STATUS
8090	052610	013711	002312	MOV	DATA,(R1)		;SET EXPD WORD #8 TO TEST DATA FIRST
8091	052614	013700	057344	MOV	T19PREV,R0		;GET PREVIOUS DATA PATTERN
8092	052620	013703	002312	MOV	DATA,R3		;GET CURRENT PATTERN
8093	052624	012704	000400	MOV	#S1.IHER,R4		;SETUP IHER EXPECTED
8094	052630	040411		BIC	R4,(R1)		;SET EXPD IHER =0
8095	052632	030400		BIT	R4,R0		;PREVIOUS =1?
8096	052634	001403		BEQ	275#		;BR IF NO
8097	052636	030403		BIT	R4,R3		;CURRENT =0?
8098	052640	001001		BNE	275#		;BR IF NO
8099	052642	050411		BIS	R4,(R1)		;SET EXPD IHER =1
8100	052644	012704	001000	275#:	MOV	#S1.IFMK,R4	;SETUP IFMK EXPECTED
8101	052650	040411		BIC	R4,(R1)		;SET EXPD IFMK =0
8102	052652	030400		BIT	R4,R0		;PREVIOUS =1?
8103	052654	001403		BEQ	280#		;BR IF NO
8104	052656	030403		BIT	R4,R3		;CURRENT =0?
8105	052660	001001		BNE	280#		;BR IF NO
8106	052662	050411		BIS	R4,(R1)		;SET EXPD IFMK =1
8107	052664	012704	002000	280#:	MOV	#S1.ICER,R4	;SETUP ICER EXPECTED
8108	052670	040411		BIC	R4,(R1)		;SET EXPD ICER =0

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

8109 052672 030400      BIT      R4,R0      ;PREVIOUS =1?
8110 052674 001403      BEQ      285$      ;BR IF NO
8111 052676 030403      BIT      R4,R3      ;CURRENT =0?
8112 052700 001001      BNE      285$      ;BR IF NO
8113 052702 050411      BIS      R4,(R1)    ;SET EXPD ICER =1
8114 052704 011100      MOV      (R1),R0    ;GET EXPD WORD
8115                      ; If previous IIDENT=1 and current is IIDENT=1 then EXPD= 0 else 1
8116 052706 012704 004000      MOV      $S1.IIDENT,R4 ;IIDENT
8117 052712 050400      BIS      R4,R0      ;ASSUME EXPD=1
8118 052714 030437 057344      BIT      R4,T19PREV ;PREVIOUS IIDENT=1?
8119 052720 001403      BEQ      288$      ;BR IF NO
8120 052722 030403      BIT      R4,R3      ;IS CURRENT IIDENT=1?
8121 052724 001401      BEQ      288$      ;BR IF NO
8122 052726 040400      BIC      R4,R0      ;SET EXPD=0
8123 052730 052700 040000      BIS      $S1.I2RES,R0 ;IRESV2 EXPD ALWAYS=1
8124 052734 052700 020000      BIS      $S1.I1RES,R0 ;IRESV1 EXPD ALWAYS=1
8125 052740 042700 100000      BIC      $S1.PARERR,R0 ;IGNORE PARERR
8126 052744 032712 100000      BIT      $S1.PARERR,(R2) ;IS PARERR SET IN RECV?
8127 052750 C01402      BEQ      290$      ;BR IF NO
8128 052752 052700 100000      BIS      $S1.PARERR,R0 ;SET IN EXPD
8129 052756 010011      MOV      R0,(R1)    ;SETUP FINAL EXPD IN WORD #8
8130 052760 016261 000002 000002      MOV      2(R2),2(R1) ;SET EXPD WORD #9 = RECV (NOT TESTING)
8131 052766 005000      CLR      R0        ;HIGH RECV ADDRESS FOR CKMSG2
8132 052770 012701 061702      MOV      $T198FR,R1 ;LOW RECV ADDRESS FOR CKMSG2
8133 052774 012702 057412      MOV      $T19EXP,R2 ;EXPD ADDRESS
8134 053000 012703 000024      MOV      $20,R3   ;NUMBER OF BYTES TO COMPARE
8135 053004 004737 011500      JSR      PC,CKMSG2 ;EXPD EQUAL RECV?
8136 053010                      FORCERROR 302$,NOTSSR ;@RD
8137 053020 103404      BCS      310$      ;BR IF YES
8138 053022                      NEXT.ERRNO
8139 053022      302$:  ERRHRD  ERRNO,T198CMP,MSGLOOP ;REPORT ERROR
                        TRAP      C$ERHRD
                        .WORD     813
                        .WORD     T198CMP
                        .WORD     MSGLOOP
8140 053032      310$:  CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
                        ; Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
8141 053034 004737 061350      JSR      PC,T19R5FIF ;SETUP PKT FOR WRITE MISC Reset STATUS
8142 053040 012704 062030      MOV      $T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8143 053044 010465 000000      MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8144 053050 004737 016336      JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
8145 053054                      FORCERROR 322$ ;@RD FORCE ERROR IF FORCER=1
8146 053070 103407      BCS      330$      ;BR IF CARRY SET (GOOD RETURN)
8147 053072 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
8148 053074                      NEXT.ERRNO
8149 053074      322$:  ERDF  ERRNO,T192SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                        TRAP      C$ERDF
                        .WORD     814
                        .WORD     T192SSR
                        .WORD     PKTSSR
8150 053074 104455
8151 053076 001456
8152 053100 057630
8153 053102 012046
8154 053104 005237 002222      INC      FATFLG ;SET FATAL ERROR FLAG
8155 053110 104406      330$:  CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
                        ; Do a Write Subsystem READ STATUS
8156 053112 004737 051330      JSR      PC,T19SRD ;SETUP PACKET FOR READ STATUS
8157 053116 012704 062030      MOV      $T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET

```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

8156	053122	010465	000000		MOV R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE
8157	053126	004737	016336		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET
8158	053132				FORCERROR 342#	;BR IF CARRY SET (GOOD RETURN)
8159	053146	103407			BCS 350#	;SAVE CONTENTS OF TSSR
8160	053150	010001			MOV R0,R1	
8161	053152				NEXT.ERRNO	
8162	053152			342#:	ERRDF ERRNO,T193SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	053152	104455				TRAP C#ERDF
	053154	001457				.WORD 815
	053156	057674				.WORD T193SSR
	053160	012046				.WORD PKTSSR
8163	053162	005237	002222		INC FATFLG	;SET FATAL ERROR FLAG
8164	053166			350#:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
	053166	104406				TRAP C#CLP1
8165	053170	004737	061570		JSR PC,T19SETEXP	;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
8166	053174	012701	057432		MOV #T19EXSTA,R1	;GET EXPECTED READ STATUS
8167	053200	012702	061722		MOV #T19BFSTA,R2	;GET RCV READ STATUS
8168	053204	011211			MOV (R2),(R1)	;SET EXPD WORD #8 = RCV TEMP
8169	053206	042711	002000		BIC #S1.ICER,(R1)	;SET EXPD ICER =0
8170	053212	042711	001000		BIC #S1.IFMR,(R1)	;SET EXPD IFMR =0
8171	053216	042711	000400		BIC #S1.IHER,(R1)	;SET EXPD IHER =0
8172	053222	016261	000002	000002	MOV 2(R2),2(R1)	;SET EXPD WORD #9 = RCV (NOT TESTING)
8173	053230	005000			CLR R0	;HIGH RCV ADDRESS FOR CKMSG2
8174	053232	012701	061702		MOV #T19BFR,R1	;LOW RCV ADDRESS FOR CKMSG2
8175	053236	012702	057412		MOV #T19EXP,R2	;EXPD ADDRESS
8176	053242	012703	000024		MOV #20,R3	;NUMBER OF BYTES TO COMPARE
8177	053246	004737	011500		JSR PC,CKMSG2	;EXPD EQUAL RCV?
8178	053252				FORCERROR 362#,NOTSSR	;BR IF YES
8179	053262	103404			BCS 370#	
8180	053264				NEXT.ERRNO	
8181	053264			362#:	ERRHRD ERRNO,T197CMP,MSGSTAT	;REPORT ERROR
	053264	104456				TRAP C#ERHRD
	053266	001460				.WORD 816
	053270	060633				.WORD T197CMP
	053272	012350				.WORD MSGSTAT
8182	053274			370#:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
	053274	104406				TRAP C#CLP1
8183						
8184	053276	013737	002312	057344	MOV DATA,T19PREV	;SETUP PREVIOUS DATA FOR EXPD CALC.
8185	053304	013703	002316		MOV TSTPTR,R3	;RESTORE CURRENT TSTBLK POINTER
8186	053310	020327	003062		CMP R3,#TBLEND	;END OF TSTBLK?
8187	053314	103002			BHIS 400#	;BR IF YES
8188	053316	000137	052312		JMP 200#	;DO NEXT TSTBLK PATTERN
8189	053322			400#:		
8190						
8191	053322				ENDSUB	;////////// END SUBTEST //////////
	053322					L10067:
	053322	104403				TRAP C#ESUB
8192						
8193	053324	005737	002222		TST FATFLG	;ANY FATAL ERRORS ?
8194	053330	001402			BEQ 460#	;BRANCH IF NOT
8195	053332	004737	017202		JSR PC,CKDROP	;TRY TO DROP THE UNIT
8196	053336			460#:		
8197						
8198						
8199						
8200						

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

8201
8202          .SBTTL TEST 8: SUBTEST 2: LOOPBACK READ/WRITE SIGNALS TEST
8203
8204
8205      ;**
8206      ; TEST 8: SUBTEST 2:
8207      ; SUBTEST DESCRIPTION:
8208      ;
8209      ;     This subtest verifies the Read/Write data loopback path.
8210      ;     The Read/Write data signals are IR<7:0> and IW<7:0>
8211      ;     respectively.
8212      ;
8213      ; TEST STEPS:
8214      ;
8215      ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
8216      ; BEGIN
8217      ;     Write to TSSR register to soft initialize the controller
8218      ;     Do WRITE CHARACTERISTICS to check for Extended Features Switch
8219      ;     If Extended Features Hardware Switch Clear then:
8220      ;         Do Write Subsystem Write Miscellaneous to Set Extended Features.
8221      ;     Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
8222      ;     Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
8223      ;     Do a WRITE NPR to set loopback and tape direction OUT
8224      ;     Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
8225      ;     Do a READ FIFO with tape direction OUT to load tape out write latch
8226      ;     Do a WRITE NPR to set loopback and tape direction IN
8227      ;     Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
8228      ;     to strobe loopback data into FIFO.
8229      ;     Do a READ FIFO with tape direction IN to read data
8230      ;     If Data read from FIFO NOT= to Data sent Then Print Error
8231      ;     Do a Write Subsystem READ STATUS
8232      ;     If Input Ready NOT=1 Then Print Error
8233      ;     If Output Ready NOT=0 Then Print Error
8234      ;     If Data In Miss NOT=0 Then Print Error
8235      ; END
8236      ;--
8237
8238      BGNSUB          ;////////// BEGIN SUBTEST //////////
8239                      T8.2:
8240                      TRAP C#BSUB
8241
8242      ; Write to TSSR register to soft initialize the controller
8243      5$:
8244      JSR PC,SOFINIT          ;WRITE TO TSSR TO SOFT INITIALIZE
8245      BCS 10$                ;BR IF SOFT INIT OKAY
8246      MOV R0,R1              ;SAVE CONTENTS OF TSSR
8247      ERRDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
8248
8249                      TRAP C#ERDF
8250                      .WORD 816
8251                      .WORD SFIERR
8252                      .WORD SFIMSG
8253
8254      ; Do WRITE CHARACTERISTICS to check for Extended Features Switch
8255      10$:
8256      CLR FATFLG              ;CLEAR FATAL ERROR FLAG
8257      MOV #T19PACKET,R4       ;GET THE ADDRESS OF COMMAND PACKET
8258      JSR PC,WRTCHR           ;DO WRITE CHARACTERISTICS COMMAND
8259      FORCERROR 12$           ;DO FORCE ERROR IF FORCER=1
8260      BCS 15$                 ;BR IF CARRY SET (GOOD RETURN)
8261      MOV R0,R1              ;SAVE CONTENTS OF TSSR

```

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

```

8252 053414      NEXT.ERRNO
8253 053414      124:  ERRDF  ERRNO,T19SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP  C#ERDF
                                .WORD 817
                                .WORD T19SSR
                                .WORD PKTSSR
      053414 104455
      053416 001461
      053420 057573
      053422 012046
8254 053424 005237 002222      INC  FATFLG      ;SET FATAL ERROR FLAG
8255 053430      154:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      053430 104406                                TRAP  C#CLP1
8256      ;      If Extended Features Hardware Switch Clear then:
8257      ;      Do Write Subsystem Write Miscellaneous to Set Extended Features.
8258 053432 012701 061702      MOV  #T19BFR,R1      ;MESSAGE BUFFER ADDRESS
8259 053436 032761 000200 000012  BIT  #X2.EXTF,XST2(R1)      ;EXTENDED FEATURES SWITCH SET?
8260 053444 001026      BNE  30$      ;BR IF YES
8261 053446 004737 061532      JSR  PC,T19SEXT      ;SETUP PACKET FOR WRITE MISC INVERT
8262 053452 012704 062030      MOV  #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
8263 053456 010465 000000      MOV  R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
8264 053462 004737 016336      JSR  PC,CHKTSSR      ;WAIT FOR SSR TO SET
8265 053466      FORCERROR 22$      ;GOODFORCE ERROR IF FORCER=1
8266 053502 103407      BCS  30$      ;BR IF CARRY SET (GOOD RETURN)
8267 053504 010001      MOV  R0,R1      ;SAVE CONTENTS OF TSSR
8268 053506      NEXT.ERRNO
8269 053506      224:  ERRDF  ERRNO,T192SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP  C#ERDF
                                .WORD 818
                                .WORD T192SSR
                                .WORD PKTSSR
      053506 104455
      053510 001462
      053512 057630
      053514 012046
8270 053516 005237 002222      INC  FATFLG      ;SET FATAL ERROR FLAG
8271 053522      304:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      053522 104406                                TRAP  C#CLP1
8272      ;      Do WRITE CHARACTERISTICS to select reserved unit 7
8273 053524 012704 061660      MOV  #T19PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
8274 053530 004737 010662      JSR  PC,WRTCHR      ;DO WRITE CHARACTERISTICS COMMAND
8275 053534      FORCERROR 42$      ;GOODFORCE ERROR IF FORCER=1
8276 053550 103407      BCS  50$      ;BR IF CARRY SET (GOOD RETURN)
8277 053552 010001      MOV  R0,R1      ;SAVE CONTENTS OF TSSR
8278 053554      NEXT.ERRNO
8279 053554      424:  ERRDF  ERRNO,T19SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP  C#ERDF
                                .WORD 819
                                .WORD T19SSR
                                .WORD PKTSSR
      053554 104455
      053556 001463
      053560 057573
      053562 012046
8280 053564 005237 002222      INC  FATFLG      ;SET FATAL ERROR FLAG
8281 053570      504:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      053570 104406                                TRAP  C#CLP1
8282
8283
8284      ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
8285 053572 012703 002752      MOV  #TSTBLK,R3      ;GET FIRST PATTERN ADDRESS
8286 053576 012337 002312 1004:  MOV  (R3)+,DATA      ;GET A TEST PATTERN
8287 053602 042737 177400 002312  BIC  #C<377>,DATA      ;DATA IS BYTE
8288 053610 010337 002316      MOV  R3,TSTPTR      ;SETUP CURRENT TSTBLK POINTER
8289      ;      Do a WRITE NPR to set loopback and tape direction OUT
8290 053614 012700 000100      MOV  #NP.OUT,R0      ;SET TAPE DIRECTION OUT
8291 053620 052700 000040      BIS  #NP.LOOP,R0      ;SET LOOPBACK
8292 053624 004737 061372      JSR  PC,T19SNPR      ;SETUP T19PK2 FOR WRITE NPR
8293 053630 012704 062030      MOV  #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET

```

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

8294	053634	010465	000000	MOV	R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE
8295	053640	004737	016336	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
8296	053644			FORCERROR	102#	;BDFORCE ERROR IF FORCER=1
8297	053660	103407		BCS	105#	;BR IF CARRY SET (GOOD RETURN)
8298	053662	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR
8299	053664			NEXT.ERRNO		
8300	053664			102#:	ERRDF ERRNO,T194SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	053664	104455				TRAP C#ERDF
	053666	001464				.WORD 820
	053670	057741				.WORD T194SSR
	053672	012046				.WORD PKTSSR
8301	053674	005237	002222	INC	FATFLG	;SET FATAL ERROR FLAG
8302	053700			105#:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
	053700	104406				TRAP C#CLP1
8303				:	Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT	
8304	053702	012700	000001	MOV	#1,R0	;WRITE 1 BYTE
8305	053706	012701	002312	MOV	#DATA,R1	;FIFO WRITE DATA ADDRESS
8306	053712	004737	061436	JSR	PC,T19WFIF	;SETUP T19PK2 FOR WRITE FIFO
8307	053716	C12704	062030	MOV	#T19PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
8308	053722	010465	000000	MOV	R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE
8309	053726	004737	016336	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
8310	053732			FORCERROR	107#	;BDFORCE ERROR IF FORCER=1
8311	053746	103407		BCS	110#	;BR IF CARRY SET (GOOD RETURN)
8312	053750	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR
8313	053752			NEXT.ERRNO		
8314	053752			107#:	ERRDF ERRNO,T195SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	053752	104455				TRAP C#ERDF
	053754	001465				.WORD 821
	053756	060004				.WORD T195SSR
	053760	012046				.WORD PKTSSR
8315	053762	005237	002222	INC	FATFLG	;SET FATAL ERROR FLAG
8316	053766			110#:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
	053766	104406				TRAP C#CLP1
8317				:	Do a READ FIFO with tape direction OUT to load tape out write latch	
8318	053770	012700	000001	MOV	#1,R0	;SET READ BYTE COUNT
8319	053774	004737	061416	JSR	PC,T19RFIF	;SETUP T19PK2 FOR READ FIFO
8320	054000	012704	062030	MOV	#T19PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
8321	054004	010465	000000	MOV	R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE
8322	054010	004737	016336	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
8323	054014			FORCERROR	122#	;BDFORCE ERROR IF FORCER=1
8324	054030	103407		BCS	130#	;BR IF CARRY SET (GOOD RETURN)
8325	054032	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR
8326	054034			NEXT.ERRNO		
8327	054034			122#:	ERRDF ERRNO,T196SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	054034	104455				TRAP C#ERDF
	054036	001466				.WORD 822
	054040	060050				.WORD T196SSR
	054042	012046				.WORD PKTSSR
8328	054044	005237	002222	INC	FATFLG	;SET FATAL ERROR FLAG
8329	054050			130#:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
	054050	104406				TRAP C#CLP1
8330				:	Do a WRITE NPR to set loopback and tape direction IN	
8331	054052	005000		CLR	R0	;CLR NP.OUT TO SET TAPE DIRECTION IN
8332	054054	052700	000040	BIS	#NP.LOOP,R0	;SET LOOPBACK
8333	054060	004737	061372	JSR	PC,T19SNPR	;SETUP T19PK2 FOR WRITE NPR
8334	054064	012704	062030	MOV	#T19PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
8335	054070	010465	000000	MOV	R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE

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

8336	054074	004737	016336	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
8337	054100			FORCERROR	1424	;BDOFORCE ERROR IF FORCER=1
8338	054114	103407		BCS	1504	;BR IF CARRY SET (GOOD RETURN)
8339	054116	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR
8340	054120			NEXT.ERRNO		
8341	054120			1424:	ERRDF ERRNO,T194SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	054120	104455				TRAP C4ERDF
	054122	001467				.WORD 823
	054124	057741				.WORD T194SSR
	054126	012046				.WORD PKTSSR
8342	054130	005237	002222	INC	FATFLG	;SET FATAL ERROR FLAG
8343	054134			1504:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
	054134	104406				TRAP C4CLP1
8344				:	Do a WRITE FIFO with byte count equal to 1 and Tape direction IN	
8345	054136	012700	000001	MOV	#1,R0	;WRITE 1 BYTE
8346	054142	012701	002312	MOV	#DATA,R1	;FIFO WRITE DATA ADDRESS
8347	054146	004737	061436	JSR	PC,T19WFIF	;SETUP T19PK2 FOR WRITE FIFO
8348	054152	012704	062030	MOV	#T19PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
8349	054156	C10465	000000	MOV	R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE
8350	054162	004737	016336	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
8351	054166			FORCERROR	1624	;BDOFORCE ERROR IF FORCER=1
8352	054202	103407		BCS	1704	;BR IF CARRY SET (GOOD RETURN)
8353	054204	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR
8354	054206			NEXT.ERRNO		
8355	054206			1624:	ERRDF ERRNO,T195SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	054206	104455				TRAP C4ERDF
	054210	001470				.WORD 824
	054212	060004				.WORD T195SSR
	054214	012046				.WORD PKTSSR
8356	054216	005237	002222	INC	FATFLG	;SET FATAL ERROR FLAG
8357	054222			1704:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
	054222	104406				TRAP C4CLP1
8358				:	Do a READ FIFO with tape direction IN to read data	
8359				:	If Data read from FIFO NOT= to Data sent Then Print Error	
8360	054224	012700	000001	MOV	#1,R0	;SET READ BYTE COUNT
8361	054230	004737	061416	JSR	PC,T19RFIF	;SETUP T19PK2 FOR READ FIFO
8362	054234	012704	062030	MOV	#T19PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
8363	054240	010465	000000	MOV	R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE
8364	054244	004737	016336	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
8365	054250			FORCERROR	1824	;BDOFORCE ERROR IF FORCER=1
8366	054264	103407		BCS	1904	;BR IF CARRY SET (GOOD RETURN)
8367	054266	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR
8368	054270			NEXT.ERRNO		
8369	054270			1824:	ERRDF ERRNO,T196SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	054270	104455				TRAP C4ERDF
	054272	001471				.WORD 825
	054274	060050				.WORD T196SSR
	054276	012046				.WORD PKTSSR
8370	054300	005237	002222	INC	FATFLG	;SET FATAL ERROR FLAG
8371	054304			1904:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
	054304	104406				TRAP C4CLP1
8372	054306	004737	061570	JSR	PC,T19SETEXP	;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
8373	054312	012701	057432	MOV	#T19EXSTA,R1	;GET EXPECTED READ STATUS
8374	054316	012702	061722	MOV	#T19BFSTA,R2	;GET RCV READ STATUS
8375	054322	013711	002312	MOV	DATA,(R1)	;SET EXPD WORD #8 = DATA
8376	054326	016261	000002 000002	MOV	2(R2),2(R1)	;SET EXPD WORD #9 = RCV (NOT TESTING)
8377	054334	005000		CLR	R0	;HIGH RCV ADDRESS FOR CKMSG2

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

```

8378 054336 012701 061702      MOV      #T198FR,R1      ;LOW RECV ADDRESS FOR CKMSG2
8379 054342 012702 057412      MOV      #T19EXP,R2      ;EXPD ADDRESS
8380 054346 012703 000022      MOV      #18.,R3      ;NUMBER OF BYTES TO COMPARE
8381 054352 004737 011500      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
8382 054356      FORCERROR      202$,NOTSSR      ;880
8383 054366 103404      BCS      210$      ;BR IF YES
8384 054370      NEXT.ERRNO
8385 054370 202$: ERRHRD ERRNO,T199CMP,MSGSUB ;REPORT ERROR
      054370 104456      TRAP      C$ERHRD
      054372 001472      .WORD      826
      054374 061010      .WORD      T199CMP
      054376 013742      .WORD      MSGSUB
8386 054400 210$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      054400 104406      TRAP      C$CLP1
8387      ; Do a Write Subsystem READ STATUS
8388      ; If Input Ready NOT=1 Then Print Error
8389      ; If Output Ready NOT=0 Then Print Error
8390      ; If Data In Miss NOT=0 Then Print Error
8391 054402 C04737 061330      JSR      PC,T19SRD      ;SETUP PACKET FOR READ STATUS
8392 054406 012704 062030      MOV      #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
8393 054412 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
8394 054416 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
8395 054422      FORCERROR      212$      ;880FORCE ERROR IF FORCER=1
8396 054436 103407      BCS      220$      ;BR IF CARRY SET (GOOD RETURN)
8397 054440 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
8398 054442      NEXT.ERRNO
8399 054442 212$: ERRDF ERRNO,T193SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      054442 104455      TRAP      C$ERDF
      054444 001473      .WORD      827
      054446 057674      .WORD      T193SSR
      054450 012046      .WORD      PKTSSR
8400 054452 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
8401 054456 220$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      054456 104406      TRAP      C$CLP1
8402 054460 004737 061570      JSR      PC,T19SETEXP      ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
8403 054464 012701 057432      MOV      #T19EXSTA,R1      ;GET EXPECTED READ STATUS
8404 054470 012702 061722      MOV      #T198FSTA,R2      ;GET RECV READ STATUS
8405 054474 012221      MOV      (R2)+,(R1)+      ;SET EXPD WORD #8 = RECV TEMP
8406 054476 011211      MOV      (R2),(R1)      ;SET EXPD WORD #9 = RECV TEMP
8407 054500 052711 000020      BIS      #S2.INRDY,(R1)      ;SET EXP INPUT READY= 1
8408 054504 042711 000040      BIC      #S2.OUTRDY,(R1)      ;SET EXP OUTPUT READY= 0
8409 054510 042711 000200      BIC      #S2.DIM,(R1)      ;SET EXP DATA IN MISS = 0
8410 054514 005000      CLR      R0      ;HIGH RECV ADDRESS FOR CKMSG2
8411 054516 012701 061702      MOV      #T198FR,R1      ;LOW RECV ADDRESS FOR CKMSG2
8412 054522 012702 057412      MOV      #T19EXP,R2      ;EXPD ADDRESS
8413 054526 012703 000024      MOV      #20.,R3      ;NUMBER OF BYTES TO COMPARE
8414 054532 004737 011500      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
8415 054536      FORCERROR      232$,NOTSSR      ;880
8416 054546 103404      BCS      240$      ;BR IF YES
8417 054550      NEXT.ERRNO
8418 054550 232$: ERRHRD ERRNO,T196CMP,MSGSTAT ;REPORT ERROR
      054550 104456      TRAP      C$ERHRD
      054552 001474      .WORD      828
      054554 060550      .WORD      T196CMP
      054556 012350      .WORD      MSGSTAT
8419 054560 240$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      054560 104406      TRAP      C$CLP1

```

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

```

8420
8421
8422
8423 054562          FORCEEXIT          255#          ;880
8424 054572 013703 002316      MOV      TSTPTR,R3      ;RESTORE CURRENT TSTBLK POINTER
8425 054576 020327 003062      CMP      R3,#TBLEND      ;END OF TSTBLK?
8426 054602 103002          BHIS      255#          ;BR IF YES
8427 054604 000137 053576      JMP      100#          ;DO ANOTHER TSTBLK PATTERN
8428 054610          255#
8429
8430 054610          ENDSUB          ;////////// END SUBTEST //////////
      054610          L10070:          TRAP      C#ESUB
      054610 104403
8431
8432 054612 005737 002222      TST      FATFLG          ;ANY FATAL ERRORS ?
8433 054616 001402          BEQ      260#          ;BRANCH IF NOT
8434 054620 004737 017202      JSR      PC,CKDROP        ;TRY TO DROP THE UNIT
8435 054624          260#
8436
8437
8438          .SBTTL TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST
8439
8440
8441 ;**
8442 ; TEST 8: SUBTEST 3:
8443 ; SUBTEST DESCRIPTION:
8444 ;
8445 ;       This subtest verifies the Write Strobe loopback path
8446 ;       can strobe data from the FIFO to the Data lines.
8447 ;       The signal IRESV3 drives IWSTR (write strobe) to write
8448 ;       data from the FIFO to the tape data out latch.
8449 ;
8450 ; TEST STEPS:
8451 ;
8452 ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
8453 ; BEGIN
8454 ;       Write to TSSR register to soft initialize the controller
8455 ;       Do WRITE CHARACTERISTICS to check for Extended Features Switch
8456 ;       If Extended Features Hardware Switch Clear then:
8457 ;       Do Write Subsystem Write Miscellaneous to Set Extended Features.
8458 ;       Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
8459 ;       Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
8460 ;       Do a WRITE NPR to set loopback and tape direction OUT
8461 ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
8462 ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
8463 ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 0 to load write data latch
8464 ;       Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
8465 ;       Do a WRITE NPR to set loopback and tape direction IN
8466 ;       Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
8467 ;       to strobe loopback data into FIFO.
8468 ;       Do a READ FIFO with tape direction IN to read data
8469 ;       If Data read from FIFO NOT= to Data sent Then Print Error
8470 ;
8471 ; END
8472 ;--
8473 054624          BGNSUB          ;////////// BEGIN SUBTEST //////////
      054624          T8.3:

```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```

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

```

Address	Offset	Label	Operation	Comments	Register/Flag	Value	Device	SSR	PKT
055044	001477							831	
055046	057573							T19SSR	
055050	012046							PKTSSR	
8515	055052	005237	002222	INC	FATFLG				
8516	055056			50\$:	CKLOOP				
055056	104406							SET	C#CLP1
8517								TRAP	
8518									
8519	055060	012703	002752						
8520	055064	012337	002312	100\$:	MOV	(R3)+,DATA			
8521	055070	042737	177400		BIC	#C<377>,DATA			
8522	055076	010337	002316		MOV	R3,TSTPTR			
8523									
8524	055102	012700	000100		Do a WRITE NPR to set loopback and tape direction OUT				
8525	055106	052700	000040		MOV	#NP.OUT,R0			
8526	055112	004737	061372		BIS	#NP.LOOP,R0			
8527	055116	012704	062030		JSR	PC,T19SNPR			
8528	055122	010465	000000		MOV	#T19PK2,R4			
8529	055126	004737	016336		MOV	R4,TSDB(R5)			
8530	055132				JSR	PC,CHKTSSR			
8531	055146	103407			FORCERROR	102\$			
8532	055150	010001			BCS	105\$			
8533	055152				MOV	R0,R1			
8534	055152			102\$:	ERRDF	ERRNO,T194SSR,PKTSSR			
055152	104455								
055154	001500								
055156	057741								
055160	012046								
8535	055162	005237	002222	105\$:	INC	FATFLG			
8536	055166				CKLOOP				
055166	104406							SET	C#CLP1
8537								TRAP	
8538	055170	012700	000002		Do a WRITE FORMAT to set IRESV3==>IWSTR = 1				
8539	055174	004737	061512		MOV	#WF.I3RES,P0			
8540	055200	012704	062030		JSR	PC,T19WFMT			
8541	055204	010465	0000C0		MOV	#T19PK2,R4			
8542	055210	004737	016336		MOV	R4,TSDB(R5)			
8543	055214				JSR	PC,CHKTSSR			
8544	055230	103407			FORCERROR	112\$			
8545	055232	010001			BCS	120\$			
8546	055234				MOV	R0,R1			
8547	055234			112\$:	ERRDF	ERRNO,T198SSR,PKTSSR			
055234	104455								
055236	001501								
055240	060162								
055242	012046								
8548	055244	005237	002222	120\$:	INC	FATFLG			
8549	055250				CKLOOP				
055250	104406							SET	C#CLP1
8550								TRAP	
8551	055252	012700	000001		Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT				
8552	055256	012701	002312		MOV	#1,R0			
8553	055262	004737	061436		MOV	#DATA,R1			
8554	055266	012704	062030		JSR	PC,T19WFIF			
8555	055272	010465	000000		MOV	#T19PK2,R4			
8556	055276	004737	016336		MOV	R4,TSDB(R5)			

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```

8558 055316 103407      BCS      140$      ;BR IF CARRY SET (GOOD RETURN)
8559 055320 010001      MOV      RO,R1      ;SAVE CONTENTS OF TSSR
8560 055322      NEXT.ERRNO
8561 055322 132$:      ERDFF      ERRNO,T195SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      834
                                .WORD      T195SSR
                                .WORD      PKTSSR
                                055322 104455
                                055324 001502
                                055326 060004
                                055330 012046
8562 055332 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
8563 055336 140$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                055336 104406
;      Do a WRITE FORMAT to set IRESV3==>IWSTR = 0
      CLR      RO      ;SET IRESV3==>IWSTR=0
      JSR      PC,T19WFMT ;SETUP T9PK2 FOR WRITE FORMAT
      MOV      #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
      MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
      JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
      FORCERROR 152$      ;@@DFORCE ERROR IF FORCER=1
8571 055376 103407      BCS      160$      ;BR IF CARRY SET (GOOD RETURN)
8572 055400 010001      MOV      RO,R1      ;SAVE CONTENTS OF TSSR
8573 055402      NEXT.ERRNO
8574 055402 152$:      ERDFF      ERRNO,T198SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      835
                                .WORD      T198SSR
                                .WORD      PKTSSR
                                055402 104455
                                055404 001503
                                055406 060162
                                055410 012046
8575 055412 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
8576 055416 160$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                055416 104406
;      Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
      MOV      #WF.I3RES,RO ;IRESV3==>IWSTR=1
      JSR      PC,T19WFMT ;SETUP T9PK2 FOR WRITE FORMAT
      MOV      #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
      MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
      JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
      FORCERROR 172$      ;@@DFORCE ERROR IF FORCER=1
8584 055460 103407      BCS      180$      ;BR IF CARRY SET (GOOD RETURN)
8585 055462 010001      MOV      RO,R1      ;SAVE CONTENTS OF TSSR
8586 055464      NEXT.ERRNO
8587 055464 172$:      ERDFF      ERRNO,T198SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      836
                                .WORD      T198SSR
                                .WORD      PKTSSR
                                055464 104455
                                055466 001504
                                055470 060162
                                055472 012046
8588 055474 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
8589 055500 180$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                055500 104406
8590
8591 ;      Do a WRITE NPR to set loopback and tape direction IN
      CLR      RO      ;CLR NP.OUT TO SET TAPE DIRECTION IN
      BIS      #NP.LOOP,RO ;SET LOOPBACK
      JSR      PC,T19SNPR ;SETUP T19PK2 FOR WRITE NPR
      MOV      #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
      MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
      JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
      FORCERROR 182$      ;@@DFORCE ERROR IF FORCER=1
8598 055530      BCS      190$      ;BR IF CARRY SET (GOOD RETURN)
8599 055544 103407

```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```

8600 055546 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
8601 055550      NEXT.ERRNO
8602 055550      182$:  ERRDF  ERRNO,T194SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                        TRAP      C$ERDF
                        .WORD      837
                        .WORD      T194SSR
                        .WORD      PKTSSR
                        055550 104455
                        055552 001505
                        055554 057741
                        055556 012046
8603 055560 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
8604 055564      190$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
                        ; Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
8605 055566 012700 000001      MOV      #1,R0      ;WRITE 1 BYTE
8606 055572 012701 002312      MOV      #DATA,R1      ;FIFO WRITE DATA ADDRESS
8607 055576 004737 061436      JSR      PC,T19WFIF      ;SETUP T19PK2 FOR WRITE FIFO
8608 055602 012704 062030      MOV      #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
8609 055606 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
8610 055612 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
8611 055616      FORCERROR 202$      ;000FORCE ERROR IF FORCER=1
8612 055632 103407      BCS      210$      ;BR IF CARRY SET (GOOD RETURN)
8613 055634 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
8614 055636      NEXT.ERRNO
8615 055636      202$:  ERRDF  ERRNO,T195SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                        TRAP      C$ERDF
                        .WORD      838
                        .WORD      T195SSR
                        .WORD      PKTSSR
                        055636 104455
                        055640 001506
                        055642 060004
                        055644 012046
8617 055646 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
8618 055652      210$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
                        ; Do a READ FIFO with tape direction IN to read data
8619 055654 012700 000001      MOV      #1,R0      ;SET READ BYTE COUNT
8620 055660 004737 061416      JSR      PC,T19RFIF      ;SETUP T19PK2 FOR READ FIFO
8621 055664 012704 062030      MOV      #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
8622 055670 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
8623 055674 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
8624 055700      FORCERROR 222$      ;000FORCE ERROR IF FORCER=1
8625 055714 103407      BCS      230$      ;BR IF CARRY SET (GOOD RETURN)
8626 055716 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
8627 055720      NEXT.ERRNO
8628 055720      222$:  ERRDF  ERRNO,T196SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                        TRAP      C$ERDF
                        .WORD      839
                        .WORD      T196SSR
                        .WORD      PKTSSR
                        055720 104455
                        055722 001507
                        055724 060050
                        055726 012046
8630 055730 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
8631 055734      230$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
                        ; If Data read from FIFO NOT= to Data sent Then Print Error
8632 055736 004737 061570      JSR      PC,T19SETEXP      ;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
8633 055742 012701 057432      MOV      #T19EXSTA,R1      ;GET EXPECTED READ STATUS
8634 055746 012702 061722      MOV      #T198FSTA,R2      ;GET RCV READ STATUS
8635 055752 013711 002312      MOV      DATA,(R1)      ;SET EXPD WORD #8 = DATA
8636 055756 016261 000002 000002      MOV      2(R2),2(R1)      ;SET EXPD WORD #9 = RCV (NOT TESTING)
8637 055764 005000      CLR      R0      ;HIGH RCV ADDRESS FOR CKMSG2
8638 055766 012701 061702      MOV      #T198FR,R1      ;LOW RCV ADDRESS FOR CKMSG2
8639 055772 012702 057412      MOV      #T19EXP,R2      ;EXPD ADDRESS
8640 055776 012703 000022      MOV      #18.,R3      ;NUMBER OF BYTES TO COMPARE

```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```

8642 056002 004737 011500      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
8643 056006                FORCERROR 242$,NOTSSR      ;@BD
8644 056016 103404            BCS      250$          ;BR IF YES
8645 056020                NEXT.ERRNO
8646 056020 242$:            ERRHRD  ERRNO,T19WSTR,MSGSUB ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD     840
                                .WORD     T19WSTR
                                .WORD     MSGSUB
8647 056030 104406            250$: CKLOOP              ;LOOP ON ERROR, IF FLAG
                                SET      C$CLP1
                                TRAP
8648 056030
8649
8650 056032                FORCEXIT  255$              ;@BD
8651 056042 013703 002316      MOV      TSTPTR,R3      ;RESTORE CURRENT TSTBLK POINTER
8652 056046 020327 003062      CMP      R3,#TBLEND    ;END OF TSTBLK?
8653 056052 103002            BHIS     255$          ;BR IF YES
8654 056054 000137 055064      JMP      100$          ;DO ANOTHER TSTBLK PATTERN
8655 056060 255$:
8656
8657 056060                ENDSUB                      ;////////// END SUBTEST //////////
                                L10071:
                                TRAP      C$ESUB
8658 056060 104403
8659 056062 005737 002222      TST      FATFLG        ;ANY FATAL ERRORS ?
8660 056066 001402            BEQ      260$          ;BRANCH IF NOT
8661 056070 004737 017202      JSR      PC,CKDROP      ;TRY TO DROP THE UNIT
8662 056074 260$:

```

```

.SBTTL TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

```

; **
; TEST 8: SUBTEST 4:

```

```

; SUBTEST DESCRIPTION:

```

```

; This subtest verifies the Read Strobe loopback path
; can strobe the data from the Data lines to the FIFO.
; The signal IRESV4 drives IRSTR (read strobe) to write
; from the data lines to the FIFO.

```

```

; TEST STEPS:

```

```

; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
; BEGIN

```

```

; Write to TSSR register to soft initialize the controller
; Do WRITE CHARACTERISTICS to check for Extended Features Switch
; If Extended Features Hardware Switch Clear then:
; Do Write Subsystem Write Miscellaneous to Set Extended Features.
; Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
; Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
; Do a WRITE NPR to set loopback and tape direction OUT
; Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
; Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
; Do a READ FIFO with tape direction OUT to load tape out write latch
; Do a WRITE NPR to set loopback and tape direction IN
; Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 to write loop data to FIFO

```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

8692      ;      Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
8693      ;      (to strobe loopback data into FIFO.)
8694      ;      Do a READ FIFO with tape direction IN to read data
8695      ;      If Data read from FIFO NOT= to Data sent Then Print Error
8696      ;      END
8697      ;--
8698 056074      BGNSUB                      ;////////// BEGIN SUBTEST //////////
      056074      T8.4:                      TRAP      C#BSUB
      056074 104402
8699      ;      Write to TSSR register to soft initialize the controller
8700 056076      5$:
8701 056076 004737 015774      JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
8702 056102 103405      BCS      10$      ;BR IF SOFT INIT OKAY
8703 056104 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
8704 056106      ERDFF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
      056106 104455      TRAP      C#ERDFF
      056110 001510      .WORD      840
      056112 003652      .WORD      SFIERR
      056114 C12034      .WORD      SFIMSG
8705      ;      Do WRITE CHARACTERISTICS to check for Extended Features Switch
8706 056116 005037 002222      10$:      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
8707 056122 012704 061660      MOV      @T19PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
8708 056126 004737 010662      JSR      PC,WRITCHR      ;DO WRITE CHARACTERISTICS COMMAND
8709 056132      FORCERROR      12$      ;BDDFORCE ERROR IF FORCER=1
8710 056146 103407      BCS      15$      ;BR IF CARRY SET (GOOD RETURN)
8711 056150 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
8712 056152      NEXT.ERRNO
8713 056152      12$:      ERDFF      ERRNO,T19SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      056152 104455      TRAP      C#ERDFF
      056154 001511      .WORD      841
      056156 057573      .WORD      T19SSR
      056160 012046      .WORD      PKTSSR
8714 056162 005237 002222      15$:      INC      FATFLG      ;SET FATAL ERROR FLAG
8715 056166      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      056166 104406      TRAP      C#CLP1
8716      ;      If Extended Features Hardware Switch Clear then:
8717      ;      Do Write Subsystem Write Miscellaneous to Set Extended Features.
8718 056170 012701 061702      MOV      @T19BFR,R1      ;MESSAGE BUFFER ADDRESS
8719 056174 032761 000200 000012      BIT      @X2.EXTF,XST2(R1)      ;EXTENDED FEATURES SWITCH SET?
8720 056202 001026      BNE      30$      ;BR IF YES
8721 056204 004737 061532      JSR      PC,T19SEXT      ;SETUP PACKET FOR WRITE MISC INVERT
8722 056210 012704 062030      MOV      @T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
8723 056214 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
8724 056220 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
8725 056224      FORCERROR      22$      ;BDDFORCE ERROR IF FORCER=1
8726 056240 103407      BCS      30$      ;BR IF CARRY SET (GOOD RETURN)
8727 056242 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
8728 056244      NEXT.ERRNO
8729 056244      22$:      ERDFF      ERRNO,T192SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      056244 104455      TRAP      C#ERDFF
      056246 001512      .WORD      842
      056250 057630      .WORD      T192SSR
      056252 012046      .WORD      PKTSSR
8730 056254 005237 002222      30$:      INC      FATFLG      ;SET FATAL ERROR FLAG
8731 056260      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      056260 104406      TRAP      C#CLP1
8732      ;      Do WRITE CHARACTERISTICS to select reserved unit 7

```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

8733	056262	012704	061660		MOV	#T19PACKET,R4		;GET THE ADDRESS OF COMMAND PACKET
8734	056266	004737	010662		JSR	PC,WRTCHR		;DO WRITE CHARACTERISTICS COMMAND
8735	056272				FORCERROR	42#		;B0DFORCE ERROR IF FORCER=1
8736	056306	103407			BCS	50#		;BR IF CARRY SET (GOOD RETURN)
8737	056310	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR
8738	056312				NEXT.ERRNO			
8739	056312			42#:	ERRDF	ERRNO,T19SSR,PKTSSR		;DEVICE FATAL SSR FAILED TO SET
	056312	104455					TRAP	C#ERDF
	056314	001513					.WORD	843
	056316	057573					.WORD	T19SSR
	056320	012046					.WORD	PKTSSR
8740	056322	005237	002222		INC	FATFLG		;SET FATAL ERROR FLAG
8741	056326			50#:	CKLOOP			;LOOP ON ERROR, IF FLAG SET
	056326	104406					TRAP	C#CLP1
8742								
8743								
8744	056330	012703	002752					
8745	056334	012337	002312		MOV	#TSTBLK,R3		;GET FIRST PATTERN ADDRESS
8746	056340	C42737	177400	002312	100#:	(R3)+,DATA		;GET A TEST PATTERN
8747	056346	010337	002316		BIC	#C<377>,DATA		;DATA IS BYTE
8748					MOV	R3,TSTPTR		;SETUP CURRENT TSTBLK POINTER
8749	056352	012700	000100					;Do a WRITE NPR to set loopback and tape direction OUT
8750	056356	052700	000040		MOV	#NP.OUT,R0		;SET TAPE DIRECTION OUT
8751	056362	004737	061372		BIS	#NP.LOOP,R0		;SET LOOPBACK
8752	056366	012704	062030		JSR	PC,T19SNPR		;SETUP T19PK2 FOR WRITE NPR
8753	056372	010465	000000		MOV	#T19PK2,R4		;GET WRITE SUBSYSTEM COMMAND PACKET
8754	056376	004737	016336		MOV	R4,TSDB(R5)		;SET THE PACKET ADDRESS TO EXECUTE
8755	056402				JSR	PC,CHKTSSR		;WAIT FOR SSR TO SET
8756	056416	103407			FORCERROR	102#		;B0DFORCE ERROR IF FORCER=1
8757	056420	010001			BCS	105#		;BR IF CARRY SET (GOOD RETURN)
8758	056422				MOV	R0,R1		;SAVE CONTENTS OF TSSR
8759	056422			102#:	NEXT.ERRNO			
	056422	104455			ERRDF	ERRNO,T194SSR,PKTSSR		;DEVICE FATAL SSR FAILED TO SET
	056424	001514					TRAP	C#ERDF
	056426	057741					.WORD	844
	056430	012046					.WORD	T194SSR
	056430	012046					.WORD	PKTSSR
8760	056432	005237	002222		INC	FATFLG		;SET FATAL ERROR FLAG
8761	056436			105#:	CKLOOP			;LOOP ON ERROR, IF FLAG SET
	056436	104406					TRAP	C#CLP1
8762								
8763	056440	012700	000001					
8764	056444	004737	061512		Do a WRITE FORMAT to set IRESV4==>IRSTR = 1			
8765	056450	012704	062030		MOV	#WF.I4RES,R0		;IRESV4==>IRSTR=1
8766	056454	010465	000000		JSR	PC,T19WFM		;SETUP T9PK2 FOR WRITE FORMAT
8767	056460	004737	016336		MOV	#T19PK2,R4		;GET WRITE SUBSYSTEM COMMAND PACKET
8768	056464				MOV	R4,TSDB(R5)		;SET THE PACKET ADDRESS TO EXECUTE
8769	056500	103407			JSR	PC,CHKTSSR		;WAIT FOR SSR TO SET
8770	056502	010001			FORCERROR	112#		;B0DFORCE ERROR IF FORCER=1
8771	056504				BCS	120#		;BR IF CARRY SET (GOOD RETURN)
8772	056504			112#:	MOV	R0,R1		;SAVE CONTENTS OF TSSR
	056504	104455			NEXT.ERRNO			
	056506	001515			ERRDF	ERRNO,T198SSR,PKTSSR		;DEVICE FATAL SSR FAILED TO SET
	056510	060162					TRAP	C#ERDF
	056512	012046					.WORD	845
	056512	012046					.WORD	T198SSR
	056512	012046					.WORD	PKTSSR
8773	056514	005237	002222		INC	FATFLG		;SET FATAL ERROR FLAG
8774	056520			120#:	CKLOOP			;LOOP ON ERROR, IF FLAG SET
	056520	104406					TRAP	C#CLP1

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

8775 ; Do a WRITE FIFO with byte count equal to 1 and tape direction OUT
8776 056522 012700 000001 MOV #1,R0 ;WRITE 1 BYTE
8777 056526 012701 002312 MOV #DATA,R1 ;FIFO WRITE DATA ADDRESS
8778 056532 004737 061436 JSR PC,T19WFIF ;SETUP T19PK2 FOR WRITE FIFO
8779 056536 012704 062030 MOV #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8780 056542 010465 000000 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8781 056546 004737 016336 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
8782 056552 FORCERROR 132# ;B80FORCE ERROR IF FORCER=1
8783 056566 103407 BCS 140# ;BR IF CARRY SET (GOOD RETURN)
8784 056570 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
8785 056572 NEXT.ERRNO
8786 056572 132# ERRDF ERRNO,T195SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP C#ERDF
                                .WORD 846
                                .WORD T195SSR
                                .WORD PKTSSR
8787 056602 005237 002222 INC FATFLG ;SET FATAL ERROR FLAG
8788 056606 104406 140# CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                TRAP C#CLP1
8789 ; Do a READ FIFO with tape direction OUT to load tape out write latch
8790 056610 012700 000001 MOV #1,R0 ;SET READ BYTE COUNT
8791 056614 004737 061416 JSR PC,T19RFIF ;SETUP T19PK2 FOR READ FIFO
8792 056620 012704 062030 MOV #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8793 056624 010465 000000 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8794 056630 004737 016336 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
8795 056634 FORCERROR 152# ;B80FORCE ERROR IF FORCER=1
8796 056650 103407 BCS 160# ;BR IF CARRY SET (GOOD RETURN)
8797 056652 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
8798 056654 NEXT.ERRNO
8799 056654 152# ERRDF ERRNO,T196SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP C#ERDF
                                .WORD 847
                                .WORD T196SSR
                                .WORD PKTSSR
8800 056664 005237 002222 INC FATFLG ;SET FATAL ERROR FLAG
8801 056670 104406 160# CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                TRAP C#CLP1
8802 ; Do a WRITE NPR to set loopback and tape direction IN
8803 056672 005000 CLR R0 ;CLR NP.OUT TO SET TAPE DIRECTION IN
8804 056674 052700 000040 BIS #NP.LOOP,R0 ;SET LOOPBACK
8805 056700 004737 061372 JSR PC,T19SNPR ;SETUP T19PK2 FOR WRITE NPR
8806 056704 012704 062030 MOV #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8807 056710 010465 000000 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8808 056714 004737 016336 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
8809 056720 FORCERROR 182# ;B80FORCE ERROR IF FORCER=1
8810 056734 103407 BCS 190# ;BR IF CARRY SET (GOOD RETURN)
8811 056736 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
8812 056740 NEXT.ERRNO
8813 056740 182# ERRDF ERRNO,T194SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP C#ERDF
                                .WORD 848
                                .WORD T194SSR
                                .WORD PKTSSR
8814 056750 005237 002222 INC FATFLG ;SET FATAL ERROR FLAG
8815 056754 104406 190# CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                TRAP C#CLP1
8816 ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 0

```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

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

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

8859 057216 013711 002312      MOV      DATA,(R1)      ;SET EXPD WORD #8 = DATA
8860 057222 016261 000002 000002  MOV      2(R2),2(R1)      ;SET EXPD WORD #9 = RECV (NOT TESTING)
8861 057230 005000              CLR      R0              ;HIGH RECV ADDRESS FOR CKMSG2
8862 057232 012701 061702      MOV      #T198FR,R1      ;LOW RECV ADDRESS FOR CKMSG2
8863 057236 012702 057412      MOV      #T19EXP,R2      ;EXPD ADDRESS
8864 057242 012703 000022      MOV      #18.,R3      ;NUMBER OF BYTES TO COMPARE
8865 057246 004737 011500      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
8866 057252              FORCERROR 302$,NOTSSR      ;###
8867 057262 103404              BCS      310$      ;BR IF YES
8868 057264              NEXT.ERRNO
8869              302$:      ERRHRD  ERRNO,T19RSTR,MSGSUB      ;REPORT ERROR
                        057264 104456              TRAP      C$ERHRD
                        057266 001524              .WORD      852
                        057270 061200              .WORD      T19RSTR
                        057272 013742              .WORD      MSGSUB
8870 057274 104406              310$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                        057274              TRAP      C$CLP1
8871
8872
8873 057276              FORCEEXIT 355$      ;###
8874 057306 013703 002316      MOV      TSTPTR,R3      ;RESTORE CURRENT TSTBLK POINTER
8875 057312 020327 003062      CMP      R3,#TBLEND      ;END OF TSTBLK?
8876 057316 103002              BHS      355$      ;BR IF YES
8877 057320 000137 056334      JMP      100$      ;DO ANOTHER TSTBLK PATTERN
8878 057324              355$:
8879
8880 057324      ENDSUB      ;////////// END SUBTEST ///////////
                        057324              L10072:
                        057324 104403              TRAP      C$ESUB
8881
8882 057326 005737 002222      TST      FATFLG      ;ANY FATAL ERRORS ?
8883 057332 001402              BEQ      360$      ;BRANCH IF NOT
8884 057334 004737 017202      JSR      PC,CKDROP      ;TRY TO DROP THE UNIT
8885 057340              360$:
8886
8887 057340      EXIT      TST      ;////////// EXIT TEST ///////////
                        057340 104432              TRAP      C$EXIT
                        057342 002602              .WORD      L10066-.
8888
8889
8890      ;*
8891      ;LOCAL STORAGE FOR THIS TEST
8892      ;-
8893
8894 057344 000000      T19PREV:      .WORD      0      ;DRIVE SIGNAL 1-0 TRANSITION FLAG
8895      ;*
8896      ; LOOPBACK DRIVE SIGNAL TABLE
8897      ; THIS TABLE IS USED BY T19CNVT TO SETUP
8898      ; A DRIVE PATTERN FROM THE TEST DATA INPUT PATTERN.
8899
8900      ; WRITE CONTROL SIGNALS ARE OF FORM WC.XXX
8901      ; WRITE FORMAT SIGNALS ARE OF FORM WF.XXXX
8902      ;-
8903 057346      T19BCTL:      WC.IGO      ;WRITE CONTROL DRIVE SIGNALS
8904 057346 000001      WC.IFEN      ;IGO==>IFPT      DATA<0>
8905 057350 000002      WC.IRWU      ;IFEN==>IFBY      DATA<1>
8906 057352 000004      WC.IRWU      ;IRWU==>IRWD      DATA<2>

```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

8907 057354 000010      WC.IREW      ;IREW==>IDBY      DATA<3>
8908 057356 002000      WF.IERASE*256. ;IFAD==>ILOP      DATA<4>
8909 057360 000040      WC.IITAD      ;ITAD1==>IONL      DATA<5>
8910 057362 000100      WC.IOTAD      ;ITAD0==>IRDY      DATA<6>
8911 057364 000200      WC.IFAD       ;IERASE==>ISPEED DATA<7>
8912 057366 004000      WF.IEDIT*256. ;IEDIT==>IHER      DATA<8>
8913 057370 010000      WF.IWFM*256.  ;IWFM==>IFMK       DATA<9>
8914 057372 020000      WF.IREV*256.  ;IREV==>ICER       DATA<10>
8915 057374 040000      WF.IWRT*256.  ;IWRT==>IIDENT     DATA<11>
8916 057376 100000      WF.IHISP*256. ;IHISP==>IEOT      DATA<12>
8917 057400 000000      .WORD 0        ;IRESV2 (UNUSED) DATA<13>
8918 057402 000000      .WORD 0        ;IRESV1 (UNUSED) DATA<14>
8919 057404 000000      .WORD 0        ;PARERR (UNTESTED) DATA<15>
8920
8921 057406      T19MSK:      ;MASK OF UNTESTED BITS IN READ STATUS BYTES
8922      ;UNTESTED BITS ARE SET TO 1
8923 057406      377          .BYTE +C<000> ;BYTE 0 MASK
8924 057407      037          .BYTE +C<340> ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
8925 057410      360          .BYTE +C<017> ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
8926 057411      000          .BYTE 0      ;MAKE IT EVEN
8927
8928 057412      T19EXP:      ;BEGIN EXPECTED DATA BUFFER
8929 057412 000000      .WORD 0          ;MESSAGE TYPE
8930 057414 000000      .WORD 0          ;DATA FIELD LENGTH
8931 057416 000000      .WORD 0          ;RBPGR
8932 057420 000000      .WORD 0          ;XST0
8933 057422 000000      .WORD 0          ;XST1
8934 057424 000000      .WORD 0          ;XST2
8935 057426 000000      .WORD 0          ;XST3
8936 057430 000000      .WORD 0          ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
8937 057432      T19EXSTA: .BLKB 64.      ;EXPECTED READ STATUS AND WRITE FIFO DATA
8938 057532      T19EXEND:      ;END EXPECTED DATA BUFFER
8939
8940      ;*
8941      ;LOCAL TEXT MESSAGES FOR TEST
8942      ;-
8943 057532      124      162      141 TST19ID: .ASCIZ 'Transport Bus Interface Loopback'
8944 057573      127      122      111 T19SSR: .ASCIZ 'WRITE CHARACTERISTICS Failed'
8945 057630      127      122      111 T192SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Misc) Failed'
8946 057674      127      122      111 T193SSR: .ASCIZ 'WRITE SUBSYSTEM (Read Status) Failed'
8947 057741      127      122      111 T194SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Npr) Failed'
8948 060004      127      122      111 T195SSR: .ASCIZ 'WRITE SUBSYSTEM (Write FIFO) Failed'
8949 060050      127      122      111 T196SSR: .ASCIZ 'WRITE SUBSYSTEM (Read FIFO) Failed'
8950 060113      127      122      111 T197SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Control) Failed'
8951 060162      127      122      111 T198SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Format) Failed'
8952 060230      106      111      106 T191CMP: .ASCIZ 'FIFO Status in WORD #9 Incorrect after Initialize'
8953 060312      122      145      141 T192CMP: .ASCIZ 'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
8954 060406      124      141      160 T193CMP: .ASCIZ 'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
8955 060474      122      145      141 T195CMP: .ASCIZ 'Read FIFO Data not equal to Write FIFO Data'
8956 060550      106      111      106 T196CMP: .ASCIZ 'FIFO Status (in WORD #9) Incorrect after READ FIFO'
8957 060633      124      141      160 T197CMP: .ASCIZ 'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
8958 060721      103      157      156 T198CMP: .ASCIZ 'Control Signal Loopback Data Error, Data is in WORD #8'
8959 061010      122      145      141 T199CMP: .ASCIZ 'Read/Write Loopback Data Error, Data is in WORD #8'
8960 061073      114      157      157 T19WSTR: .ASCIZ 'Loopback Data Error when strobed by Write strobe, Data is in WORD #8'
8961 061200      114      157      157 T19RSTR: .ASCIZ 'Loopback Data Error when strobed by Read Strobe, Data is in WORD #8'
8962
8963      .EVEN

```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

8964
8965
8966
8967
8968 061304
8969 061304
8970 061310 012701 061702
8971 061314 012702 000120
8972 061320 105021
8973 061322 005302
8974 061324 003375
8975 061326 000207
8976
8977
8978
8979
8980 061330
8981 061330 004737 061304
8982 061334 012700 062040
8983 061340 112720 000005
8984 061344 105010
8985 061346 000207
8986
8987
8988
8989
8990 061350
8991 061350 004737 061304
8992 061354 012700 062040
8993 061360 112720 000010
8994 061364 112710 000030
8995 061370 000207
8996
8997
8998
8999
9000
9001
9002
9003
9004
9005 061372
9006 061372 004737 061304
9007 061376 012701 062040
9008 061402 112721 000011
9009 061406 052700 000020
9010 061412 110011
9011 061414 000207
9012
9013
9014
9015
9016
9017
9018
9019 061416
9020 061416 004737 061304

;+
; CLEAR MESSAGE BUFFER
;-
T19CLRBUF:
    SAVREG
    MOV     #T198FR,R1
    MOV     #T198END-T198FR,R2
104:    CLRB  (R1)+
    DEC     R2
    BGT     104
    RTS     PC
; SAVE R1-R5 UNTIL NEXT RETURN
; GET MESSAGE BUFFER ADDRESS
; SIZE OF MESSAGE BUFFER IN BYTES
; CLEAR A BYTE
; DONE?
; BR IF NO
; RETURN

;+
; SETUP T19PK2 PACKET FOR READ STATUS
;-
T19SRD:
    JSR     PC,T19CLRBUF
    MOV     #T19DT2,R0
    MOV     #PW.RDSTATUS,(R0)+
    CLR     (R0)
    RTS     PC
; CLEAR MESSAGE BUFFER
; WRITE SUBSYSTEM DATA BUFFER
; STORE READ STATUS COMMAND IN BSEL0
; CLEAR BSEL1
; RETURN

;+
; SETUP T19PK2 PACKET FOR WRITE MISC Reset Tape Status F-FLOPS
;-
T19RSFIF:
    JSR     PC,T19CLRBUF
    MOV     #T19DT2,R0
    MOV     #PW.WMISC,(R0)+
    MOV     #MS.RSFIF!MS.RSTAP,(R0)
    RTS     PC
; CLEAR MESSAGE BUFFER
; WRITE SUBSYSTEM DATA BUFFER
; STORE WRITE MISCELLANEOUS IN BSEL0
; STORE BSEL1 CLEAR FIFO CODES
; RETURN

;+
; SETUP T19PK2 PACKET FOR WRITE NPR
;-
; INPUT:
; R0 CONTAINS BSEL1 NPR DATA
;
; SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
;-
T19SNPR:
    JSR     PC,T19CLRBUF
    MOV     #T19DT2,R1
    MOV     #PW.WNPR,(R1)+
    BIS     #NP.WRP,R0
    MOV     R0,(R1)
    RTS     PC
; CLEAR MESSAGE BUFFER
; WRITE SUBSYSTEM DATA BUFFER
; STORE WRITE NPR IN BSEL0
; DON'T WRITE WRONG PARITY
; STORE NPR DATA IN BSEL1
; RETURN

;+
; SETUP T19PK2 PACKET FOR READ FIFO
;-
; INPUT:
; R0 CONTAINS SEL2 BYTE COUNT
;-
T19RFIF:
    JSR     PC,T19CLRBUF
; CLEAR MESSAGE BUFFER

```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

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

```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

9078 061550
9079 061550 012701 057412
9080 061554 012700 000120
9081 061560 105021
9082 061562 005300
9083 061564 003375
9084 061566 000207
9085
9086
9087
9088
9089 061570
9090 061570 012702 057412
9091 061574 012703 061702
9092 061600 012700 000010
9093 061604 012322
9094 061606 005300
9095 061610 003375
9096 061612 000207
9097
9098
9099
9100
9101
9102
9103
9104
9105
9106
9107
9108
9109
9110
9111
9112
9113
9114 061614
9115 061614
9116 061620 012701 057346
9117 061624 005002
9118 061626 012703 000020
9119 061632 006000
9120 061634 103001
9121 061636 051102
9122 061640 005721
9123 061642 005303
9124 061644 003372
9125 061646 010200
9126 061650 000207
9127
9128
9129
9131 061660
9133
9134
9135
9136 061660

T19CLEXP:
MOV #T19EXP,R1 ;GET EXPD ADDRESS
MOV #T19XEND-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 EXF=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:
;
; T19BCTL - 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 #T19BCTL,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

;+
;WRITE CHARACTERISTICS COMMAND PACKET
;-
T19PACKET:
;COMMAND PACKET FOR TEST

```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

9137 061660 100004      .WORD 100004      ;WRITE CHARACTERISTICS COMMAND, WITH ACK
9138 061662 061670      .WORD T19DATA     ;ADDRESS OF CHARACTERISTICS BLOCK
9139 061664 000000      .WORD 0
9140 061666 000012      .WORD 10.         ;MINIMUM MESSAGE PACKET SIZE
9141
9142 061670      T19DATA:      ;CHARACTERISTICS DATA BLOCK
9143 061670 061702      .WORD T19BFR      ;ADDRESS OF MESSAGE BUFFER
9144 061672 000000      .WORD 0
9145 061674 000024      .WORD 20.         ;LENGTH OF MESSAGE BUFFER
9146 061676 000000      .WORD 0           ;ESS,ENB,EAI,ERI
9147 061700 000007      .WORD 7           ;EXTENDED FEATURES UNIT NO.
9148
9149
9150      ;MESSAGE BUFFER FOR ALL TEST 8 COMMANDS
9151
9152 061702      T19BFR:      ;BEGIN MESSAGE BUFFER
9153 061702 000000      .WORD 0           ;MESSAGE TYPE
9154 061704 000000      .WORD 0           ;DATA FIELD LENGTH
9155 061706 C00000      .WORD 0           ;RBPGR
9156 061710 000000      .WORD 0           ;XST0
9157 061712 000000      .WORD 0           ;XST1
9158 061714 000000      .WORD 0           ;XST2
9159 061716 000000      .WORD 0           ;XST3
9160 061720 000000      .WORD 0           ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
9161 061722      T19BFSTA: .BLKB 64.      ;READ STATUS AND WRITE FIFO BUFFER
9162 062022      T19BEND:      ;END OF MESSAGE BUFFER
9163
9164      ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
9165
9167      .=<.10>&177770
9169 062030      T19PK2:      ;WRITE SUBSYSTEM WITH ACK
9170 062030 100006      .WORD P.WRTSUB!P.ACK ;LOW ADDRESS OF DATA BLOCK
9171 062032 062040      .WORD T19DT2      ;HIGH ADDRESS OF DATA BLOCK
9172 062034 000000      .WORD 0           ;MINIMUM MESSAGE PACKET SIZE
9173 062036 000012      .WORD 10.
9174
9175 062040      T19DT2:      ;DATA BLOCK
9176 062040      .BYTE 0           ;BSEL0
9177 062041      .BYTE 0           ;BSEL1
9178 062042 000000      .WORD 0           ;SEL2
9179 062044      .BLKB 64.         ;WRITE FIFO DATA OUTPUT BUFFER
9180
9181
9182 062144      ENDTST
9183      L10066:      TRAP      C#ETST
9184      .SBTTL TEST 9: READ/WRITE DATA PARITY TEST
9185
9186      ;**
9187      ; TEST DESCRIPTION:
9188      ;
9189      ; This test verifies that the Write Data Parity generator
9190      ; and the Read Data Parity checker operate properly. The
9191      ; Transport Bus signal loopback mode is enabled and a
9192      ; Set Wrong parity function is executed. Then various
9193      ; Write Subsystem Memory functions are performed to
9194      ; write data to and from the FIFO in loopback mode.
9195      ; The program then checks to insure a Read Data parity

```

TEST 9: READ/WRITE DATA PARITY TEST

```

9194      : error occurred.
9195      : A Reset FIFO is done and the Read Data parity
9196      : error bit is again tested to insure it cleared.
9197      : Finally a Clear wrong parity function is done
9198      : and it is verified the data word can pass in loopback
9199      : mode without setting Read Data parity error.
9200      :
9201      : TEST STEPS:
9202      :
9203      : REPEAT FOR LOOPCNT
9204      : BEGIN
9205      :   Write to TSSR register to soft initialize the controller
9206      :   Do WRITE CHARACTERISTICS to check for Extended Features Switch
9207      :   If Extended Features Hardware Switch Clear then:
9208      :     Do Write Subsystem Write Miscellaneous to Set Extended Features.
9209      :     Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
9210      : REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
9211      : BEGIN
9212      :   (* Verify Write Wrong Parity Sets Parity Error *)
9213      :   Do a WRITE NPR to set loopback and tape direction OUT
9214      :     and SET Write Wrong Parity.
9215      :   Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
9216      :   Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
9217      :   Do a READ FIFO with tape direction OUT to load tape out write latch
9218      :     (this is when wrong parity (IWP) is set)
9219      :   Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 (sets read strobe low)
9220      :     (Read Strobe sets PAR IN H [Parity Error])
9221      :   Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
9222      :   Do a Write Subsystem READ STATUS
9223      :   If Read Data parity error NOT=1 Then Print Error
9224      :   Do a Write Misc to RESET FIFO
9225      :   Do a Write Subsystem READ STATUS
9226      :   If Read Data parity error NOT=0 Then Print Error
9227      :
9228      :   (* Verify Data can be transferred without a Parity Error *)
9229      :   Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
9230      :   Do a WRITE NPR to set loopback and tape direction OUT
9231      :     and CLEAR Write Wrong Parity.
9232      :   Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
9233      :   Do a READ FIFO with tape direction OUT to load tape out write latch
9234      :   Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 (sets read strobe low)
9235      :     (Read Strobe should NOT set PAR IN H [Parity Error] here)
9236      :   Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
9237      :   Do a Write Subsystem READ STATUS
9238      :   If Read Data parity error NOT=0 Then Print Error
9239      :
9240      : END
9241      :--
9242
9243
9244 062146      BGNTST
9249 062146      T9::
9250 062146 012700 064532      MOV      @TST20ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
9251 062152 004737 016510      JSR      PC,TSTSETUP      ;DO INITIAL TEST SETUP
9252 062156 012737 000012 002216      MOV      @10.,LOOPCNT      ;PERFORM 10 ITERATIONS
9253

```

TEST 9: READ/WRITE DATA PARITY TEST

```

9254 062164          BGNSUB                      ;////////// BEGIN SUBTEST //////////
          062164          T9.1:
          062164 104402          TRAP      C#BSUB
9255          ;      Write to TSSR register to soft initialize the controller
9256 062166          ;
9257 062166 004737 015774      5$:      JSR      PC,SOFINIT          ;WRITE TO TSSR TO SOFT INITIALIZE
9258 062172          BCS      10$          ;BR IF SOFT INIT OKAY
9259 062174          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
9260 062176          ERDFF   ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
          062176 104455          TRAP      C#ERDF
          062200          .WORD      900
          062202          .WORD      SFIERR
          062204          .WORD      SFIMSG
9261          ;      Do WRITE CHARACTERISTICS to check for Extended Features Switch
9262 062206 005037 002222      10$:      CLR      FATFLG          ;CLEAR FATAL ERROR FLAG
9263 062212          MOV      #T20PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
9264 062216 004737 010662      JSR      PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
9265 062222          FORCERROR 12$          ;FORCE ERROR IF FORCER=1
9266 062236          BCS      15$          ;BR IF CARRY SET (GOOD RETURN)
9267 062240          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
9268 062242          NEXT.ERRNO
9269          12$:      ERDFF   ERRNO,T20SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
          062242 104455          TRAP      C#ERDF
          062244          .WORD      901
          062246          .WORD      T20SSR
          062250          .WORD      PKTSSR
9270 062252 005237 002222      15$:      INC      FATFLG          ;SET FATAL ERROR FLAG
9271 062256          CKLOOP          ;LOOP ON ERROR, IF FLAG SET
          062256 104406          TRAP      C#CLP1
9272          ;      If Extended Features Hardware Switch Clear then:
9273          ;      Do Write Subsystem Write Miscellaneous to Set Extended Features.
9274 062260 012701 065752      MOV      #T20BFR,R1          ;MESSAGE BUFFER ADDRESS
9275 062264 032761 000200 000012      BIT      #X2.EXTF,XST2(R1) ;EXTENDED FEATURES SWITCH SET?
9276 062272          BNE      30$          ;BR IF YES
9277 062274          JSR      PC,T20SEXT          ;SETUP PACKET FOR WRITE MISC INVERT
9278 062300          MOV      #T20PK2,R4          ;GET WRITE SUBSYSTEM COMMAND PACKET
9279 062304          MOV      R4,TSD8(R5)          ;SET THE PACKET ADDRESS TO EXECUTE
9280 062310          JSR      PC,CHKTSSR          ;WAIT FOR SSR TO SET
9281 062314          FORCERROR 22$          ;FORCE ERROR IF FORCER=1
9282 062330          BCS      30$          ;BR IF CARRY SET (GOOD RETURN)
9283 062332          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
9284 062334          NEXT.ERRNO
9285          22$:      ERDFF   ERRNO,T202SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
          062334 104455          TRAP      C#ERDF
          062336          .WORD      902
          062340          .WORD      T202SSR
          062342          .WORD      PKTSSR
9286 062344 005237 002222      30$:      INC      FATFLG          ;SET FATAL ERROR FLAG
9287 062350          CKLOOP          ;LOOP ON ERROR, IF FLAG SET
          062350 104406          TRAP      C#CLP1
9288          ;      Do WRITE CHARACTERISTICS to select reserved unit 7
9289 062352 012704 065730      MOV      #T20PACKET,R4          ;GET THE ADDRESS OF COMMAND PACKET
9290 062356          JSR      PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
9291 062362          FORCERROR 42$          ;FORCE ERROR IF FORCER=1
9292 062376          BCS      50$          ;BR IF CARRY SET (GOOD RETURN)
9293 062400          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
9294 062402          NEXT.ERRNO

```

TEST 9: READ/WRITE DATA PARITY TEST

```

9295 062402      424:  ERRDF  ERRNO,T20SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      062402 104455                                     TRAP  C#ERDF
      062404 001607                                     .WORD 903
      062406 064561                                     .WORD T20SSR
      062410 012046                                     .WORD PKTSSR
9296 062412 005237 002222
9297 062416 104406      504:  INC  FATFLG      ;SET FATAL ERROR FLAG
      062416 104406      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                     TRAP  C#CLP1

9298
9299
9300      ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
9301 062420 012703 002752      1004:  MOV  #TSTBLK,R3      ;GET FIRST PATTERN ADDRESS
9302 062424 012337 002312      MOV  (R3)+,DATA      ;GET A TEST PATTERN
9303 062430 042737 177400      BIC  #C<377>,DATA      ;DATA IS BYTE
9304 062436 010337 002316      MOV  R3,TSTPTR      ;SETUP CURRENT TSTBLK POINTER
9305      ; Do a WRITE NPR to set loopback and tape direction OUT and
9306      ; and SET Write Wrong Parity.
9307 062442 012700 000100      MOV  #NP.OUT,R0      ;SET TAPE DIRECTION OUT
9308 062446 C52700 000040      BIS  #NP.LOOP,R0      ;SET LOOPBACK
9309 062452 042700 000020      BIC  #NP.WRP,R0      ;SET WRITE WRONG PARITY (INVERTED)
9310 062456 004737 065516      JSR  PC,T20WNP      ;SETUP T20PK2 FOR WRITE NPR
9311 062462 012704 066100      MOV  #T20PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
9312 062466 010465 000000      MOV  R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
9313 062472 004737 016336      JSR  PC,CHKTSSR      ;WAIT FOR SSR TO SET
9314 062476      FORCERROR 1024      ;B0DFORCE ERROR IF FORCER=1
9315 062512 103407      BCS  1054      ;BR IF CARRY SET (GOOD RETURN)
9316 062514 010001      MOV  R0,R1      ;SAVE CONTENTS OF TSSR
9317 062516      NEXT.ERRNO
9318 062516      1024:  ERRDF  ERRNO,T204SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      062516 104455                                     TRAP  C#ERDF
      062520 001610                                     .WORD 904
      062522 064727                                     .WORD T204SSR
      062524 012046                                     .WORD PKTSSR
9319 062526 005237 002222
9320 062532 104406      1054:  INC  FATFLG      ;SET FATAL ERROR FLAG
      062532 104406      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                     TRAP  C#CLP1
      ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
9321      MOV  #WF.I4RES,R0      ;IRESV4==>IRSTR = 1
9322 062534 012700 000001      JSR  PC,T20WFM      ;SETUP T20PK2 FOR WRITE FORMAT
9323 062540 004737 065612      MOV  #T20PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
9324 062544 012704 066100      MOV  R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
9325 062550 010465 000000      JSR  PC,CHKTSSR      ;WAIT FOR SSR TO SET
9326 062554 004737 016336      FORCERROR 1124      ;B0DFORCE ERROR IF FORCER=1
9327 062560      BCS  1204      ;BR IF CARRY SET (GOOD RETURN)
9328 062574 103407      MOV  R0,R1      ;SAVE CONTENTS OF TSSR
9329 062576 010001      NEXT.ERRNO
9330 062600
9331 062600      1124:  ERRDF  ERRNO,T208SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      062600 104455                                     TRAP  C#ERDF
      062602 001611                                     .WORD 905
      062604 065101                                     .WORD T208SSR
      062606 012046                                     .WORD PKTSSR
9332 062610 005237 002222
9333 062614 104406      1204:  INC  FATFLG      ;SET FATAL ERROR FLAG
      062614 104406      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                     TRAP  C#CLP1
      ; Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
9334      MOV  #1,R0      ;WRITE 1 BYTE
9335 062616 012700 000001      MOV  #DATA,R1      ;FIFO WRITE DATA ADDRESS
9336 062622 012701 002312

```

TEST 9: READ/WRITE DATA PARITY TEST

9337	062626	004737	065556	JSR	PC,T20WFIF	;SETUP T20PK2 FOR WRITE FIFO
9338	062632	012704	066100	MOV	#T20PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
9339	062636	010465	000000	MOV	R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE
9340	062642	004737	016336	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
9341	062646			FORCERROR	152#	;BDDFORCE ERROR IF FORCER=1
9342	062662	103407		BCS	160#	;BR IF CARRY SET (GOOD RETURN)
9343	062664	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR
9344	062666			NEXT.ERRNO		
9345	062666			152#:	ERRDF ERRNO,T205SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	062666	104455				TRAP C#ERDF
	062670	001612				.WORD 906
	062672	064772				.WORD T205SSR
	062674	012046				.WORD PKTSSR
9346	062676	005237	002222	INC	FATFLG	;SET FATAL ERROR FLAG
9347	062702			160#:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
	062702	104406				TRAP C#CLP1
9348				:	Do a READ FIFO with tape direction OUT to load tape out write latch	
9349				:	(this is when wrong parity (IWP) is set)	
9350	062704	C12700	000001	MOV	#1,R0	;SET READ BYTE COUNT
9351	062710	004737	065536	JSR	PC,T20RFIF	;SETUP T20PK2 FOR READ FIFO
9352	062714	012704	066100	MOV	#T20PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
9353	062720	010465	000000	MOV	R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE
9354	062724	004737	016336	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
9355	062730			FORCERROR	172#	;BDDFORCE ERROR IF FORCER=1
9356	062744	103407		BCS	180#	;BR IF CARRY SET (GOOD RETURN)
9357	062746	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR
9358	062750			NEXT.ERRNO		
9359	062750			172#:	ERRDF ERRNO,T206SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	062750	104455				TRAP C#ERDF
	062752	001613				.WORD 907
	062754	065036				.WORD T206SSR
	062756	012046				.WORD PKTSSR
9360	062760	005237	002222	INC	FATFLG	;SET FATAL ERROR FLAG
9361	062764			180#:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
	062764	104406				TRAP C#CLP1
9362				:	Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 (sets read strobe low)	
9363				:	(Read Strobe sets PAR IN H [Parity Error])	
9364	062766	005000		CLR	R0	;IRESV4==>IRSTR = 0
9365	062770	004737	065612	JSR	PC,T20WFMT	;SETUP T20PK2 FOR WRITE FORMAT
9366	062774	012704	066100	MOV	#T20PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
9367	063000	010465	000000	MOV	R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE
9368	063004	004737	016336	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
9369	063010			FORCERROR	192#	;BDDFORCE ERROR IF FORCER=1
9370	063024	103407		BCS	200#	;BR IF CARRY SET (GOOD RETURN)
9371	063026	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR
9372	063030			NEXT.ERRNO		
9373	063030			192#:	ERRDF ERRNO,T208SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	063030	104455				TRAP C#ERDF
	063032	001614				.WORD 908
	063034	065101				.WORD T208SSR
	063036	012046				.WORD PKTSSR
9374	063040	005237	002222	INC	FATFLG	;SET FATAL ERROR FLAG
9375	063044			200#:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
	063044	104406				TRAP C#CLP1
9376				:	Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)	
9377	063046	012700	000001	MOV	#WF.I4RES,R0	;IRESV4==>IRSTR = 1
9378	063052	004737	065612	JSR	PC,T20WFMT	;SETUP T20PK2 FOR WRITE FORMAT

TEST 9: READ/WRITE DATA PARITY TEST

```

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

```

TEST 9: READ/WRITE DATA PARITY TEST

```

9421 063314 012704 066100      MOV      #T20PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
9422 063320 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
9423 063324 004737 016336      JSR      PC,CHKTSSR    ;WAIT FOR SSR TO SET
9424 063330      FORCERROR      282$      ;$$$FORCE ERROR IF FORCER=1
9425 063344 103407      BCS      290$      ;BR IF CARRY SET (GOOD RETURN)
9426 063346 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
9427 063350      NEXT.ERRNO
9428 063350      282$: ERRDF      ERRNO,T202SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
          063350 104455      TRAP      C$ERDF
          063352 001620      .WORD      912
          063354 064616      .WORD      T202SSR
          063356 012046      .WORD      PKTSSR
9429 063360 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
9430 063364      290$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
          063364 104406      TRAP      C$CLP1
9431      ;
9432      ; Do a Write Subsystem READ STATUS
          ; If Read Data parity error NOT=0 Then Print Error
9433 063366 004737 065704      JSR      PC,T20SETEXP    ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
9434 063372 C12701 064432      MOV      #T20EXSTA,R1    ;GET EXPECTED READ STATUS
9435 063376 012702 065772      MOV      #T20BFSTA,R2    ;GET RECV READ STATUS
9436 063402 011211      MOV      (R2),(R1)    ;SET EXPD WORD #8 = RECV TEMP
9437 063404 016261 000002 000002 MOV      2(R2),2(R1)    ;SET EXPD WORD #9 = RECV (NOT TESTED)
9438 063412 042711 100000      BIC      #S1.PARERR,(R1)    ;SET EXP PAR ERR =0
9439 063416 005000      CLR      R0      ;HIGH RECV ADDRESS FOR CKMSG2
9440 063420 012701 065752      MOV      #T20BFR,R1    ;LOW RECV ADDRESS FOR CKMSG2
9441 063424 012702 064412      MOV      #T20EXP,R2    ;EXPD ADDRESS
9442 063430 012703 000024      MOV      #20,,R3      ;NUMBER OF BYTES TO COMPARE
9443 063434 004737 011500      JSR      PC,CKMSG2    ;EXPD EQUAL RECV?
9444 063440      FORCERROR      302$,NOTSSR    ;$$$
9445 063450 103404      BCS      320$      ;BR IF YES
9446 063452      NEXT.ERRNO
9447 063452      302$: ERRHRD      ERRNO,T20RSF,MSGSTAT    ;REPORT ERROR
          063452 104456      TRAP      C$ERHRD
          063454 001621      .WORD      913
          063456 065256      .WORD      T20RSF
          063460 012350      .WORD      MSGSTAT
9448 063462      320$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
          063462 104406      TRAP      C$CLP1
9449      ;
9450      ; (* Verify Data can be transferred without a Parity Error *)
          ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
9451 063464 012700 000001      MOV      #WF.I4RES,R0    ;IRESV4==>IRSTR = 1
9452 063470 004737 065612      JSR      PC,T20WFMT    ;SETUP T20PK2 FOR WRITE FORMAT
9453 063474 012704 066100      MOV      #T20PK2,R4    ;GET WRITE SUBSYSTEM COMMAND PACKET
9454 063500 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
9455 063504 004737 016336      JSR      PC,CHKTSSR    ;WAIT FOR SSR TO SET
9456 063510      FORCERROR      332$      ;$$$FORCE ERROR IF FORCEK=1
9457 063524 103407      BCS      340$      ;BR IF CARRY SET (GOOD RETURN)
9458 063526 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
9459 063530      NEXT.ERRNO
9460 063530      332$: ERRDF      ERRNO,T208SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
          063530 104455      TRAP      C$ERDF
          063532 001622      .WORD      914
          063534 065101      .WORD      T208SSR
          063536 012046      .WORD      PKTSSR
9461 063540 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
9462 063544      340$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
          063544 104406      TRAP      C$CLP1

```

TEST 9: READ/WRITE DATA PARITY TEST

```

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

```

TEST 9: READ/WRITE DATA PARITY TEST

```

064006 104406
9506      ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 (sets read strobe low)
9507      ; (Read Strobe sets PAR IN H [Parity Error])
9508 064010 005000      CLR R0 ; IRESV4==>IRSTR = 0
9509 064012 004737 065612 JSR PC,T20WFMT ; SETUP T20PK2 FOR WRITE FORMAT
9510 064016 012704 066100 MOV #T20PK2,R4 ; GET WRITE SUBSYSTEM COMMAND PACKET
9511 064022 010465 000000 MOV R4,TSD8(R5) ; SET THE PACKET ADDRESS TO EXECUTE
9512 064026 004737 016336 JSR PC,CHKTSSR ; WAIT FOR SSR TO SET
9513 064032      FORCERROR 412# ; ##DFORCE ERROR IF FORCER=1
9514 064046 103407      BCS 420# ; BR IF CARRY SET (GOOD RETURN)
9515 064050 010001      MOV R0,R1 ; SAVE CONTENTS OF TSSR
9516 064052      NEXT.ERRNO
9517 412# : ERRDF ERRNO,T208SSR,PKTSSR ; DEVICE FATAL SSR FAILED TO SET
                                TRAP C#ERDF
                                .WORD 918
                                .WORD T208SSR
                                .WORD PKTSSR
9518 064062 005237 002222 INC FATFLG ; SET FATAL ERROR FLAG
9519 064066      CKLOOP ; LOOP ON ERROR, IF FLAG SET
                                TRAP C#CLP1
9520      ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
9521 064070 012700 000001 MOV #WF.I4RES,R0 ; IRESV4==>IRSTR = 1
9522 064074 004737 065612 JSR PC,T20WFMT ; SETUP T20PK2 FOR WRITE FORMAT
9523 064100 012704 066100 MOV #T20PK2,R4 ; GET WRITE SUBSYSTEM COMMAND PACKET
9524 064104 010465 000000 MOV R4,TSD8(R5) ; SET THE PACKET ADDRESS TO EXECUTE
9525 064110 004737 016336 JSR PC,CHKTSSR ; WAIT FOR SSR TO SET
9526 064114      FORCERROR 432# ; ##DFORCE ERROR IF FORCER=1
9527 064130 103407      BCS 440# ; BR IF CARRY SET (GOOD RETURN)
9528 064132 010001      MOV R0,R1 ; SAVE CONTENTS OF TSSR
9529 064134      NEXT.ERRNO
9530 432# : ERRDF ERRNO,T208SSR,PKTSSR ; DEVICE FATAL SSR FAILED TO SET
                                TRAP C#ERDF
                                .WORD 919
                                .WORD T208SSR
                                .WORD PKTSSR
9531 064144 005237 002222 INC FATFLG ; SET FATAL ERROR FLAG
9532 064150      CKLOOP ; LOOP ON ERROR, IF FLAG SET
                                TRAP C#CLP1
9533 064150 104406
9534      ; Do a Write Subsystem READ STATUS
9535 064152 004737 065476 JSR PC,T20SRD ; SETUP PACKET FOR READ STATUS
9536 064156 012704 066100 MOV #T20PK2,R4 ; GET WRITE SUBSYSTEM COMMAND PACKET
9537 064162 010465 000000 MOV R4,TSD8(R5) ; SET THE PACKET ADDRESS TO EXECUTE
9538 064166 004737 016336 JSR PC,CHKTSSR ; WAIT FOR SSR TO SET
9539 064172      FORCERROR 452# ; ##DFORCE ERROR IF FORCER=1
9540 064206 103407      BCS 460# ; BR IF CARRY SET (GOOD RETURN)
9541 064210 010001      MOV R0,R1 ; SAVE CONTENTS OF TSSR
9542 064212      NEXT.ERRNO
9543 452# : ERRDF ERRNO,T203SSR,PKTSSR ; DEVICE FATAL SSR FAILED TO SET
                                TRAP C#ERDF
                                .WORD 920
                                .WORD T203SSR
                                .WORD PKTSSR
9544 064222 005237 002222 INC FATFLG ; SET FATAL ERROR FLAG
9545 064226      CKLOOP ; LOOP ON ERROR, IF FLAG SET
                                TRAP C#CLP1
9546      ; If Read Data parity error NOT=0 Then Print Error

```

TEST 9: READ/WRITE DATA PARITY TEST

9547	064230	004737	065704	JSR	PC,T20SETEXP	;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
9548	064234	012701	064432	MOV	#T20EXSTA,R1	;GET EXPECTED READ STATUS
9549	064240	012702	065772	MOV	#T20BFSTA,R2	;GET RECV READ STATUS
9550	064244	011211		MOV	(R2),(R1)	;SET EXPD WORD #8 = RECV TEMP
9551	064246	016261	000002	MOV	2(R2),2(R1)	;SET EXPD WORD #9 = RECV (NOT TESTED)
9552	064254	042711	100000	BIC	#S1.PARERR,(R1)	;SET EXP PAR ERR =0
9553	064260	005000		CLR	R0	;HIGH RECV ADDRESS FOR CKMSG2
9554	064262	012701	065752	MOV	#T20BFR,R1	;LOW RECV ADDRESS FOR CKMSG2
9555	064266	012702	064412	MOV	#T20EXP,R2	;EXPD ADDRESS
9556	064272	012703	000024	MOV	#20.,R3	;NUMBER OF BYTES TO COMPARE
9557	064276	004737	011500	JSR	PC,CKMSG2	;EXPD EQUAL RECV?
9558	064302			FORCERROR	472#,NOTSSR	;880
9559	064312	103404		BCS	480#	;BR IF YES
9560	064314			NEXT.ERRNO		
9561	064314			ERRHRD	ERRNO,T20CWP,MSGSTAT	;REPORT ERROR
	064314	104456				TRAP C#ERHRD
	064316	001631				.WORD 921
	064320	065357				.WORD T20CWP
	064322	C12350				.WORD MSGSTAT
9562	064324					SET
	064324	104406		480#:	CKLOOP	TRAP C#CLP1
9563						
9564	064326			FORCEEXIT	555#	;880
9565	064336	013703	002316	MOV	TSTPTR,R3	;RESTORE CURRENT TSTBLK POINTER
9566	064342	020327	003062	CMP	R3,#TBLEND	;END OF TSTBLK?
9567	064346	103002		BHIS	555#	;BR IF YES
9568	064350	000137	062424	JMP	100#	;DO ANOTHER TSTBLK PATTERN
9569	064354			555#:		
9570						
9571	064354			ENDSUB		;/!!!!!!!!!!!! END SUBTEST !!!!!!!!!!!!!
	064354					L10074:
	064354	104403				TRAP C#ESUB
9572						
9573	064356	005737	002222	TST	FATFLG	;ANY FATAL ERRORS ?
9574	064362	001402		BEQ	560#	;BRANCH IF NOT
9575	064364	004737	017202	JSR	PC,CKDROP	;TRY TO DROP THE UNIT
9576	064370			560#:		
9577	064370	004737	016456	JSR	PC,TSTLOOP	;DO ITERATIONS?
9578	064374	103002		BCC	565#	;BR IF NO
9579	064376	000137	050056	JMP	T18LOOP	;LOOP UNTIL ITERATIONS DONE
9580	064402			565#:		
9581	064402			EXIT	TST	;/!!!!!!!!!!!! EXIT TEST !!!!!!!!!!!!!
	064402	104432				TRAP C#EXIT
	064404	001610				.WORD L10073-
9582						
9583						
9584				;		
9585				LOCAL STORAGE FOR THIS TEST		
9586				;		
9587						
9588						
9589	064406			T20MSK:		;MASK OF UNTESTED BITS IN READ STATUS
9590						;UNTESTED BITS ARE SET TO 1
9591	064406	377		.BYTE	#C<000>	;BYTE 0 MASK
9592	064407	037		.BYTE	#C<340>	;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
9593	064410	360		.BYTE	#C<017>	;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
9594	064411	000		.BYTE	0	;MAKE IT EVEN

TEST 9: READ/WRITE DATA PARITY TEST

```

9595
9596 064412
9597 064412 000000
9598 064414 000000
9599 064416 000000
9600 064420 000000
9601 064422 000000
9602 064424 000000
9603 064426 000000
9604 064430 000000
9605 064432
9606 064532
9607
9608
9609
9610
9611 064532 122 145 141 TST20ID: .ASCIZ 'Read/Write Data Parity'
9612 064561 127 122 111 T20SSR: .ASCIZ 'WRITE CHARACTERISTICS Failed'
9613 064616 127 122 111 T202SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Misc) Failed'
9614 064662 127 122 111 T203SSR: .ASCIZ 'WRITE SUBSYSTEM (Read Status) Failed'
9615 064727 127 122 111 T204SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Npr) Failed'
9616 064772 127 122 111 T205SSR: .ASCIZ 'WRITE SUBSYSTEM (Write FIFO) Failed'
9617 065036 127 122 111 T206SSR: .ASCIZ 'WRITE SUBSYSTEM (Read FIFO) Failed'
9618 065101 127 122 111 T208SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Format) Failed'
9619 065147 122 145 141 T20SWP: .ASCIZ 'Read Data Parity Error (PARERR) Failed to Set after Write Wrong Parity'
9620 065256 122 145 141 T20RSF: .ASCIZ 'Read Data Parity Error (PARERR) Failed to Clear after RESET FIFO'
9621 065357 122 145 141 T20CMP: .ASCIZ 'Read Data Parity Error (PARERR) occurred in Data Loopback'
9622
9623
9624
9625
9626
9627 065452
9628 065452
9629 065456 012701 065752
9630 065462 012702 000120
9631 065466 105021
9632 065470 005302
9633 065472 003375
9634 065474 000207
9635
9636
9637
9638
9639 065476
9640 065476 004737 065452
9641 065502 012700 066110
9642 065506 112720 000005
9643 065512 105010
9644 065514 000207
9645
9646
9647
9648
9649
9650
9651

```

```

T20EXP:
    .WORD 0
    .WORD 0
    .WORD 0
    .WORD 0
    .WORD 0
    .WORD 0
    .WORD 0
    .WORD 0
T20EXSTA: .BLKB 64.
T20XEND:
;
; LOCAL TEXT MESSAGES FOR TEST
;
; BEGIN EXPECTED DATA BUFFER
; MESSAGE TYPE
; DATA FIELD LENGTH
; RBPGR
; XST0
; XST1
; XST2
; XST3
; XST4 (ALWAYS PRESENT FOR WRITE SUB.)
; EXPECTED READ STATUS AND WRITE FIFO DATA
; END EXPECTED DATA BUFFER
;
; CLEAR MESSAGE BUFFER
;
T20CLRBUF:
    SAVREG
    MOV     @T20BFR,R1
    MOV     @T20BEND-T20BFR,P2
104:      CLRB    (R1)+
    DEC     R2
    BGT     104
    RTS     PC
;
; SAVE R1-R5 UNTIL NEXT RETURN
; GET MESSAGE BUFFER ADDRESS
; SIZE OF MESSAGE BUFFER IN BYTES
; CLEAR A BYTE
; DONE?
; BR IF NO
; RETURN
;
; SETUP T20PK2 PACKET FOR READ STATUS
;
T20SRD:
    JSR     PC,T20CLRBUF
    MOV     @T20DT2,R0
    MOVB    @PW.RDSTATUS,(R0)+
    CLRB    (R0)
    RTS     PC
;
; CLEAR MESSAGE BUFFER
; WRITE SUBSYSTEM DATA BUFFER
; STORE READ STATUS COMMAND IN BSEL0
; CLEAR BSEL1
; RETURN
;
; SETUP T20PK2 PACKET FOR WRITE NPR
;
; INPUT:
; R0 CONTAINS BSEL1 NPR DATA

```

TEST 9: READ/WRITE DATA PARITY TEST

```

9652
9653
9654 065516
9655 065516 004737 065452
9656 065522 012701 066110
9657 065526 112721 000011
9658 065532 110011
9659 065534 000207
9660
9661
9662
9663
9664
9665
9666
9667 065536
9668 065536 004737 065452
9669 065542 012701 066110
9670 065546 112721 000003
9671 065552 110021
9672 065554 000207
9673
9674
9675
9676
9677
9678
9679
9680 065556
9681 065556
9682 065562 004737 065452
9683 065566 012702 066110
9684 065572 112722 000004
9685 065576 110022
9686 065600 005022
9687 065602 112122
9688 065604 005300
9689 065606 003375
9690 065610 000207
9691
9692
9693
9694
9695
9696
9697
9698 065612
9699 065612 004737 065452
9700 065616 012701 066110
9701 065622 112721 000007
9702 065626 110021
9703 065630 000207
9704
9705
9706
9707
9708

;
;--
T20WNPR:
    JSR    PC,T20CLRBUF    ;CLEAR MESSAGE BUFFER
    MOV    #T20DT2,R1     ;WRITE SUBSYSTEM DATA BUFFER
    MOVB   #PW.WNPR,(R1)+  ;STORE WRITE NPR IN BSEL0
    MOVB   R0,(R1)         ;STORE NPR DATA IN BSEL1
    RTS    PC              ;RETURN

;+
; SETUP T20PK2 PACKET FOR READ FIFO
; INPUT:
; R0 CONTAINS SEL2 BYTE COUNT
;--
T20RFIF:
    JSR    PC,T20CLRBUF    ;CLEAR MESSAGE BUFFER
    MOV    #T20DT2,R1     ;WRITE SUBSYSTEM DATA BUFFER
    MOVB   #PW.RFIFO,(R1)+ ;STORE READ FIFO IN BSEL0
    MOVB   R0,(R1)+        ;STORE BYTE COUNT IN BSEL1
    RTS    PC              ;RETURN

;+
; SETUP T20PK2 PACKET FOR WRITE FIFO
; INPUT:
; R0 CONTAINS BYTE COUNT
; R1 CONTAINS DATA PATTERN BLOCK ADDRESS
;--
T20WFIF:
    SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
    JSR    PC,T20CLRBUF    ;CLEAR MESSAGE BUFFER
    MOV    #T20DT2,R2     ;WRITE SUBSYSTEM DATA BUFFER
    MOVB   #PW.WFIFO,(R2)+ ;STORE WRITE FIFO IN BSEL0
    MOVB   R0,(R2)+        ;STORE BYTE COUNT IN BSEL1
    MOVB   (R2)+           ;CLEAR SEL2 (UNUSED)
    10$:   MOVB   (R1)+,(R2)+ ;STORE DATA PATTERN BYTE
    DEC    R0              ;DONE ALL BYTES?
    BGT    10$             ;BR IF NO
    RTS    PC              ;RETURN

;+
; SETUP T20PK2 FOR WRITE FORMAT TRANSPORT REGISTER
; INPUT:
; R0 CONTAINS DRIVING DATA PATTERN
;--
T20WFMT:
    JSR    PC,T20CLRBUF    ;CLEAR MESSAGE BUFFER
    MOV    #T20DT2,R1     ;WRITE SUBSYSTEM DATA BUFFER
    MOVB   #PW.WFMT,(R1)+ ;STORE WRITE FORMAT IN BSEL0
    MOVB   R0,(R1)+        ;STORE DATA WORD IN BSEL1
    RTS    PC              ;RETURN

;+
; SETUP T20PK2 PACKET FOR WRITE MISC.
; R0 CONTAINS WRITE MISC DATA
;--

```

TEST 9: READ/WRITE DATA PARITY TEST

```

9709 065632          T20WMISC:
9710 065632 012701 066110      MOV     #T20DT2,R1          ;WRITE SUBSYSTEM DATA BUFFER
9711 065636 112721 000010      MOVB    #PW.WMISC,(R1)+      ;STORE WRITE MISCELLANEOUS IN BSEL0
9712 065642 110011              MOVB    R0,(R1)             ;STORE INVERT EXTENDED FEATURES IN BSEL1
9713 065644 000207              RTS     PC                   ;RETURN
9714
9715      ;*
9716      ; SETUP T20PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
9717 065646          T20SEXT:
9718 065646 012700 066110      MOV     #T20DT2,R0          ;WRITE SUBSYSTEM DATA BUFFER
9719 065652 112720 000010      MOVB    #PW.WMISC,(R0)+      ;STORE WRITE MISCELLANEOUS IN BSEL0
9720 065656 112710 000200      MOVB    #MS.EXT,(R0)         ;STORE INVERT EXTENDED FEATURES IN BSEL1
9721 065662 000207              RTS     PC                   ;RETURN
9722
9723      ;*
9724      ; CLEAR EXPECTED DATA MESSAGE BUFFER
9725 065664          T20CLEXP:
9726 065664 012701 064412      MOV     #T20EXP,R1          ;GET EXPD ADDRESS
9727 065670 C12700 000120      MOV     #T20EXEND-T20EXP,R0    ;GET EXPD SIZE
9728 065674 105021 104:      CLRB    (R1)+                ;CLEAR A BYTE
9729 065676 005300              DEC     R0                  ;DONE?
9730 065700 003375              BGT     10$                 ;BR IF NO
9731 065702 000207              RTS     PC                   ;RETURN
9732
9733      ;*
9734      ;Set WORDS 0-7 of expd message BUFFER = to recv since not testing
9735      ;
9736 065704          T20SETEXP:
9737 065704 012702 064412      MOV     #T20EXP,R2          ;GET EXPD
9738 065710 012703 065752      MOV     #T20BFR,R3          ;GET READ STATUS RECV BUFFER
9739 065714 012700 000010      MOV     #8,R0               ;SET WORDS 0-7 EXP=RECV
9740 065720 012322 5$:      MOV     (R3)+,(R2)+            ;SET EXPD=RECV
9741 065722 005300              DEC     R0                  ;DONE WORDS 0-7 WORDS?
9742 065724 003375              BGT     5$                 ;BR IF NO
9743 065726 000207              RTS     PC                   ;RETURN
9744
9745      ;
9746      ;
9750      ;
9751      ;WRITE CHARACTERISTICS COMMAND PACKET
9752      ;
9753 065730          T20PACKET:
9754 065730 100004      .WORD    100004                  ;COMMAND PACKET FOR TEST
9755 065732 065740      .WORD    T20DATA                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
9756 065734 000000      .WORD    0                      ;ADDRESS OF CHARACTERISTICS BLOCK
9757 065736 000012      .WORD    10.                     ;MINIMUM MESSAGE PACKET SIZE
9758
9759 065740          T20DATA:
9760 065740 065752      .WORD    T20BFR                  ;CHARACTERISTICS DATA BLOCK
9761 065742 000000      .WORD    0                      ;ADDRESS OF MESSAGE BUFFER
9762 065744 000024      .WORD    20.                     ;LENGTH OF MESSAGE BUFFER
9763 065746 000000      .WORD    0                      ;ESS,ENB,EAI,ERI
9764 065750 000007      .WORD    7                      ;EXTENDED FEATURES UNIT NO.
9765
9766      ;MESSAGE BUFFER FOR ALL TEST 17 COMMANDS
9767
9768

```

TEST 9: READ/WRITE DATA PARITY TEST

```

9769 065752          T20BFR:          ;BEGIN MESSAGE BUFFER
9770 065752 000000    .WORD 0          ;MESSAGE TYPE
9771 065754 000000    .WORD 0          ;DATA FIELD LENGTH
9772 065756 000000    .WORD 0          ;RBPGR
9773 065760 000000    .WORD 0          ;XST0
9774 065762 000000    .WORD 0          ;XST1
9775 065764 000000    .WORD 0          ;XST2
9776 065766 000000    .WORD 0          ;XST3
9777 065770 000000    .WORD 0          ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
9778 065772          T20BFSTA: .BLKB 64. ;READ STATUS AND WRITE FIFO BUFFER
9779 066072          T20BEND:          ;END OF MESSAGE BUFFER
9780          ;
9781          ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
9782          ;
9784          .=<+.10>E177770
9786 066100          T20PK2:          ;WRITE SUBSYSTEM WITH ACK
9787 066100 100006    .WORD P.WRTSUB!P.ACK ;LOW ADDRESS OF DATA BLOCK
9788 066102 066110    .WORD T20DT2      ;HIGH ADDRESS OF DATA BLOCK
9789 066104 C00000    .WORD 0          ;MINIMUM MESSAGE PACKET SIZE
9790 066106 000012    .WORD 10.
9791          T20DT2:          ;DATA BLOCK
9792 066110          .BYTE 0          ;BSEL0
9793 066110          .BYTE 0          ;BSEL1
9794 066111          .WORD 0          ;SEL2
9795 066112 000000    .BLKB 64.        ;WRITE FIFO DATA OUTPUT BUFFER
9796 066114
9797
9798
9799 066214          ENDTST
          066214
          066214 104401
          L10073: TRAP C$ETST

          .SBTTL TEST 10: MANUAL INTERVENTION

;THE MANUAL INTERVENTION TEST IS A STANDALONE ROUTINE (NOT REALLY A "TEST")
;THAT ALLOWS THE OPERATOR TO CHECK OUT VARIOUS ELEMENTS AND FUNCTIONS OF
;THE SUBSYSTEM THAT CANNOT BE MANIPULATED BY THE PROGRAM ALONE. WHEN
;THIS ROUTINE IS STARTED, IT FIRST PRINTS OUT A MENU OF SELECTABLE
;SUBTESTS AND THEN WAITS FOR THE OPERATOR TO TYPE IN A SELECTION CODE.
;THE ONLY WAYS TO EXIT THIS ROUTINE AND RETURN TO THE DIAGNOSTIC SUPERVISOR
;ARE BY TYPING <CTRL-C> OR SELECTING CODE 7.
;SELECTION CODES AND SUBROUTINES ARE:
;
;      CODE  ROUTINE
;      0     HELP. PRINTS THIS MENU.
;      1     TURN ON ALL M7196 LED INDICATORS
;      2     TURN OFF ALL M7196 LED INDICATORS
;      3     OFFLINE/ONLINE ATTENTION TEST
;      4     WRITE-PROTECT TEST
;      5     INITIATE TRANSPORT SERVO EXERCISER
;      6     PRINT EXTENDED TRANSPORT STATUS
;      7     EXIT (RETURN TO SUPERVISOR)
;
;EACH MENU ITEM CORRESPONDS TO A SUBTEST. AS FOLLOWS:
;

```

TEST 10: MANUAL INTERVENTION

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

TEST 10: MANUAL INTERVENTION

```

9883      ;
9884      ;
9885      ;
9886      ;
9887 066216      BGNTST
          066216
9892 066216      RFLAGS R0      ;GET OPERATOR FLAGS      T10::
          066216 104421      ;BR, IF OK TO RUN      TRAP C#RFLA
          066220 001403      ;"TEST NOT EXECUTED"
9893 066222 012700 071570      BR 3#      ;JUMP IF NOT FIRST TEST
9894 066226 000402
9895 066230      21#:
9896 066230 012700 072705      MOV #T38ID,R0      ;TEST ID MESSAGE
9897 066234 004737 016510      JSR PC,TSTSETUP      ;DO THE COMMON SETUP
9898 066240 004737 017762      JSR PC,CHKMAN      ;IS MANUAL INTERVENTION ALLOWED?
9899 066244 103402      BCS 22#      ;BR, IF MANUAL INTER ALLOWED
9900 066246 000137 070772      JMP 64#      ;JUMP IF NOT ALLOWED
9901 066252      22#:
9902 066252 C05037 002222      CLR FATFLG      ;CLEAR THE FATAL ERROR FLAG
9903 066256 012737 176750 071004      MOV #65000.,T38DLY      ;SET UP DELAY COUNTER
9904 066264 004737 015774      JSR PC,SOFINIT      ;DO A SOFT INIT
9905 066270 103427      BCS 23#      ;BRANCH IF OK
9906 066272 010001      MOV R0,R1      ;CONTENTS OF TSSR REGISTER
9907 066274 032701 000200      BIT #SSR,R1      ;CHECK FOR TSSR SET
9908 066300 001023      BNE 23#      ;KEEP GOING IF NOT SET
9909 066302      DELAY 250      ;CALL DELAY ROUTINE
          066302 012727 000250      MOV #250,(PC)+
          066306 000000      .WORD 0
          066310 013727 002116      MOV L#DLY,(PC)+
          066314 000000      .WORD 0
          066316 005367 177772      DEC -6(PC)
          066322 001375      BNE -4
          066324 005367 177756      DEC -22(PC)
          066330 001367      BNE -20
9914 066332 005337 071004      DEC T38DLY      ;BUMP COUNTER DOWN
9915 066336 001352      BNE 5#      ;BR, IF MORE TIME LEFT
9916 066340      ERRDF ERRNO,SFIERR,SFIMSG      ;REPORT FATAL ERROR
          066340 104455      TRAP C#ERDF
          066342 001751      .WORD 1001
          066344 003652      .WORD SFIERR
          066346 012034      .WORD SFIMSG
9917 066350 012700 072732      23#: MOV #MIMENU,R0      ;MENU OF MANUAL INTERVENTIONS
9918 066354 012701 000005      MOV #5,R1      ;MAXIMUM ALLOWED SELECTION
9919 066360 004737 017540      JSR PC,GETSEL      ;GO GET THE OPERATORS SELECTION
9920 066364 010004      MOV R0,R4      ;GET NUMBER FROM ROUTINE
9921 066366 006304      ASL R4      ;CONVERT TO WORD OFFSET
9922 066370 000174 066374      JMP #6#(R4)      ;JUMP TO PROPER LOOP
9923 066374 066252      6#: .WORD 2#      ;RETYPE THE MENU
9924 066376 066410      .WORD 10#      ; 1 TURN ON LED'S
9925 066400 066672      .WORD 15#      ; 2 TURN OFF LED'S
9926 066402 067124      .WORD 20#      ; 3 ONLINE ATTENTION
9927 066404 067560      .WORD 25#      ; 4 WRITE PROTECT
9928 066406 070514      .WORD 35#      ; 5 EXTENDED TRANSPORT STATUS
9929 066410      10#: PRINTF #T38MS2      ;TELL OPERATOR TO CNTRL-C FOR EXIT
          066410 012746 072601      MOV #T38MS2,-(SP)
          066414 012746 000001      MOV #1,-(SP)
          066420 010600      MOV SP,R0

```

Address	Offset	PC	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	
---------	--------	----	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

TEST 10: MANUAL INTERVENTION

```

9977 066646 000240      12:  NOP                      ;ALLOW OPERATOR TO TYPE +C
9978 066650 012702 001750      MOV      #1000.,R2          ;SET-UP INNER LOOP
9979 066654 110165 000000      14:  MOV      R1,TSD8(R5)      ;WRITE DATA TO TSD8
9980 066660 016500 000002      MOV      TSSR(R5),R0        ;READ TSSR
9981 066664 005302          DEC      R2                  ;REDUCE INNER COUNT
9982 066666 001372          BNE      14:                ;LOOP TILL EXPIRES
9983 066670 000766          BR       12:                ;LOOP UNTIL HALTED
9984
9985 066672          15:  PRINTF  #T38MS2                ;TYPE CNTL C TO EXIT
      066672 012746 072601          MOV      #T38MS2,-(SP)
      066676 012746 000001          MOV      #1,-(SP)
      066702 010600          MOV      SP,R0
      066704 104417          TRAP     C:PNTF
      066706 062706 000004          ADD      #4,SP
9986 066712 004737 015774          JSR      PC,SOFINIT        ;DO SOFT INIT OF CONTROLLER
9987 066716 103405          BCS      200:              ;BR IF SOFT INIT = OK
9991 066720 010001          MOV      R0,R1                ;SAVE CONTENTS OF TSSR
9992 066722          ERD8F  ERRNO,SFIERR,SFMSG          ;DEVICE FATAL ERROR DURING INIT
      066722 104455          TRAP     C:ERD8F
      066724 001755          .WORD    1005
      066726 003652          .WORD    SFIERR
      066730 012034          .WORD    SFMSG
9993 066732
9994 066732 013737 002202 071530 200:  MOV      UNITN,T38DSW      ;SET UNIT NUMBER
9995 066740 012704 071510      MOV      #T38PK2,R4        ;SUBROUTINE NEEDS PACKET ADDRESS
9996 066744 004737 010662      JSR      PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
9997 066750 103405          BCS      210:              ;BR, IF COMMAND ISSUED OK
10001 066752 010001          MOV      R0,R1                ;SAVE CONTENTS OF TSSR
10002 066754          ERHRD  ERRNO,WRTMSG,SFMSG          ;WRITE CHARACTERISTICS FAILED
      066754 104456          TRAP     C:ERHRD
      066756 001756          .WORD    1006
      066760 005056          .WORD    WRTMSG
      066762 012034          .WORD    SFMSG
10003
10004
10005
10006
10007
10008 066764
10009 066764 112737 000000 071021
10010 066772 112737 000025 071020
10011 067000 012704 071010
10012 067004 010465 000000
10013 067010 004737 016336
10014 067014 103405
10015 067016 010001
10019 067020
      067020 104455          TRAP     C:ERD8F
      067022 001757          .WORD    1007
      067024 072206          .WORD    T38SSR
      067026 012046          .WORD    PKTSSR
10020 067030          250:  CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
      067030 104406          TRAP     C:CLP1
10021 067032          SETPRI  #PRI07                    ;RAISE THE PRIORITY
      067032 012700 000340          MOV      #PRI07,R0
      067036 104441          TRAP     C:SPRI
10022 067040 005037 070776          CLR      TTION2        ;ASSUME INTERRUPTS ARE ENABLED

```

TEST 10: MANUAL INTERVENTION

10023	067044	032737	000100	177560	BIT	#100,0#TTICSR	;ARE TTI INTERRUPTS ON ?
10024	067052	001005			BNE	710#	;BRANCH IF YES
10025	067054	005237	070776		INC	TTION2	;FLAG SET IF INTERRUPTS OFF
10026	067060	052737	000100	177560	BIS	#100,0#TTICSR	;ENABLE INTERRUPTS
10027	067066	012701	000060		MOV	#TTIVEC,R1	;START OF TTI VECTORS
10028	067072	011137	071000	710#:	MOV	(R1),TVSAV2	;SAVE THE CURRENT TTI VECTOR
10029	067076	012721	070276		MOV	#590#,(R1)+	;SET NEW INTERRUPT ROUTINE
10030	067102	011137	071002		MOV	(R1),TPSAV2	;SAVE THE VECTOR PRIORITY
10031	067106	012711	000340		MOV	#PRI07,(R1)	;USE PRIORITY SEVEN
10032	067112				SETPRI	#PRI00	;LOWER INTERRUPT BR LEVEL
	067112	012700	000000				MOV #PRI00,R0
	067116	104441					TRAP C#SPRI
10033	067120	000240		260#:	NOP		;ALLOW CNTL C
10034	067122	000776			BR	260#	;LOOP UNTIL STOPPED
10035							
10036							
10037	067124			20#:	PRINTF	#T38MS2	;TELL'EM WHAT TO TYPE
	067124	012746	072601				MOV #T38MS2,-(SP)
	067130	C12746	000001				MOV #1,-(SP)
	067134	010600					MOV SP,R0
	067136	104417					TRAP C#PNTF
	067140	062706	000004				ADD #4,SP
10038	067144				SETPRI	#PRI00	;LOWER PRIORITY TO ALLOW INTERRUPTS
	067144	012700	000000				MOV #PRI00,R0
	067150	104441					TRAP C#SPRI
10039	067152	005037	002224		CLR	INTRECV	;CLEAR INTERRUPT RECEIVED FLAG
10040	067156	004737	015774		JSR	PC,SOFINIT	;DO SOFT INIT OF CONTROLLER
10041	067162	103405			BCS	300#	;BR IF SOFT INIT = OK
10045	067164	010001			MOV	R0,R1	;SAVE CONTENTS OF TSSR
10046	067166				ERRDF	ERRNO,SFIERR,SFIMSG	;DEVICE FATAL ERROR DURING INIT
	067166	104455					TRAP C#ERDF
	067170	001760					.WORD 1008
	067172	003652					.WORD SFIERR
	067174	012034					.WORD SFIMSG
10047	067176			300#:			
10048	067176	013737	002202	071530	MOV	UNITN,T38DSW	;SET UNIT NUMBER IN PACKET
10049	067204	012737	000040	071526	MOV	#BIT5,T38EAI	;ENABLE ATTENTION INTERRUPTS
10050	067212	012704	071510		MOV	#T38PK2,R4	;SUBROUTINE NEEDS PACKET ADDRESS
10051	067216	004737	010662		JSR	PC,WRTCHR	;ISSUE WRITE CHARACTERISTICS
10052	067222	103405			BCS	310#	;BR, IF COMMAND ISSUED OK
10056	067224	010001			MOV	R0,R1	;SAVE CONTENTS OF TSSR
10057	067226				ERRHRD	ERRNO,WRTMSG,SFIMSG	;WRITE CHARACTERISTICS FAILED
	067226	104456					TRAP C#ERHRD
	067230	001761					.WORD 1009
	067232	005056					.WORD WRTMSG
	067234	012034					.WORD SFIMSG
10058	067236			310#:			
10059	067236	012704	071540		MOV	#T38PK3,R4	;SET UP NEW PACKET FOR MESS BUF REL
10060	067242	010465	000000		MOV	R4,TSDB(R5)	;MESSAGE BUFFER RELEASE,ACK,CVC=1 CMD
10061	067246	004737	016250		JSR	PC,WAITF	;WAIT FOR SSR TO SET
10062	067252	005002			CLR	R2	;MAKE SURE ALL IS CLEAR
10063	067254	016501	000002		MOV	TSSR(R5),R1	;GET TSSR STATUS
10064	067260	032701	000100		BIT	#OFL,R1	;IS OFL SET
10065	067264	001402			BEQ	320#	;BR, IF OFL IS NOT SET
10066	067266	052702	000100		BIS	#OFL,R2	;SET OFL IN EXPECTED
10067	067272	052702	000200	320#:	BIS	#SSR,R2	;SET UP EXPECTED
10068	067276	020201			CMP	R2,R1	;IS EVERYTHING OK

Address	Offset	Hex	Dec	Label	Instruction	Comment	Register
10069	067300	001404			BEQ	350#	
10073	067302				ERRHRD	ERRNO,T38SST,PKTSSR	
	067302	104456					
	067304	001762					
	067306	072416					
	067310	012046					
10074	067312			350#:	CKLOOP		
	067312	104406					
10075	067314				PRINTF	#T38MS1	
	067314	012746	072506				
	067320	012746	000001				
	067324	010600					
	067326	104417					
	067330	062706	000004				
10076	067334				PRINTF	#T38MS2	
	067334	012746	072601				
	067340	012746	000001				
	067344	010600					
	067346	104417					
	067350	062706	000004				
10077	067354				SETPRI	#PRI07	
	067354	012700	000340				
	067360	104441					
10078	067362	005037	070776		CLR	TTION2	
10079	067366	032737	000100	177560	BIT	#100,#TTICSR	
10080	067374	001005			BNE	720#	
10081	067376	005237	070776		INC	TTION2	
10082	067402	052737	000100	177560	BIS	#100,#TTICSR	
10083	067410	012701	000060		MOV	#TTIVEC,R1	
10084	067414	011137	071000	720#:	MOV	(R1),TVSAV2	
10085	067420	012721	070276		MOV	#590#,(R1)+	
10086	067424	011137	071002		MOV	(R1),TPSAV2	
10087	067430	012711	000340		MOV	#PRI07,(R1)	
10088	067434				SETPRI	#PRI00	
	067434	012700	000000				
	067440	104441					
10089	067442	000240			NOP		
10090	067444	005737	002224	360#:	TST	INTRECV	
10091	067450	001001			BNE	370#	
10092	067452	000773			BR	360#	
10093	067454			370#:	PRINTF	#T38INT	
	067454	012746	072276				
	067460	012746	000001				
	067464	010600					
	067466	104417					
	067470	062706	000004				
10094	067474	016501	000002		MOV	TSSR(R5),R1	
10095	067500	032701	000100		BIT	#OFL,R1	
10096	067504	001011			BNE	380#	
10097	067506				PRINTF	#T38ONL	
	067506	012746	072326				
	067512	012746	000001				
	067516	010600					
	067520	104417					
	067522	062706	000004				
10098	067526	000410			BR	390#	
10099	067530			380#:	PRINTF	#T38OFL	

TEST 10: MANUAL INTERVENTION

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

TEST 10: MANUAL INTERVENTION

```

10138 067726 001402      BEQ      435$      ;BR, IF FLAG NOT SET
10139 067730 000137 066252  JMP      2$      ;RE-WRITE MENU
10140 067734 017737 113164 071562 435$: MOV      @FREE,T38WR      ;SET UP WRITE BUFFER ADDRESS
10141 067742 012704 071560      MOV      @T38PK4,R4      ;GET PACKET ADDRESS
10142 067746 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS
10143 067752 004737 016250      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
10144 067756 016501 000002      MOV      TSSR(R5),R1      ;GET TSSR
10145 067762 012702 100206      MOV      @SC!SSR!BIT1!BIT2,R2      ;SET UP EXPECTED
10146 067766 020102      CMP      R1,R2      ;ARE THEY EQUAL (CORRECT)
10147 067770 001404      BEQ      440$      ;BR, IF CORRECT STATUS
10151 067772      ERRHRD      ERRNO,T38WRT,PKTSSR      ;"TSSR INCORRECT AFTER WRITE COMMAND
                                TRAP      C$ERHRD
                                .WORD      1014
                                .WORD      T38WRT
                                .WORD      PKTSSR
                                TRAP      C$CLP1
10152 070002      440$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
10153 070004 015701 071034      MOV      T38FR+6,R1      ;READ XSTO CONTENTS
10154 070010 010102      MOV      R1,R2      ;SET UP EXPECTED
10155 070012 052702 004000      BIS      @BIT11,R2      ;SET THE WRITE LOCK ERROR BIT (XSTO)
10156 070016 020102      CMP      R1,R2      ;WAS THE BIT SET
10157 070020 001404      BEQ      450$      ;BR, IF IT WAS (GOOD)
10161 070022      ERRHRD      ERRNO,T38WLE,EXPREC      ;"WRITE LOCK ERROR BIT NOT SET"
                                TRAP      C$ERHRD
                                .WORD      1015
                                .WORD      T38WLE
                                .WORD      EXPREC
                                TRAP      C$CLP1
10162 070032      450$: CKLOOP      ;LOOP IF SELECTED
10163 070034 000137 066252      JMP      2$      ;GO BACK TO MENU
10164
10165      ;*****
10166      ;      SERVO EXERCISER NO LONGER USED
10167      ;*****
10168 070040      30$: PRINTB      @T38MS3      ;"EXE ANY OTHER MENU SELECTION TO STOP
10169 070040      MOV      @T38MS3,-(SP)
                                MOV      @1,-(SP)
                                MOV      SP,R0
                                TRAP      C$PNTB
                                ADD      @4,SP
10170 070060 004737 073342      JSR      PC,T38REST      ;SET PACKET TO INITIAL VALUES
10171 070064 004737 015774      JSR      PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
10172 070070 103405      BCS      500$      ;BR IF SOFT INIT = OK
10176 070072 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
10177 070074      ERDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL ERROR DURING INIT
                                TRAP      C$ERDF
                                .WORD      1016
                                .WORD      SFIERR
                                .WORD      SFIMSG
10178 070104 013737 002202 071530 500$: MOV      UNITN,T38DSW      ;SET UNIT NUMBER
10179 070112 012704 071510      MOV      @T38PK2,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
10180 070116 004737 010662      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
10181 070122 103405      BCS      510$      ;BR, IF COMMAND ISSUED OK
10185 070124 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
10186 070126      ERRHRD      ERRNO,WRTMSG,SFIMSG      ;WRITE CHARACTERISTICS FAILED
                                TRAP      C$ERHRD
                                .WORD      1016
                                .WORD      WRTMSG
                                .WORD      SFIMSG

```

Address	Hex	Dec	Label	Operation	Comments
070130	001771				
070132	005056				
070134	012034				
10187	070136		5104:		
10188	070136	112737	000000	071021	MOV B #0, T38BS1 ; CLEAR BIT #4
10189	070144	112737	000020	071020	MOV B #20, T38BS0 ; EXECUTE DRIVE TEST 22
10190	070152	012704	071010		MOV #T38PACKET, R4 ; SET UP NEW WRT. SUBSYS MEM. COMMAND
10191	070156	010465	000000		MOV R4, TSD8(R5) ; SET THE PACKET ADDRESS
10192	070162	004737	016336		JSR PC, CHKTSSR ; WAIT FOR SSR TO SET
10193	070166	103405			BCS 5504 ; BR IF CARRY SET (GOOD RETURN)
10194	070170	010001			MOV R0, R1 ; SAVE CONTENTS OF TSSR
10198	070172				ERRDF ERRNO, T38SSR, PKTSSR ; DEVICE FATAL SSR FAILED TO SET
	070172	104455			
	070174	001772			
	070176	072206			
	070200	012046			
10199	070202		5504:	CKLOOP	; LOOP ON ERROR, IF FLAG SET
	070202	104406			
10200	070204			SETPRI #PRI07	; RAISE THE PRIORITY
	070204	012700	000340		
	070210	104441			
10201	070212	005037	070776		
10202	070216	032737	000100	177560	CLR TTION2 ; ASSUME INTERRUPTS ARE ENABLED
10203	070224	001005			BIT #100, #TTICSR ; ARE TTI INTERRUPTS ON ?
10204	070226	005237	070776		BNE 5554 ; BRANCH IF YES
10205	070232	052737	000100	177560	INC TTION2 ; FLAG SET IF INTERRUPTS OFF
10206	070240	012701	000060	5554:	BIS #100, #TTICSR ; ENABLE INTERRUPTS
10207	070244	011137	071000		MOV #TTIVEC, R1 ; START OF TTI VECTORS
10208	070250	012721	070276		MOV (R1), TVSAV2 ; SAVE THE CURRENT TTI VECTOR
10209	070254	011137	071002		MOV #5904, (R1) ; SET NEW INTERRUPT ROUTINE
10210	070260	012711	000340		MOV (R1), TPSAV2 ; SAVE THE VECTOR PRIORITY
10211	070264				MOV #PRI07, (R1) ; USE PRIORITY SEVEN
	070264	012700	000000		SETPRI #PRI00 ; LOWER INTERRUPT BR LEVEL
	070270	104441			
10212	070272	000240		5604:	NOP ; LOOP AWHILE
10213	070274	000776			BR 5604 ; STAY IN "TIGHT" LOOP
10214					
10215					
10216					
10217					
10218	070276	010046		5904:	MOV R0, -(SP) ; SAVE WORK REGISTER
10219	070300	113700	177562		MOV B #TTIBFR, R0 ; GET THE OPERATOR INPUT
10220	070304	042700	000200		BIC #200, R0 ; STRIP OFF PARITY BIT
10221	070310	122700	000015		CMPI #15, R0 ; IS IT A CARRIAGE RETURN ?
10222	070314	001075			BNE 5914 ; JUST EXIT IF NOT
10223	070316	012766	066252	000002	MOV #24, 2(SP) ; RETURN TO MASTER MENU
10224	070324	005066	000004		CLR 4(SP) ; FORCE PRIORITY 0
10225	070330	013737	011000	000060	MOV TVSAV2, #TTIVEC ; RESTORE VECTOR
10226	070336	013737	071002	000062	MOV TPSAV2, #TTIVEC+2 ; RESTORE SUPER PRIORITY
10227	070344	112737	000625	071020	MOV B #25, T38BS0 ; STOP DRIVE TEST 22
10228	070352	112737	000040	071021	MOV B #0, T38BS1 ; CLEAR BS1
10229	070360	012704	071010		MOV #T38PACKET, R4 ; SET UP NEW WRT. SUBSYS MEM. COMMAND
10230	070364	010465	000000		MOV R4, TSD8(R5) ; SET THE PACKET ADDRESS
10231	070370	012737	176750	071004	MOV #65000, T38DLY ; SET UP DELAY COUNTER
10232	070376	004737	016250	5924:	JSR PC, WAITF ; DO A WAIT FOR SSR
10233	070402	016501	000002		MOV TSSR(R5), R1 ; CONTENTS OF TSSR REGISTER
10234	070406	032701	000200		BIT #SSR, R1 ; CHECK FOR T

[illegible]

TEST 10: MANUAL INTERVENTION

Address	Offset	Hex	Dec	Label	Operation	Comments
070616	104406					
10279	070620	112737	000000	071021	MOV B	#0, T38BS1
10280	070626	112737	000024	071020	MOV B	#24, T38BS0
10281	070634	012704	071010		MOV	#T38PACKET, R4
10282	070640	010465	000000		MOV	R4, TSD8(R5)
10283	070644	012737	000144	071004	MOV	#100, T38DLY
10284	070652	004737	016250	620\$:	JSR	PC, WAITF
10285	070656	016501	000002		MOV	TSSR(R5), R1
10286	070662	032701	000200		BIT	#SSR, R1
10287	070666	001017			BNE	630\$
10288	070670				DELAY	250
	070670	012727	000250			
	070674	000000				
	070676	013727	002116			
	070702	000000				
	070704	005367	177772			
	070710	001375				
	070712	005367	177756			
	070716	001367				
10289	070720	005337	071004		DEC	T38DLY
10290	070724	001352			BNE	620\$
10291	070726	004737	016336	630\$:	JSR	PC, CHKTSSR
10292	070732	103405			BCS	650\$
10293	070734	010001			MOV	R0, R1
10297	070736				ERRDF	ERRNO, T38SSR, PKTSSR
	070736	104455				
	070740	001776				
	070742	072206				
	070744	012046				
10298	070746			650\$:	CKLOOP	
	070746	104406				
10299	070750	012700	071046		MOV	#T38BFR+20, R0
10300	070754	005001			CLR	R1
10301	070756	005037	003134		CLR	KTENABLE
10302	070762	004737	073400		JSR	PC, T38MBP
10303	070766	000137	066252		JMP	2\$
10304						
10305						
10306				63\$:	JMP	200
10307	070772			64\$:	EXIT	TST
	070772	104432				
	070774	003056				
10308						
10309						
10310						
10311						
10312						
10313						
10314						
10315						
10316						
10317						
10318						
10319	070776	000000			TTION2:	.WORD 0
10320	071000	000000			TVSAV2:	.WORD 0
10321	071002	000000			TPSAV2:	.WORD 0
10322						

TEST 10: MANUAL INTERVENTION

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

TEST 10: MANUAL INTERVENTION

```

10388 071645      045      116      045 T38MS3: .ASCIZ 'MNA Type <RETURN> To Stop Servo Exerciser, Return To Menu'
10389 071740      124      123      123 T38WRT: .ASCIZ 'TSSR Not Correct After WRITE, With WRITE PROTECT On'
10390 072024      127      122      111 T38WRL: .ASCIZ 'WRITE LOCKED Bit Not Set In XST0'
10391 072065      127      122      111 T38WLE: .ASCIZ 'WRITE LOCK ERROR Bit Not Set In XST0'
10392 072132      127      122      111 T38NBA: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted'
10393 072206      103      157      156 T38SSR: .ASCIZ 'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
10394 072276      045      116      045 T38INT: .ASCIZ 'MNA Interrupt Received'
10395 072326      045      116      045 T38ONL: .ASCIZ 'MNA Drive Is Now ON-LINE'
10396 072362      045      116      045 T38OFL: .ASCIZ 'MNA Drive Is Now OFF-LINE'
10397 072416      103      157      156 T38SST: .ASCIZ 'Contents of TSSR Incorrect After MESSAGE BUFFER RELEASE'
10398 072506      045      116      045 T38MS1: .ASCIZ 'MNA Toggle ON-LINE Switch to Generate ATTENTION Interrupts'
10399 072601      045      116      045 T38MS2: .ASCIZ 'MNA Type RETURN To Return To Menu'
10400 072645      111      163      040 T38MSG: .ASCIZ 'Is Write-Protected Tape Mounted'
10401 072705      115      141      156 T38ID: .ASCIZ 'Manual Intervention'
10402                                     .EVEN
10403 072732      072756      073030      073056 MIMENU: .WORD 1,2,3,4,5,6
10404 072746      073225      073270      073337      .WORD 8,9,10,0
10405
10406 072756      012      123      105 1: .ASCIZ '<12> SELECT OPERATION FROM FOLLOWING OPTIONS:'
10407 073030      012      011      060 2: .ASCIZ '<12>' 0 Display This Menu'
10408 073056      011      061      011 3: .ASCIZ ' 1 Turn On All M7196 LED's'
10409 073110      011      062      011 4: .ASCIZ ' 2 Turn Off All M7196 LED's'
10410 073143      011      063      011 5: .ASCIZ ' 3 Offline/Online Attention'
10411 073177      011      064      011 6: .ASCIZ ' 4 Write Protect Test'
10412 073225      011      065      011 8: .ASCIZ ' 5 Print Extended Transport Status'
10413 073270      011      136      103 9: .ASCIZ ' 9C To Return to Diagnostic Supervisor'
10414 073337      000      104: .ASCIZ ''
10415                                     .EVEN
10416
10417 ;*
10418 ;LOCAL STORAGE FOR THIS TEST
10419 ;*
10420
10421 073340      000000      T38DAT: .WORD 0 ;LOGICAL RESPONSE TO QUESTION
10422 073342      T38REST:
10423 073342      SAVREG
10424 073346      012701      071010      MOV #T38PACKET,R1 ;SAVE THE REGISTERS
10425 073352      012721      140206      MOV #140206,(R1)+ ;START OF THE PACKET
10426 073356      012721      071020      MOV #T38TAD,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
10427 073362      005021      CLR (R1)+ ;ADDRESS OF DATA BLOCK
10428 073364      012721      000006      MOV #6,(R1)+ ;EXTENDED ADDRESS
10429 073370      005021      CLR (R1)+ ;SIZE OF DATA BLOCK IN BYTES
10430 073372      005021      CLR (R1)+ ;CLEAR BSEL0 AND BSEL1
10431 073374      005011      CLR (R1)+ ;CLEAR SEL2
10432 073376      000207      RTS PC ;CLEAR DATA AREA
10433                                     ;RETURN
10434
10435 ;*
10436 ;
10437 ;THIS ROUTINE PRINTS THE CONTENTS OF
10438 ;THE 256 BYTE MESSAGE BUFFER RETURNED BY THE
10439 ;TSV-05.
10440 ;
10441 ;INPUT:
10442 ;
10443 ; R0 LOW ORDER ADDRESS OF MESSAGE BUFFER
10444 ; R1 HIGH ORDER ADDRESS OF MESSAGE BUFFER

```

TEST 10: MANUAL INTERVENTION

```

10445      ;      NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
10446      ;
10447      ;
10448      ; -
10449
10450 073400 T38MBP:
10451 073400 SAVREG
10452 073404 010005 MOV R0,R5 ;SAVE THE REGISTERS
10453 073406 005737 003134 TST KTENABLE ;SAVE LOW ORDER ADDRESS
10454 073412 001001 BNE 910$ ;ADDRESS ABOVE 28K?
10455 073414 005001 CLR R1 ;BR IF YES
10456 073416 010103 910$: MOV R1,R3 ;SET HIGH ORDER ADDRESS TO 0
10457 073420 006100 ROL R0 ;SAVE HIGH ORDER ADDRESS
10458 073422 006101 ROL R1 ;SHIFT BIT15 TO C BIT
10459 073424 PRINTX #T38AS0,R1,R5 ;SHIFT TO HIGH ORDER FOR PRINTOUT
                                ;PRINT MESSAGE BUFFER ADDRESS
                                MOV R5,-(SP)
                                MOV R1,-(SP)
                                MOV #T38AS0,-(SP)
                                MOV #3,-(SP)
                                MOV SP,R0
                                TRAP C#PNTX
                                ADD #10,SP
10460 073450 PRINTX #T38AS1 ;PRINT HEADER FOR CONTENTS
                                MOV #T38AS1,-(SP)
                                MOV #1,-(SP)
                                MOV SP,R0
                                TRAP C#PNTX
                                ADD #4,SP
10461 073470 MOV R5,R1 ;COPY LOW ORDER ADDRESS
10462 073472 MOV R3,R0 ;COPY HIGH ORDER ADDRESS
10463 073474 001403 BEQ 913$ ;BR IF NOT ABOVE 28K
10464 073476 004737 017316 JSR PC,SETMAP ;SETUP PAR ADDRESS IN R0
10465 073502 010005 MOV R0,R5 ;GET PAR FORMAT ADDRESS ABOVE 28K
10466 073504 010537 074050 913$: MOV R5,T38CNT ;HOLD ADDRESS
10467 073510 011504 911$: MOV (R5),R4 ;GET BUFFER ENTRY
10468 073512 022704 125252 CMP #125252,R4 ;CHECK FOR NO LOAD CONDITION
10469 073516 001417 BEQ 912$ ;BR, IF BUFFER WASN'T LOADED
10470 073520 010403 MOV R4,R3 ;MAKE COPY
10471 073522 042704 170377 BIC #170377,R4 ;ONLY BITS 11,10,9 AND 8 ARE SAVED
10472 073526 000241 CLC ;CLEAR CARRY
10473 073530 006004 ROR R4 ;11 TO 10 BIT POSITION
10474 073532 006004 ROR R4 ;10 TO 9 BIT POSITION
10475 073534 006004 ROR R4 ;9 TO 8 BIT POSITON
10476 073536 006004 ROR R4 ;8 TO 7 BIT POSITION
10477 073540 042703 177760 BIC #177760,R3 ;ONLY BITS 3,2,1 AND 0 ARE SAVED
10478 073544 060403 ADD R4,R3 ;"OR'EM TOGETHER
10479 073546 010325 MOV R3,(R5)+ ;PUT BACK IN BUFFER
10480 073550 020527 0/1502 CMP R5,#T38EB ;END OF BUFFER YET
10481 073554 001355 BNE 911$ ;BR, IF NOT AT END YET
10482 073556 013705 074050 912$: MOV T38CNT,R5 ;PUT ADDRESS BACK
10483 073562 012704 000001 MOV #1,R4 ;START BYTE NUMBER AT ONE
10484 073566 915$: PRINTX #T38ASN,R4,(R5)+ ;PRT MEM BUFFER W/NEWLINE
                                MOV (R5)+,-(SP)
                                MOV R4,-(SP)
                                MOV #T38ASN,-(SP)
                                MOV #3,-(SP)
                                MOV SP,R0
10485 073566 012546
10486 073570 010446
10487 073572 012746 074024
10488 073576 012746 000003
10489 073602 010600

```

TEST 10: MANUAL INTERVENTION

```

10485 073604 104415 000010
10486 073606 062706 074050
10487 073612 005037 074050
10488 073616 000412
10489 073620 012546
10490 073622 010446
10491 073624 012746 074005
10492 073630 012746 000003
10493 073634 010600
10494 073636 104415
10495 073640 062706 000010
10496 073644 005237 074050
10497 073650 005204
10498 073652 020427 000200
10499 073656 003010
10500 073660 023727 074050 000004
10501 073666 001401
10502 073670 000753
10503 073672 005037 074050
10504 073676 000733
10505 073700 000207
10506 073702 045 116 045 T38AS0: .ASCIZ 'MNA Message Buffer Address = #01#05'
10507 073747 045 116 045 T38AS1: .ASCIZ 'MNA Message Buffer Contents.'
10508 074005 045 101 040 T38ASC: .ASCIZ 'A #D4#A: #03'
10509 074024 045 116 045 T38ASN: .ASCIZ 'MNA Byte#D4#A: #03'
10510 .EVEN
10511 T38CNT: .WORD ;COUNTER FOR PRINT
10512 .ENDST
10513
10514
10515
10516
10517
10518
10519
10520
10521
10522
10523
10524
10525
10526
10527
10528
10529
10530

```

CLR T38CNT ;CLEAR COUNTER
BR 921# ;SKIP OTHER PRINT
PRINTX #T38ASC,R4,(R5)+ ;PRINT THE CONTENTS OF MEMORY BUFFER

MOV (R5)+,-(SP)
MOV R4, -(SP)
MOV #T38ASC, -(SP)
MOV #3, -(SP)
MOV SP, R0
TRAP C#PNTX
ADD #10, SP

921# : INC T38CNT ;BUMP COUNTER
INC R4 ;NUMBER OF THE NEXT
CMP R4, #128. ;DONE ALL YET ?
BGT 50# ;BRANCH IF ALL DONE
CMP T38CNT, #4 ;DONE FOUR YET
BEQ 925# ;BR, IF THREE DONE
BR 920# ;KEEP GOING
925# : CLR T38CNT ;CLEAR COUNTER
BR 915# ;PRINT WITH NEW LINE
50# : RTS PC ;RETURN

L10075: TRAP C#ETST

.SBTTL TEST 11: CONFIGURATION TYPEOUT

```

;THIS IS A STANDALONE ROUTINE THAT PRINTS OUT ON THE CONSOLE TERMINAL
;THE CONFIGURATION OF THE M7196 MODULE AND TSV05 SUBSYSTEM. SPECIFICALLY,
;THE FOLLOWING INFORMATION IS PRESENTED:
;
; 1.0 STATE OF THE EXTENDED FEATURES SWITCH ON THE M7196: ON (EXTENDED
;     FEATURES ENABLED) OR OFF (EXTENDED FEATURES DISABLED).
;
; 2.0 STATE OF THE BUFFERING ENABLE SWITCH ON THE M7196: ON
;     (BUFFERING ENABLED) OR OFF (BUFFERING DISABLED).
;
; 3.0 MICROCODE REVSION LEVEL OF THE M7196.
;
; 4.0 NUMBER OF TAPE TRANSPORTS CONNECTED TO THE CONTROLLER.
;
; 5.0 UNIT SELECT CODE AND STATE (ONLINE/OFFLINE, WRITE ENABLED/PROTECTED)
;     OF EACH CONNECTED TRANSPORT. IN ADDITION, THE PROGRAM WILL INDICATE,
;     FOR EACH ON-LINE TRANSPORT, WHETHER OR NOT IT IS EQUIPPED WITH THE
;     EXTENDED TAPE STATUS READOUT FEATURE.
;
;THE OPERATOR IS EXPECTED TO READ THE PRINTOUT AND VERIFY THAT IT MATCHES
;THE ACTUAL CONFIGURATION AT HAND. IF, FOR EXAMPLE, THE PROGRAM INDICATES

```

TEST 11: CONFIGURATION TYPEOUT

```

10531 ; THAT IT "SEES" TWO TRANSPORTS CONNECTED WHEN IN FACT ONLY ONE IS PRESENT.
10532 ; THE OPERATOR MUST INTERPRET THIS AS AN ERROR AND ATTEMPT TO FIND THE
10533 ; CAUSE (BAD CABLE, FAULTY UNIT-SELECT DECODING IN THE TRANSPORT, ETC.).
10534 ; [SINCE THE CONTROLLER CAN ONLY ACCESS UNIT 0 IF IT IS IN "STANDARD"
10535 ; MODE, THE PROGRAM WILL FORCE THE MODULE INTO EXTENDED MODE VIA THE
10536 ; WRITE SUBSYSTEM MEMORY COMMAND IN ORDER TO SCAN FOR CONNECTED TRANSPORTS.]
10537 ;
10538 ;
10539 ; THIS ROUTINE, WHEN ITS ACTIONS ARE COMPLETED, WILL EXIT BACK TO THE
10540 ; DIAGNOSTIC SUPERVISOR SO THAT IF ADDITIONAL UNITS (CONTROLLERS) ARE
10541 ; SELECTED (E.G., FROM THE INITIAL STARTUP DIALOG), THE ROUTINE WILL BE
10542 ; REENTERED SO THAT THEIR CONFIGURATIONS CAN BE PRINTED.
10543 ;
10544 BGNTST
10545 074054
10546 074054
10547 074054
10548 074054 104421
10549 074054 001403
10550 074056 C12700 076333
10551 074060 000402
10552 074066 012700 077534
10553 074072 004737 016510
10554 074076 004737 017762
10555 074102 103402
10556 074104 000137 075336
10557 074110
10558 074110 004737 015774
10559 074114 103405
10560 074116 010001
10561 074120
10562 074120 104455
10563 074122 002115
10564 074124 003652
10565 074126 012034
10566 074130
10567 074130 104406
10568 074132 013737 002202 076300
10569 074140 012704 076260
10570 074144 004737 010662
10571 074150 103405
10572 074152 010001
10573 074154
10574 074154 104456
10575 074156 002116
10576 074160 005056
10577 074162 012034
10578 074164
10579 074164 104406
10580 074166 013701 075610
10581 074172 012746 077252
10582 074176 012746 000001
10583 074202 010600
10584 074204 104415
10585 074206 062706 000004
10586 074212 032701 000200
10587 074216 001011

632: RFLAGS R0 ; GET OPERATOR FLAGS
      BEQ 10$ ; BR, IF OK TO RUN
      MOV #T39NE,R0 ; "TEST NOT EXECUTED"
      BR 11$ ; JUMP OUT OF TEST IF NOT
10$: MOV #TST39ID,R0 ; TEST ID MESSAGE
11$: JSR PC,TSTSETUP ; DO THE COMMON SETUP
      JSR PC,CHKMAN ; IS MANUAL INTERVENTION ALLOWED?
      BCS 20$ ; BR, IF MANUAL INTERVENTION ALLOWED
      JMP 64$ ; JUMP TO OUT IF NOT

20$: JSR PC,SOFINIT ; DO SOFT INIT OF CONTROLLER
      BCS 25$ ; BR IF SOFT INIT = OK
      MOV R0,R1 ; SAVE CONTENTS OF TSSR
      ERRDF ERRNO,SFIERR,SFIMSG ; DEVICE FATAL ERROR DURING INIT
      TRAP C$ERDF
      .WORD 1101
      .WORD SFIERR
      .WORD SFIMSG

25$: CKLOOP ; LOOP IF SELECTED
      TRAP C$CLP1
      MOV UNITN,T39DSW ; SET UNIT NUMBER
      MOV #T39PK2,R4 ; SUBROUTINE NEEDS PACKET ADDRESS
      JSR PC,WRTCHR ; ISSUE WRITE CHARACTERISTICS
      BCS 50$ ; BR, IF COMMAND ISSUED OK
      MOV R0,R1 ; SAVE CONTENTS OF TSSR
      ERRHRD ERRNO,WRTMSG,SFIMSG ; WRITE CHARACTERISTICS FAILED
      TRAP C$ERHRD
      .WORD 1102
      .WORD WRTMSG
      .WORD SFIMSG

50$: CKLOOP ; LOOP IF SELECTED
      TRAP C$CLP1
      MOV T39BFR+12,R1 ; GET XST2 STATUS FROM MESSAGE BUFFER
      PRINTX #T39SFS ; "STATE OF EXTENDED FEATURES SW ="
      MOV #T39SFS,-(SP)
      MOV #1,-(SP)
      MOV SP,R0
      TRAP C$PNTX
      ADD #4,SP

      BIT #BIT7,R1 ; CHECK STATE OF E.F.S.
      BNE 100$ ; BR, IF EXT. FEA. SW. IS ON

```

TEST 11: CONFIGURATION TYPEOUT

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

TEST 11: CONFIGURATION TYPEOUT

```

10608 074450 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
10609 074452          ERRHRD  ERRNO,WRTMSG,SFMSG    ;WRITE CHARACTERISTISC FAILED
          074452 104456                                TRAP    C#ERHRD
          074454 002120                                .WORD   1104
          074456 005056                                .WORD   WRTMSG
          074460 012034                                .WORD   SFMSG
10610 074462          150$: CKLOOP                ;LOOP IF SELECTED
          074462 104406                                TRAP    C#CLP1
10611 074464 005737 002226      TST    EXTFEA        ;CHECK FOR EXTENDED FEATURES SW SWITCH
10612 074470 001036          BNE    174$          ;BR IF SWITCH IS ON
10613 074472 112737 000200 075571  MOVB   #200,T39BS1    ;WRITE MISCELLANEOUS CONT/READ STATUS
10614 074500 112737 000010 075570  MOVB   #10,T39BS0    ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
10615 074506 012704 075560      MOV    #T39PACKET,R4    ;WRITE SUBSYS MEM PACKET
10616 074512 010465 000000      MOV    R4,TSDB(R5)      ;ISSUE COMMAND
10617 074516 004737 016336      JSR    PC,CHKTSSR      ;WAIT FOR SSR
10618 074522 103405          BCS    160$          ;BR, IF NO ERROR
10619 074524 010001          MOV    R0,R1          ;ERROR, SAVE TSSR
10623 074526          ERRHRD  ERRNO,T39NBA,PKTSSR    ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
          074526 104456                                TRAP    C#ERHRD
          074530 002121                                .WORD   1105
          074532 077035                                .WORD   T39NBA
          074534 012046                                .WORD   PKTSSR
10624 074536          160$: CKLOOP                ;LOOP IF SELECTED
          074536 104406                                TRAP    C#CLP1
10625 074540 012704 076260      MOV    #T39PK2,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
10626          ;*****
10627          ;
10628          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
10629          ;
10630          ;*****
10631
10632 074544 004737 010662      JSR    PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
10633 074550 103405          BCS    170$          ;BR, IF COMMAND ISSUED OK
10637 074552 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
10638 074554          ERRHRD  ERRNO,WRTMSG,SFMSG    ;WRITE CHARACTERISTISC FAILED
          074554 104456                                TRAP    C#ERHRD
          074556 002122                                .WORD   1106
          074560 005056                                .WORD   WRTMSG
          074562 012034                                .WORD   SFMSG
10639 074564          170$: CKLOOP                ;SCOPE LOOP
          074564 104406                                TRAP    C#CLP1
10640 074566 005037 002202 174$: CLR    UNITN        ;SET TO DRIVE 0
10641 074572 013737 002202 076300 175$: MOV    UNITN,T39DSW    ;SET UNIT NUMBER
10642 074600 012704 076260      MOV    #T39PK2,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
10643 074604 004737 010662      JSR    PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
10644 074610 103405          BCS    180$          ;BR, IF COMMAND ISSUED OK
10648 074612 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
10649 074614          ERRHRD  ERRNO,WRTMSG,SFMSG    ;WRITE CHARACTERISTISC FAILED
          074614 104456                                TRAP    C#ERHRD
          074616 002123                                .WORD   1107
          074620 005056                                .WORD   WRTMSG
          074622 012034                                .WORD   SFMSG
10650 074624          180$: CKLOOP                ;LOOP IF SELECTED
          074624 104406                                TRAP    C#CLP1
10651
10652 074626 016501 000002 190$: MOV    TSSR(R5),R1    ;GET TSSR STATUS
10653 074632 032701 000100      BIT    #OFL,R1        ;CHECK FOR OFF-LINE

```

TEST 11: CONFIGURATION TYPEOUT

10654	074636	001414		BEQ	200:				;BR, IF DRIVE IS ON-LINE
10655	074640			PRINTX		#T390F2,UNITN			;"DRIVE NUMBER YX IS OFF-LINE"
	074640	013746	002202					MOV	UNITN,-(SP)
	074644	012746	076574					MOV	#T390F2,-(SP)
	074650	012746	000002					MOV	#2,-(SP)
	074654	010600						MOV	SP,R0
	074656	104415						TRAP	C#PNTX
	074660	062706	000006					ADD	#6,SP
10656	074664	000137	075220		JMP	250:			;DO NOT TRY TO GET ANYMORE INFO.
10657	074670			200:	PRINTX	#T390N2,UNITN			;"DRIVE NUMBER XX IS ON-LINE"
	074670	013746	002202					MOV	UNITN,-(SP)
	074674	012746	076640					MOV	#T390N2,-(SP)
	074700	012746	000002					MOV	#2,-(SP)
	074704	010600						MOV	SP,R0
	074706	104415						TRAP	C#PNTX
	074710	062706	000006					ADD	#6,SP
10658	074714	013701	075604		MOV	T398FR+6,R1			;READ EXTENDED STATUS (XST0)
10659	074720	032701	000004		BIT	#BIT2,R1			;IS DRIVE WRITE PROTECTED
10660	074724	C01013			BNE	210:			;BR, IF WRITE PROTECTED
10661	074726				PRINTX	#T39WPN,UNITN			;"DRIVE NUMBER IS NOT WRT PRO"
	074726	013746	002202					MOV	UNITN,-(SP)
	074732	012746	076756					MOV	#T39WPN,-(SP)
	074736	012746	000002					MOV	#2,-(SP)
	074742	010600						MOV	SP,R0
	074744	104415						TRAP	C#PNTX
	074746	062706	000006					ADD	#6,SP
10662	074752	000412			BR	220:			;SKIP OVER
10663	074754			210:	PRINTX	#T39WRT,UNITN			;"DRIVE NUMBER XX IS WRT PRO"
	074754	013746	002202					MOV	UNITN,-(SP)
	074760	012746	076703					MOV	#T39WRT,-(SP)
	074764	012746	000002					MOV	#2,-(SP)
	074770	010600						MOV	SP,R0
	074772	104415						TRAP	C#PNTX
	074774	062706	000006					ADD	#6,SP
10664	075000	012737	125252	075676	220:	MOV	#125252,T398FR+100		;SET 1 LOC TO KNOWN VALUE
10665	075006	112737	000000	075571		MOVB	#0,T398S1		;EXTENDED TAPE STATUS
10666	075014	112737	000024	075570		MOVB	#24,T398S0		;EXTENDED TAPE STATUS
10667	075022	012704	075560			MOV	#T39PACKET,R4		;WRITE SUBSYS MEM PACKET
10668	075026	010465	000000			MOV	R4,TSDB(R5)		;ISSUE COMMAND
10669	075032	012737	000144	075552		MOV	#100,T39DLY		;SET UP DELAY ROUTINE
10670	075040	004737	016250		222:	JSR	PC,WAITF		;WAIT AMHIL FOR SSR TO SET
10671	075044	016501	000002			MOV	TSSR(R5),R1		;SEE IF IT REALLY DID
10672	075050	032701	000200			BIT	#SSR,R1		;JUST CHECK THAT BIT
10673	075054	001017				BNE	225:		;BR, IF SSR IS SET
10674	075056				DELAY	250			;DELAY ABOUT .25 SEC
	075056	012727	000250					MOV	#250,(PC)+
	075062	000000						.WORD	0
	075064	013727	002116					MOV	L#DLY,(PC)+
	075070	000000						.WORD	0
	075072	005367	177772					DEC	-6(PC)
	075076	001375						BNE	.-4
	075100	005367	177756					DEC	-22(PC)
	075104	001367						BNE	.-20
10675	075106	005337	075552		DEC	T39DLY			;START DELAY COUNT DOWN
10676	075112	001352			BNE	222:			;BR, IF COUNTER IS NOT AT DONE
10677	075114	004737	016336		JSR	PC,CHKTSSR			;WAIT FOR SSR
10678	075120	103405			BCS	230:			;BR, IF NO ERROR

Address	Offset	Hex	Dec	Label	Instruction	Comment	Register	Value
10679	075122	010001			MOV	RO,R1		
10683	075124				ERRHRD	ERRNO,T39NBA,PKTSSR		
	075124	104456						
	075126	002124						
	075130	077035						
	075132	012046						
10684	075134			230:	CKLOOP			
	075134	104406						
10685	075136	023727	075676	125252	CMP	T39BFR+100,#125252		
10686	075144	001013			BNE	240:		
10687	075146				PRINTX	#T39ETN,UNITN		
	075146	013746	002202					
	075152	012746	076501					
	075156	012746	000002					
	075162	010600						
	075164	104415						
	075166	062706	000006					
10688	075172	000412			BR	250:		
10689	075174			240:	PRINTX	#T39ETS,UNITN		
	075174	013746	002202					
	075200	012746	076412					
	075204	012746	000002					
	075210	010600						
	075212	104415						
	075214	062706	000006					
10690	075220	005237	002202		INC	UNITN		
10691	075224	023727	002202	000003	CMP	UNITN,#3		
10692	075232	001402			BEQ	631:		
10693	075234	000137	074572		JMP	175:		
10694	075240	012700	075342		MOV	#NUMENU,R0		
10695	075244	012701	000001		MOV	#1,R1		
10696	075250	004737	017540		JSR	PC,GETSEL		
10697	075254	010004			MOV	R0,R4		
10698	075256	006304			ASL	R4		
10699	075260	000174	075264		JMP	86:(R4)		
10700	075264	075240		6:	.WORD	631:		
10701	075266	074054			.WORD	632:		
10702	075270			63:	SETPRI	#PRI00		
	075270	012700	000000					
	075274	104441						
10703	075276				printf	#T38MS2		
	075276	012746	072601					
	075302	012746	000001					
	075306	010600						
	075310	104417						
	075312	062706	000004					
10704	075316				PRINTX	#T39NFL		
	075316	012746	076330					
	075322	012746	000001					
	075326	010600						
	075330	104415						
	075332	062706	000004					
10705	075336			64:	EXIT	TST		
	075336	104432				</		

TEST 11: CONFIGURATION TYPEOUT

```

10708      ;
10709
10710      ;LOCAL STORAGE FOR THIS TEST
10711      ;
10712 075342 075356 075430 075456 NUMENU: .WORD 1,2,3,4,5,0
10713 075356      012      123      105 1: .ASCIZ <12>'SELECT OPERATION FROM FOLLOWING OPTIONS:'
10714 075430      012      011      060 2: .ASCIZ <12>' 0 Display This Menu'
10715 075456      011      061      011 3: .ASCIZ ' 1 Rerun this test'
10716 075501      011      136      103 4: .ASCIZ ' tC To Return to Diagnostic Supervisor'
10717 075550      000      5: .ASCIZ ''
10718      .EVEN
10719
10720 075552 000000 T39DLY: .WORD 0 ;DELAY COUNTER FOR TEST
10722      075560      .=<+.10>E177770
10724 075560 T39PACKET: ;COMMAND PACKET FOR TEST
10725 075560 140006      .WORD 140006 ;WRITE SUBSYSTEM MEM. CMD, AC VC=1
10726 075562 075570      .WORD T39TAD ;ADDRESS OF CHARACTERISTICS BLOCK
10727 075564 000000      .WORD 0
10728 075566 C00012      .WORD 10. ;STARTING VALUE OF BLOCK SIZE
10729 075570 T39TAD: ;CHARACTERISTICS DATA BLOCK
10730 075570 000      T39BS0: .BYTE 0 ;BSEL0 BYTE
10731 075571 000      T39BS1: .BYTE 0 ;BSEL1 BYTE
10732 075572 000000 T39BS2: .WORD 0 ;BSEL1 WORD
10733 075574 000000      .WORD 0 ;DATA
10734 075576 T39BFR: .BLKW 150. ;MESSAGE BUFFER
10735
10736
10738      076260
10740 076260 T39PK2: .=<+.10>E177770 ;COMMAND PACKET FOR TEST
10741 076260 140004      .WORD 140004 ;WRITE CHARA. MEM. CMD., ACK,CVC=1
10742 076262 076270      .WORD T39DTA ;ADDRESS OF SELECT DATA BLOCK
10743 076264 000000      .WORD 0
10744 076266 000012      .WORD 10. ;STARTING VALUE OF BLOCK SIZE
10745
10746
10747 076270 T39DTA: ;SELECT DATA BLOCK
10748 076270 075576      .WORD T39BFR ;ADDRESS OF MESSAGE BUFFER
10749 076272 000000      .WORD 0
10750 076274 000400      .WORD 256. ;LENGTH OF MESSAGE BUFFER
10751 076276 000000 T39EAI: .WORD 0 ;EAI BIT WORD
10752 076300 000000 T39DSW: .WORD 0 ;DRIVE SELECT WORD ETC
10754      076310      .=<+.10>E177770
10756 076310 140012 T39PK3: .WORD 140012 ;MESSAGE BUFFER RELEASE COMMAND
10757 076312 000000      .WORD 0 ;NOT USED
10758
10759      ;WRITE TAPE PACKET
10760      ;
10762      076320
10764 076320 140005 T39PK4: .WORD 140005 ;WRITE, ACK, CVC=1 COMMAND
10765 076322 000000 T39WR: .WORD 0 ;ADDRESS OF WRITE BUFFER
10766 076324 000000      .WORD 0 ;MORE ADDRESS OF WRITE BUFFER
10767 076326 000400 T39SIZ: .WORD 256. ;SIZE OF RECORD
10768
10769
10770
10771
10772      ;

```

TEST 11: CONFIGURATION TYPEOUT

```

10773          ;LOCAL TEXT MESSAGES FOR TEST
10774          ;--
10775
10776
10777
10778
10779 076330      045      116      000  T39NFL: .ASCIZ  '#N'
10780 076333      123      164      141  T39NE: .ASCIZ  'Stand-alone Configuration Typeout Not Executed'
10781 076412      045      116      045  T39ETS: .ASCIZ  '#N#A Extended Tape Status Available, Drive Number #D2'
10782 076501      045      116      045  T39ETN: .ASCIZ  '#N#A Extended Tape Status NOT Available, Drive Number #D2'
10783 076574      045      116      045  T39OF2: .ASCIZ  '#N#A Drive Number #D2#A Is Off-Line'
10784 076640      045      116      045  T39ON2: .ASCIZ  '#N#A Drive Number #D2#A Is On-Line'
10785 076703      045      116      045  T39WRT: .ASCIZ  '#N#A Drive Number #D2#A Is Write Protected'
10786 076756      045      116      045  T39WPN: .ASCIZ  '#N#A Drive Number #D2#A Is NOT Write Protected'
10787 077035      127      122      111  T39NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
10788 077111      103      157      156  T39SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
10789 077201      045      116      045  T39OUT: .ASCIZ  '#N#A Type 'C to Return to the Supervisor'
10790 077252      045      116      045  T39SFS: .ASCIZ  '#N#A State Of Extended Features Switch  ='
10791 077324      045      116      045  T39SBS: .ASCIZ  '#N#A State Of Buffering Switch      ='
10792 077376      045      101      040  T39OFF: .ASCIZ  '#A OFF'
10793 077405      045      101      040  T39ON: .ASCIZ  '#A ON '
10794 077414      045      116      045  T39MCL: .ASCIZ  '#N#A M7196 Microcode Revision Level      =#D2'
10795
10796 077472      000000          T39RL: .WORD  0
10797          .EVEN
10798          .EVEN
10799
10800          ;+
10801          ;LOCAL STORAGE FOR THIS TEST
10802          ;--
10803
10804 077474      000000          T39DAT: .WORD  0          ;LOGICAL RESPONSE TO QUESTION
10805 077476          T39REST:
10806 077476          SAVREG          ;SAVE THE REGISTERS
10807 077502      012701      075560      MOV      #T39PACKET,R1          ;START OF THE PACKET
10808 077506      012721      140006      MOV      #140006,(R1)+        ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
10809 077512      012721      075570      MOV      #T39TAD,(R1)+        ;ADDRESS OF DATA BLOCK
10810 077516      005021          CLR      (R1)+          ;EXTENDED ADDRESS
10811 077520      012721      000006      MOV      #6.,(R1)+          ;SIZE OF DATA BLOCK IN BYTES
10812 077524      005021          CLR      (R1)+          ;CLEAR BSEL0 AND BSEL1
10813 077526      005021          CLR      (R1)+          ;CLEAR SEL2
10814 077530      005011          CLR      (R1)           ;CLEAR DATA AREA
10815 077532      000207          RTS      PC              ;RETURN
10816
10817          ;+
10818          ;LOCAL TEXT MESSAGES FOR TEST
10819          ;--
10820
10821 077534      103      157      156  TST39ID: .ASCIZ  'Configuration Typeout'
10822          .EVEN
10823 077562          ENDTST
10824          L10076: TRAP      C#ETST
10825          .SBTTL  TEST 12: SCOPE LOOPS
10826
10827

```

TEST 12: SCOPE LOOPS

```

10828
10829
10830
10831
10832
10833
10834
10835
10836
10837
10838
10839
10840
10841
10842
10843
10844
10845
10846
10847
10848
10849
10850
10851
10852
10853
10854
10855
10856
10857
10858
10859
10860
10861
10862
10863
10864 077564
10865 077564
10866 077564
10867 077564 104421
10868 077564 001403
10869 077564 012700 101155
10870 077564 000402
10871 077570 012700 101222
10872 077574 004737 016510
10873 077576 004737 017762
10874 077602 004737 010276
10875 077606 004737 015774
10876 077612 103402
10877 077614 000137
10878 077620 004737
10879 077624 103405
10880 077626 010001
10881 077630
10882 077630 104455
10883 077632 002261
10884 077634 003652
10885 077636 012034
10886 077640 012700 100310

```

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

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

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

```

BGNTST
RFLAGS R0
;GET OPERATOR FLAGS
;BH, IF OK TO RUN
; "TEST NOT EXECUTED"
; JUST EXIT IF NOT
; TEST ID MESSAGE
; DO THE COMMON SETUP
; SEE IF MANUAL INTERVENTION ALLOWED
; CARRY SET IF INTERVENTION ALLOWED
; EXIT IF NO MANUAL INTERVENTION
; DO A SOFT INIT
; BRANCH IF OK
; CONTENTS OF TSSR REGISTER
; REPORT FATAL ERROR
;MENU OF SCOPE LOOP SELECTIONS

```

10886	077644	012701	000007		MOV	#7,R1	;MAXIMUM ALLOWED SELECTION
10887	077650	004737	017540		JSR	PC,GETSEL	;GO GET THE OPERATORS SELECTION
10888	077654	005700			TST	R0	;WAS ZERO SPECIFIED ?
10889	077656	001760			BEQ	2:	;REPEAT MENU IF YES.
10890	077660	020027	000007		CMP	R0,#7	;EXTENDED TSSR ?
10891	077664	001015			BNE	3:	;BRANCH IF NOT
10892	077666	005737	002226		TST	EXTFEA	;CHECK FOR EXTENDED FEATURES SET
10893	077672	001012			BNE	3:	;BR, IF IT IS ON
10894	077674				PRINTF	#EXFMSG	;WARN OPERATOR EXTENDED FEATURES CLEAR
	077674	012746	101077				MOV #EXFMSG,-(SP)
	077700	012746	000001				MOV #1,-(SP)
	077704	010600					MOV SP,R0
	077706	104417					TRAP C:PNTF
	077710	062706	000004				ADD #4,SP
10895	077714	000137	077620		JMP	2:	;GO BACK TO BASIC MENU
10896	077720	010004		3:	MOV	R0,R4	;SAVE THE MENU SELECTION
10897	077722				SETPRI	#PRI07	;RAISE THE PRIORITY
	077722	012700	000340				MOV #PRI07,R0
	077726	104441					TRAP C:SPRI
10898	077730	005037	100302		CLR	TTION	;ASSUME INTERRUPTS ARE ENABLED
10899	077734	032737	000100	177560	BIT	#100,#ATTICSR	;ARE TTI INTERRUPTS ON ?
10900	077742	001005			BNE	4:	;BRANCH IF YES
10901	077744	005237	100302		INC	TTION	;FLAG SET IF INTERRUPTS OFF
10902	077750	052737	000100	177560	BIS	#100,#ATTICSR	;ENABLE INTERRUPTS
10903	077756	012701	000060		MOV	#TTIVEC,R1	;START OF TTI VECTORS
10904	077762	011137	100304		MOV	(R1),TVECSAV	;SAVE THE CURRENT TTI VECTOR
10905	077766	012721	100210		MOV	#60,(R1)+	;SET NEW INTERRUPT ROUTINE
10906	077772	011137	100306		MOV	(R1),TPRISAV	;SAVE THE VECTOR PRIORITY
10907	077776	012711	000340		MOV	#PRI07,(R1)	;USE PRIORITY SEVEN
10908	100002				SETPRI	#PRI00	;LOWER INTERRUPT BR LEVEL
	100002	012700	000000				MOV #PRI00,R0
	100006	104441					TRAP C:SPRI
10909	100010	006304			ASL	R4	;CONVERT TO WORD OFFSET
10910	100012	000174	100016		JMP	#6*(R4)	;JUMP TO PROPER LOOP
10911	1J0016	077620		6:	.WORD	2:	;RETYPE THE MENU
10912	100020	100036			.WORD	10:	;TSBA READ ACCESS
10913	100022	100046			.WORD	15:	;TSSR READ ACCESS
10914	100024	100060			.WORD	20:	;TSSR WRITE ACCESS
10915	100026	100100			.WORD	25:	;TSDB HIGH BYTE WRITE ACCESS
10916	100030	100124			.WORD	30:	;TSDB LOW BYTE WRITE ACCESS
10917	100032	100150			.WORD	35:	;TSDB MAINTENANCE MODE
10918	100034	100170			.WORD	40:	;TSDBX WRITE ACCESS
10919							;LEAVE THE TEST
10920							
10921							
10922	100036	105065	000000	10:	CLRB	TSDB(R5)	;ENTER MAINTENANCE MODE
10923	100042	011500		12:	MOV	(R5),R0	;READ TSBA REGISTER
10924	100044	000776			BR	12:	;LOOP UNTIL HALTED
10925							
10926							
10927	100046	012703	000002	15:	MOV	#TSSR,R3	;ADDRESS OF TSSR REGISTER
10928	100052	060503			ADD	R5,R3	;POINT TO TSV05'S REGISTERS
10929	100054	011300		18:	MOV	(R3),R0	;READ TSSR REGISTER
10930	100056	000776			BR	18:	;LOOP UNTIL STOPPED
10931							
10932	100060	004737	017456	20:	JSR	PC,GETPAT	;READ THE DATA PATTERN
10933	100064	010001			MOV	R0,R1	;DATA PATTERN FOR LOOP

TEST 12: SCOPE LOOPS

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

TEST 12: SCOPE LOOPS

```

10991
10992 100276
10993 100276 104432
      100276 000736
10994
10995
10996
10997
10998
10999
11000 100302 000000
11001 100304 000000
11002 100306 000000
11003
11004
11005
11006
11007
11008
11009
11010 100310 100342 100415 100443
11011 100324 100614 100652 100720
11012
11013
11014 100342 012 123 105 11: .ASCIZ <12>'SELECT SCOPE LOOP FROM FOLLOWING OPTIONS:'
11015 100415 012 011 060 21: .ASCIZ <12>' 0 Display This Menu'
11016 100443 011 061 011 31: .ASCIZ ' 1 TSBA Read Access'
11017 100467 011 062 011 41: .ASCIZ ' 2 TSSR Read Access'
11018 100513 011 063 011 51: .ASCIZ ' 3 Initialize (TSSR Write Access)'
11019 100555 011 064 011 61: .ASCIZ ' 4 TSDB High Byte Write Access'
11020 100614 011 065 011 71: .ASCIZ ' 5 TSDB Low Byte Write Access'
11021 100652 011 066 011 81: .ASCIZ ' 6 TSDB Maintenance Mode Write Access'
11022 100720 011 067 011 91: .ASCIZ ' 7 TSDBX (TSSR High Byte) Write Access'
11023 100767 011 136 103 101: .ASCIZ ' ^C To Return to Diagnostic Supervisor'
11024 101036 000 111: .ASCIZ ''
11025 101037 124 171 160 121: .ASCIZ 'Type RETURN To Stop Scope Loops'
11026 101077 045 116 045 EXFMSG: .ASCIZ 'MSA *** Extended Features Switch Not On *** '
11027 101155 123 164 141 T4ONE: .ASCIZ 'Stand-alone Scope Loops Not Executed'
11028 101222 123 143 157 TST40ID: .ASCIZ 'Scope Loops'
11029
11030 101236
      101236
      101236 104401
      101240
11031
11033
11039
11044
11050
11051 101240
      101240
11052
11053
11054
11055
11056
11057

641:
631: EXIT TST ;EXIT THE TEST

;651: JMP 200 ;RETURN TO SUPERVISOR
;*****JUMP TO 200 IS NOT SUPPORTED ON XXDP+ VERSION 2.X.
;
;LOCAL STORAGE FOR THIS TEST
;-

TTION: .WORD 0 ;WORD SET IF SUPERVISOR TTI INTER OFF
TVECSAV: .WORD 0 ;SAVE TTI VECTOR
TPRISAV: .WORD 0 ;SAVE TTI PRIORITY

;
;MENU FOR OPERATOR INPUT FOR SCOPE LOOPS
;-

SCMENU: .EVEN
        .WORD 11,21,31,41,51,61
        .WORD 71,81,91,101,111,121,0

11: .ASCIZ <12>'SELECT SCOPE LOOP FROM FOLLOWING OPTIONS:'
21: .ASCIZ <12>' 0 Display This Menu'
31: .ASCIZ ' 1 TSBA Read Access'
41: .ASCIZ ' 2 TSSR Read Access'
51: .ASCIZ ' 3 Initialize (TSSR Write Access)'
61: .ASCIZ ' 4 TSDB High Byte Write Access'
71: .ASCIZ ' 5 TSDB Low Byte Write Access'
81: .ASCIZ ' 6 TSDB Maintenance Mode Write Access'
91: .ASCIZ ' 7 TSDBX (TSSR High Byte) Write Access'
101: .ASCIZ ' ^C To Return to Diagnostic Supervisor'
111: .ASCIZ ''
121: .ASCIZ 'Type RETURN To Stop Scope Loops'
EXFMSG: .ASCIZ 'MSA *** Extended Features Switch Not On *** '
T4ONE: .ASCIZ 'Stand-alone Scope Loops Not Executed'
TST40ID: .ASCIZ 'Scope Loops'

.EVEN
ENDTST

L10077: TRAP C1EXIT
        .WORD L10077-

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

```

HARDWARE PARAMETER CODING SECTION

```

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

```

SOFTWARE PARAMETER CODING SECTION

```

11074                                     .SBTTL  SOFTWARE PARAMETER CODING SECTION
11075
11076                                     ;**
11077                                     ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
11078                                     ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
11079                                     ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
11080                                     ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
11081                                     ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
11082                                     ; WITH THE OPERATOR.
11083                                     ;--
11084 101372 BGNSFT
11085 101372 .WORD L10101-L#SOFT/2
11086 101374 L#SOFT::
11087         GPRML  SPM1,0,-1,YES          ; GET TRANSPORT TEST FLAG.
11088         GPRML  SPM4,2,-1,YES          ; GET ITERATION CONTROL.
11089         .WORD  T#CODE
11090         .WORD  SPM4
11091         .WORD  -1
11092         GPRMD  SPM6,4,D,7777,0,7777,YES ; GET LOCAL ERROR LIMIT
11093         GPRMD  SPM7,6,D,7777,0,7777,YES ; GET GLOBAL ERROR LIMIT
11094         ENDSFT
11095         .EVEN
11096
11097 L10101:
11098
11099 SPM1:  .ASCIZ  'ENABLE TRANSPORT TESTS '
11100 SPM4:  .ASCIZ  'INHIBIT ITERATIONS '
11101 SPM6:  .ASCIZ  'PER TEST ERROR LIMIT '
11102 SPM7:  .ASCIZ  'PER UNIT ERROR LIMIT '
11103       .SBTTL  PATCH AREA
11104
11105       ;
11106       ; FINALLY A GENEROUS PATCH AREA.
11107       ;
11108       ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
11109       ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
11110       ;
11111
11112 PATCH::
11113         .BLKW  32.
11114
11115         .=.!377*1
11116 LASTAD .EVEN          ;SET LAST USED ADDRESS.
11117         .WORD  0
11118         .WORD  0
11119
11120 L#LAST::
11121         ENDMOD
11122         .END

```

Symbol table

ADDSSR	012126	G	C#AUTO=	000061	DEVNRD	022645	FUSI	004117	INTMAS	016144
ADR	000020	G	C#BRK =	000022	DEVNXR	022563	F#AU =	000015	INTR	016216 G
AMBTSS	006635		C#BSEG=	000004	DEVONL	022513	F#AUTO=	000020	INTREC	002224 G
ASSEMB=	000010		C#BSUB=	000002	DEVSUM	022456	F#BGN =	000040	INTVEC	016146
A1716 =	000003		C#CEFG=	000045	DFPTBL	002156 G	F#CLEA=	000007	INTX	004300
BADDAT	003156	G	C#CLCK=	000062	DIAGMC=	000000	F#DU =	000016	INVERT	020444 G
BADSSR	015700	G	C#CLEA=	000012	DICEB =	000001	F#END =	000041	IOKCKI=	000200
BDVPCR=	177520	G	C#CLOS=	000035	DSBINT	016204	F#HARD=	000004	IOKSTP=	000001
BENBSW	002230	G	C#CLP1=	000006	DUAD12	004643	F#HW =	000013	IPRI	002212 G
BIE =	040000		C#CVEC=	000036	DUFLG	003112 G	F#INIT=	000006	ISR	000100 G
BIT0 =	000001	G	C#DCLN=	000044	DUMMY	003062	F#JMP =	000050	IVEC	002210 G
BIT00 =	000001	G	C#DODU=	000051	EF.CON=	000036 G	F#MOD =	000000	IXE	004000 G
BIT01 =	000002	G	C#DRPT=	000024	EF.NEW=	000035 G	F#MSG =	000011	I#AU =	000041
BIT02 =	000004	G	C#DU =	000053	EF.PWR=	000034 G	F#PROT=	000021	I#AUTO=	000041
BIT03 =	000010	G	C#EDIT=	000003	EF.RES=	000037 G	F#PWR =	000017	I#CLN =	000041
BIT04 =	000020	G	C#ERDF=	000055	EF.STA=	000040 G	F#RPT =	000012	I#DU =	000041
BIT05 =	000040	G	C#ERMR=	000056	EMAXDU	016777	F#SEG =	000003	I#HRD =	000041
BIT06 =	000100	G	C#ERRO=	000060	EN =	000000	F#SOFT=	000005	I#INIT=	000041
BIT07 =	000200	G	C#ERSF=	000054	ENAIN	016152	F#SRV =	000010	I#MOD =	000041
BIT08 =	000400	G	C#ERSO=	000057	ENVIRN	020112	F#SUB =	000002	I#MSG =	000041
BIT09 =	001000	G	C#ESCA=	000010	EPRTSW	002200 G	F#SW =	000014	I#PROT=	000040
BIT1 =	000002	G	C#ESG=	000005	EPRT1	006360	F#TEST=	000001	I#PTAB=	000041
BIT10 =	002000	G	C#ESUB=	000003	EPRT2	006360	GDDAT	003160 G	I#PWR =	000041
BIT11 =	004000	G	C#ETST=	000001	ERCM	011733	GERRMA	002174 G	I#RPT =	000041
BIT12 =	010000	G	C#EXIT=	000032	ERRHI	002236 G	GETPAT	017456 G	I#SEG =	000041
BIT13 =	020000	G	C#GETB=	000026	ERRK	016756	GETSEL	017540 G	I#SETU=	000041
BIT14 =	040000	G	C#GETW=	000027	ERRLO	002240 G	G#CNT0=	000200	I#SFT =	000041
BIT15 =	100000	G	C#GMAN=	000043	ERRNO =	002261	G#DELM=	000372	I#SRV =	000041
BIT2 =	000004	G	C#GPHR=	000042	ERRVEC=	000004 G	G#DISP=	000003	I#SUB =	000041
BIT3 =	000010	G	C#GPLO=	000030	ERTABE	003376	G#EXCP=	000400	I#TST =	000041
BIT4 =	000020	G	C#GPRI=	000040	ERTABL	003176	G#HILI=	000002	J#JMP =	000167
BIT5 =	000040	G	C#INIT=	000011	ESUM	016760	G#LOLI=	000001	KIPAR0=	172340
BIT6 =	000100	G	C#INLP=	000020	EVL =	000004 G	G#NO =	000000	KIPAR1=	172342
BIT7 =	000200	G	C#MANI=	000050	EXBCNT=	000010	G#OFFS=	000400	KIPAR2=	172344
BIT8 =	000400	G	C#MEM =	000031	EXFMSG	101077	G#OFSI=	000376	KIPAR3=	172346
BIT9 =	001000	G	C#MSG =	000023	EXPBRE	015502 G	G#PRMA=	000001	KIPAR4=	172350
BOE =	000400	G	C#OPEN=	000034	EXPD	002232 G	G#PRMD=	000002	KIPAR5=	172352
BRINIT	004457		C#PNTB=	000014	EXPGOT	004533	G#PRML=	000000	KIPAR6=	172354
BSELO =	000000		C#PNTF=	000017	EXPGT2	004567	G#RADA=	000140	KIPAR7=	172356
BSEL1 =	000001		C#PNTS=	000016	EXPMG	002322 G	G#RADB=	000000	KIPDR0=	172300
CHKAMB	016044		C#PNTX=	000015	EXPREC	015474 G	G#RADD=	000040	KIPDR1=	172302
CHKMAN	017762	G	C#QIO =	000377	EXTA	005772	G#RADL=	000120	KIPDR2=	172304
CHKTSS	016336		C#RDBU=	000007	EXTEND	005770	G#RADO=	000020	KIPDR3=	172306
CKDROP	017202		C#REFG=	000047	EXTFEA	002226 G	G#XFER=	000004	KIPDR4=	172310
CKEMAX	017102		C#RESE=	000033	E#END =	002100	G#YES =	000010	KIPDR5=	172312
CKMSG	011360	G	C#REVI=	000003	E#LOAD=	000035	HIADDR=	001400	KIPDR6=	172314
CKMSG2	011500	G	C#RFLA=	000021	FATERR=	000060	HOE =	100000 G	KIPDR7=	172316
CKRAM	011114	G	C#RPT =	000025	FATFLG	002222 G	HPM1	101262	KTENAB	003134 G
CKRAM2	011224	G	C#SEFG=	000046	FERCM	011722	HPM2	101316	KTFLG	003132 G
CHOPKT	020610	G	C#SPRI=	000041	FIFEXP	012170 G	HPM3	101342	KTINIT	020272
CONFIG	017250		C#SVEC=	000037	FIF1MS	012242	IBE =	010000 G	KTOFF	017274
COUNT	002310	G	C#TPRI=	000013	FIF2MS	012311	IDU =	000040 G	KTON	017256
CSRADD	002206	G	DATA	002312 G	FNOINT	004215	IER =	020000 G	LERRMA	002172 G
CTAB	003164	G	DATASC	017514	FORCER	002176 G	IFALT	004256	LERRNO=	000000
CTABE	003176	G	DEBUGM	011632	FREE	003124 G	INCERK	017044	LISTAL=	000001
CTABM	003164	G	DEVCNT	002220 G	FREEHI	003130	INTCPC	016150	LOE =	040000 G
C#AU =	000052		DEVDR0	022726	FRESIZ	003126 G	INTFLA	016145	LOOPCN	002216 G

Symbol table

LOOPCO	013126	L10001	002176	L10073	066214	ONEFIL	= 000000	PRMSGF	014552	G
LOOPFL	003162	L10002	005766	L10074	064354	0#APTS	= 000000	PRMSGO	014732	
LOT	= 000010	L10003	012044	L10075	074052	0#AU	= 000001	PRMSG1	014777	
L#ACP	002110	L10004	012062	L10076	077562	0#BGNR	= 000001	PRMSG2	015035	
L#APT	002036	L10005	012100	L10077	101236	0#BGNS	= 000001	PROASC	014420	
L#AU	021702	L10006	012106	L10100	101262	0#DU	= 000001	PR1ASC	014465	
L#AUT	002070	L10007	012124	L10101	101402	0#ERRT	= 000000	PST32W	003152	G
L#AUTO	022106	L10010	012142	MEMADD	013754	0#GNSW	= 000001	PUNIT	021634	
L#CCP	002106	L10011	012166	MEMCK	020626	0#POIN	= 000001	PW.D11	= 000021	
L#CLEA	022166	L10012	012240	MENASC	017731	0#SETU	= 000000	PW.D13	= 000022	
L#CO	002032	L10013	012410	MENERR	017656	PASRPT	021404	PW.D22	= 000020	
L#DEPO	002011	L10014	013124	MENRES	017760	PATCH	101542	PW.NOP	= 000000	
L#DESC	003410	L10015	013752	MIMENU	072732	PATDAT	017512	PW.N01	= 000023	
L#DESP	002076	L10016	013774	MMVEC	= 000250	PC.ERA	= 002400	PW.RDE	= 000024	
L#DEVP	002060	L10017	015500	MSA.FR	= 000006	PC.IER	= 002000	PW.RDR	= 000001	
L#DISP	002124	L10020	015506	MSA.NO	= 000000	PC.N00	= 001000	PW.RDS	= 000005	
L#DLY	002116	L10021	015514	MSA.NR	= 000004	PC.REL	= 000000	PW.RFI	= 000003	
L#DTP	002040	L10022	015526	MSA.V0	= 000002	PC.REW	= 000400	PW.WCT	= 000006	
L#DTYP	002034	L10023	015550	MSGEXP	012144	PKBCNT	= 000006	PW.WFI	= 000004	
L#DU	022000	L10024	015576	MSGLOO	013064	PKHI	= 000004	PW.WFM	= 000007	
L#DUT	002072	L10025	015736	MSGSTA	012350	PKLOW	= 000002	PW.WMI	= 000010	
L#DVTY	003402	L10026	016246	MSGSUB	013742	PKTADD	007554	PW.WNP	= 000011	
L#EF	002052	L10030	021632	MS.ATT	= 000006	PKTFRM	007516	PW.WTR	= 000002	
L#ENVI	002044	L10031	021776	MS.EXT	= 000200	PKTGET	012064	P.ACK	= 100000	
L#ETP	002102	L10032	022104	MS.RSD	= 000001	PKTMES	012110	P.CMD	= 000037	
L#EXP1	002046	L10033	022164	MS.RSF	= 000020	PKTRAM	004745	P.CONT	= 000012	
L#EXP4	002064	L10034	022212	MS.RST	= 000010	PKTSSR	012046	P.CVC	= 040000	
L#EXP5	002066	L10035	022454	M8186	005554	PNT	= 001000	P.FMT	= 000140	
L#HARD	101242	L10036	023754	M8189	005645	PRAMPK	013776	P.FORM	= 000011	
L#HIME	002120	L10037	025746	NBA	= 002000	PRASC	014523	P.GETS	= 000017	
L#HPCP	002016	L10040	024230	NEWPAS	021340	PRBEXP	015470	P.IE	= 000200	
L#HPTP	002022	L10041	024474	NODEV	003114	PRBMSG	015336	P.INIT	= 000013	
L#HW	002156	L10042	031342	NOEXTF	027536	PRBREC	015472	P.MODE	= 007400	
L#ICP	002104	L10043	026322	NOINIT	004335	PRBTOT	015423	P.OPP	= 020000	
L#INIT	021106	L10044	026612	NOINTR	004221	PRBYTE	015122	P.POSI	= 000010	
L#LADP	002026	L10045	027110	NOITS	002170	PRI	= 002000	P.READ	= 000001	
L#LAST	102004	L10046	027542	NOMAN	020016	PRIADD	010160	P.SWB	= 010000	
L#LOAD	002100	L10047	034132	NOMEM	005460	PRIAO	010230	P.WRIT	= 000005	
L#LUN	002074	L10050	031676	NP.IR	= 000200	PRIBX0	007612	P.WRTC	= 000004	
L#MREV	002050	L10051	032270	NP.L00	= 000040	PRIEQU	010060	P.WRTS	= 000006	
L#NAME	002000	L10052	032676	NP.OUT	= 000100	PRIPKT	007370	QVP	002204	G
L#PRIO	002042	L10053	037724	NP.WRP	= 000020	PRIRAM	010066	RAMASC	014156	
L#PROT	021076	L10054	035216	NSI	004152	PRITAD	010274	RAMDAT	002242	G
L#PRT	002112	L10055	036150	NSINIT	004407	PRITSS	006024	RAMERR	015510	G
L#REPP	002062	L10056	050036	NUL	004527	PRITO	010356	RAMEXP	015530	G
L#REV	002010	L10057	040230	NULCR	004530	PRIT1	010421	RAMFOR	010116	
L#RPT	022214	L10060	041440	NUMENU	075342	PRI XOR	007742	RAMSIZ	002302	G
L#SOFT	101374	L10061	043000	NXM	= 004000	PRI00	= 000000	RAMTAD	015516	G
L#SPC	002056	L10062	043356	NXMFLG	003136	PRI01	= 000040	RCVHIA	002304	G
L#SPCP	002020	L10063	044630	NXMHI	003142	PRI02	= 000100	RCVLOA	002306	G
L#SPTP	002024	L10064	045674	NXMLO	003140	PRI03	= 000140	RDERR	005206	
L#STA	002030	L10065	051316	NXMTST	021002	PRI04	= 000200	RECMG	002466	G
L#SW	002166	L10066	062144	NXR	003740	PRI05	= 000240	RECV	002234	G
L#TEST	002114	L10067	053322	NXRERR	005736	PRI06	= 000300	REGSAV	017422	
L#TIML	002014	L10070	054610	NXR	003777	PRI07	= 000340	RETERR	005372	
L#UNIT	002012	L10071	056060	NXTU	021352	PRMESS	014242	REWIND	011014	G
L10000	002164	L10072	057324	OFL	= 000100	PRMNO	002320	RMCHBE	= 000167	

Symbol table

RMCHEN=	000200	S1.I1R=	020000	TSV2	002000	G	T11	074054	G	T15BF2	033430
RMMSG8=	000215	S1.I2R=	040000	TSV3	002176	G	T12	077564	G	T15BS0	033430
RMMSG8=	000234	S1.PAR=	100000	TSV4	021076	G	T12BFR	027632		T15BS1	033431
RMPKTB=	000201	S2.ATI=	000010	TSV5	022776	G	T12BKG	030307		T15DAT	032730
RMPKTE=	000210	S2.BTI=	000004	TSV6	101240	G	T12BLK	027664		T15L00	031374
RMR	= 010000	S2.DIM=	000200	TTIBFR=	177562	G	T12CHA	031300		T15PAC	032720
RWPACK	011110	S2.ILW=	000100	TTICSR=	177560	G	T12CKR	031060	G	T15PK2	033420
SC	= 100000	S2.INR=	000020	TTION	100302		T12CON	030666		T15RES	034022
SCE	= 020000	S2.OUT=	000040	TTION2	070776		T12DAT	027620		T15RT2	034074
SCHERR	005300	S2.UND=	000003	TTIVEC=	000060	G	T12DPR	030466		T15SSR	033436
SCME	005013	TBLEND=	003062	TVECSA	100304		T12GET	030046		T15S2	033432
SCMENU	100310	TCOASC	006476	TVSAV2	071000		T12HIA	027652		T15S3	033434
SDELAY	010660	TCOCOD	006676	T#ARGC=	000001		T12KT	027660		T16BEN	037610
SELASC	017724	TEMP1	003116	T#CODE=	001130		T12LOA	027654		T16BFR	037562
SELDAT=	000004	TEMP2	003120	T#ERRN=	002261		T12L00	025776		T16BFS	037602
SEL2	= 000002	TERCLS=	000016	T#EXCP=	000000		T12MSG	030211		T16CLR	037426
SETMAP	017316	TESTNO=	000014	T#FLAG=	000040		T12NIN	030375		T16DAT	037550
SETU	021436	TEXASC	006435	T#GMAN=	000000		T12NXM	030557		T16DT2	037620
SFFMSG	012102	TFCASC	006537	T#HILI=	000776		T12PAC	027610		T16D01	036202
SFHERR	003705	TIMEXP	015552	T#LAST=	000001		T12PAR	027656		T16D28	036204
SFIERR	003652	TIMSG0	015600	T#LOLI=	000000		T12SET	031236		T16D53	036206
SFIMSG	012034	TINERR	012021	T#LSYM=	010000		T12SWR	031170		T16D78	036210
SFPTBL	002166	TMPBFR	002632	T#LTNO=	000014		T12TBE	030016		T16LEN	036532
SIFLAG	003154	TNAM	016704	T#NEST=	177777		T12WRT	030122		T16L00	034152
SIMSG	011766	TPRISA	100306	T#NS0 =	000000		T121L0	026110		T16PAC	037540
SKIPT	003400	TPSAV2	071002	T#NS1 =	000005		T122L0	026444		T16PK2	037610
SOFINI	015774	TRANST	002166	T#NS2 =	000002		T123L0	026664		T16REJ	036644
SPACE	010466	TSBA	= 000000	T#NS3 =	000003		T124L0	027306		T16RES	037360
SPM1	101402	TSBAH =	000001	T#PTNU=	000000		T124TS	027662		T16SEX	037520
SPM4	101432	TSDB =	000000	T#SAVL=	177777		T13BFR	023412		T16SRD	037452
SPM6	101462	TSDBH =	000001	T#SEGL=	177777		T13DAT	023400		T16SSR	036262
SPM7	101512	TSFCOD	007236	T#SEK0=	010000		T13L00	023014		T16TSB	036430
SR0	= 177572	TSREJ =	000006	T#SUBN=	000000		T13MEM	023505		T16T01	036761
SR1	= 177574	TSSDEF	006606	T#TAGL=	177777		T13NBA	023544		T16T28	037060
SR2	= 177576	TSSR =	000002	T#TAGN=	010102		T13PAC	023370		T16T53	037160
SR3	= 172516	TSSRBI	003502	T#TEMP=	000000		T13RES	023706		T16T78	037260
SSR	= 000200	TSSRFO	006415	T#TEST=	000014		T13SSR	023617		T16WMI	037472
STATCO	012412	TSSRH =	000003	T#TSTM=	177777		T14BFR	024526		T162SS	036317
SVCGBL=	000000	TSSX	004020	T#TSTS=	000001		T14BS0	024520		T163SS	036363
SVCINS=	000000	TSTBLK	002752	T#AU =	010031		T14BS1	024521		T17BEN	047712
SVCSUB=	000001	TSTCNT	002214	T#AUT =	010033		T14BS2	024522		T17BFR	047572
SVCTAG=	000000	TSTEND	016720	T#CLE =	010034		T14DAT	024520		T17BFS	047612
SVCTST=	000001	TSTFLA	002314	T#DU =	010032		T14DTA	025140		T17CLE	047476
S#LSYM=	010000	TSTL00	016456	T#HAR=	010100		T14L00	023774		T17CLR	047310
SO.IDB=	000010	TSTPTR	002316	T#HW =	010000		T14NBA	025152		T17DAT	047560
SO.IFB=	000002	TSTSET	016510	T#INI=	010030		T14NIN	025413		T17DT2	047730
SO.IFP=	000001	TST12I	030020	T#MSG=	010025		T14PAC	024510		T17EXE	046054
SO.ILD=	000020	TST13I	023432	T#PRO=	010027		T14PK2	025130		T17EXP	045732
SO.ION=	000040	TST14I	025571	T#RPT=	010035		T14RES	025636		T17EXS	045752
SO.IRD=	000100	TST15I	034003	T#SEG=	010000		T14RST	025674		T17L00	037754
SO.IRW=	000004	TST16I	036212	T#SOF=	010101		T14SSR	025323		T17MSK	045726
SO.ISP=	000200	TST17I	046156	T#SRV=	010026		T14TSB	025505		T17PAC	047550
S1.ICE=	002000	TST18I	050546	T#SUB=	010074		T142RE	025226		T17PK2	047720
S1.IEO=	010000	TST19I	057532	T#SW =	010001		T15AM2	033512		T17RFI	047456
S1.IFM=	001000	TST20I	064532	T#TES=	010077		T15AM3	033613		T17RSF	047354
S1.IHE=	000400	TST39I	077534	T1	022776	G	T15AM4	033715		T17SET	047516
S1.IID=	004000	TST40I	101222	T10	066216	G	T15BFR	032742		T17SNP	047376

Symbol table

T17SRD	047334	T19WCT	061472	T38FLG	003150 G	T39NE	076333	WF.IED=	000010
T17SSR	046175	T19WFI	061436	T3.1	025776	T39NFL	076330	WF.IER=	000004
T17WFD	046054	T19WFM	061512	T3.2	026336	T39OFF	077376	WF.IHI=	000200
T17WFI	047422	T19WST	061073	T3.3	026626	T39OF2	076574	WF.IRE=	000040
T171CM	046515	T191CM	060230	T3.4	027124	T39ON	077405	WF.IWF=	000020
T172CM	046577	T192CM	060312	T38ASC	074005	T39ON2	076640	WF.IWR=	000100
T172SS	046232	T192SS	057630	T38ASN	074024	T39OUT	077201	WF.I3R=	000002
T173CM	046673	T193CM	060406	T38ASO	073702	T39PAC	075560	WF.I4R=	000001
T173SS	046276	T193SS	057674	T38AS1	073747	T39PK2	076260	WRTCHR	010662 G
T174CM	046757	T194SS	057741	T38BFR	071026	T39PK3	076310	WRTERR	005113
T174SS	046343	T195CM	060474	T38BSO	071020	T39PK4	076320	WRTMSG	005056
T175CM	047042	T195SS	060004	T38BS1	071021	T39RES	077476	WSMBK	020620 G
T175SS	046406	T196CM	060550	T38BS2	071022	T39RL	077472	XFERAS	015740
T176CM	047116	T196SS	060050	T38CNT	074050	T39SBS	077324	XNXM	016376
T176SS	046452	T197CM	060633	T38DAT	073340	T39SFS	077252	XORBFO	007674
T177CM	047224	T197SS	060113	T38DLY	071004	T39SIZ	076326	XORFOR	010012
T18BFR	051252	T198CM	060721	T38DSW	071530	T39SSR	077111	XST0 =	000006 G
T18CMP	050753	T198SS	060162	T38DTA	071520	T39TAD	075570	XST1 =	000010 G
T18DAT	051240	T199CM	061010	T38EAI	071526	T39WPN	076756	XST2 =	000012 G
T18DT2	051310	T2	023756 G	T38EB	071502	T39WR	076322	XST3 =	000014 G
T18FXP	050516	T2.1	023774	T38ID	072705	T39WRT	076703	XST4 =	000016 G
T18LOO	050056	T2.2	024244	T38INT	072276	T4	031344 G	XSORT=	000002
T18MSK	050512	T20BEN	066072	T38MBP	073400	T4.1	031374	XSORT=	000001
T18PAC	051230	T20BFR	065752	T38MSG	072645	T4.2	031700	XSOIE =	000040
T18PK2	051300	T20BFS	065772	T38MS1	072506	T4.3	032272	XSOILA=	000400
T18SET	051120	T20CLE	065664	T38MS2	072601	T4ONE	101155	XSOILC=	001000
T18SMI	051076	T20CLR	065452	T38MS3	071645	T5	034134 G	XSOLET=	020000
T18SRD	051056	T20CWP	065357	T38NBA	072132	T5.1	034152	XSONOT=	000200
T18SSR	050605	T20DAT	065740	T38NE	071570	T5.2	035232	XSONEF=	002000
T18XS	050542	T20DT2	066110	T38OFL	072362	T6	037726 G	XSOONL=	000100
T182SS	050642	T20EXE	064532	T38ONL	072326	T6.1	037754	XSOPED=	000010
T183SS	050706	T20EXP	064412	T38PAC	071010	T6.2	040244	XSORLL=	010000
T18BEN	062022	T20EXS	064432	T38PK2	071510	T6.3	041454	XSORLS=	040000
T18BFC	057346	T20LOO	062164	T38PK3	071540	T6.4	043014	XSOTMK=	100000
T18BFR	061702	T20MSK	064406	T38PK4	071560	T6.5	043372	XSOVCK=	000020
T18BFS	061722	T20PAC	065730	T38RES	073342	T6.6	044644	XSOWLE=	004000
T19CLE	061550	T20PK2	066100	T38SIZ	071566	T7	050040 G	XSOWLK=	000004
T19CLR	061304	T20RFI	065536	T38SSR	072206	T8	051320 G	XXCOMM	003122 G
T19CNV	061614	T20RSF	065256	T38SST	072416	T8.1	051326	X#ALWA=	000000
T19DAT	061670	T20SET	065704	T38TAD	071020	T8.2	053736	X#FALS=	000040
T19DT2	062040	T20SEX	065646	T38WLE	072065	T8.3	054624	X#OFFS=	000400
T19EXE	057532	T20SRD	065476	T38WR	071562	T8.4	056074	X#TRUE=	000020
T19EXP	057412	T20SSR	064561	T38WRL	072024	T9	062146 G	X1.COR=	020000
T19XS	057432	T20SWP	065147	T38WRT	071740	T9.1	062164	X1.DLT=	100000
T19LOO	051336	T20WFI	065556	T38BFR	075576	UAM =	000200 G	X1.MBZ=	017375
T19MSK	057406	T20WFM	065612	T39BSO	075570	UNITN	002202 G	X1.RBP=	000400
T19PAC	061660	T20WMI	065632	T39BS1	075571	UNREC =	000006	X1.SPA=	040000
T19PK2	062030	T20WNP	065516	T39BS2	075572	USI	004123	X1.UNC=	000002
T19PRE	057344	T202SS	064616	T39DAT	077474	WAITF	016250 G	X2.BUF=	000100
T19RFI	061416	T203SS	064662	T39DLY	075552	WC.IFA=	000200	X2.EXT=	000200
T19RSF	061350	T204SS	064727	T39DSW	076300	WC.IFE=	000002	X2.OPM=	100000
T19RST	061200	T205SS	064772	T39DTA	076270	WC.IG0=	000001	X2.RCE=	040000
T19SET	061570	T206SS	065036	T39EAI	076276	WC.IRE=	000010	X2.REV=	000077
T19SEX	061532	T208SS	065101	T39ETN	076501	WC.IRW=	000004	X2.SPA=	035400
T19SNP	061372	T23A	003144 G	T39ETS	076412	WC.IOT=	000100	X2.UNI=	000007
T19SRD	061330	T23B	003146 G	T39MCL	077414	WC.I1T=	000040	X2.WCF=	002000
T19SSR	057573	T3	025750 G	T39NBA	077035	WC.I5R=	000020	X3.DCK=	000010

Symbol table

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

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

Errors detected: 0

*** Assembler statistics

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

Elapsed time: 00:12:41.72
CVTS80,CVTS80,-SVC/ML,CVTS80