

B1

W wto V7
EQ 0001
USER DOCUMENTATION

MACRO M1200 20-MAR-84 08:26 PAGE 2

S

.REM_
IDENTIFICATION

PRODUCT ID: AC-T816A-MC
PRODUCT TITLE: CNTSBA0 TSV05 CTRL LT2
DECO/DEPO: 1.0
DEPARTMENT: ISS/DIAGNOSTIC SERVICES
DATE: APRIL 09, 1984

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) 1984 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 SBC-11/21, RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF A TSV05 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A SBC-11/21, SYSTEM (Q-BUS). THE PROGRAM PROVIDES ERROR MESSAGES WHICH IDENTIFY FAILING FUNCTIONS THAT AID IN THE REPAIR OF THE DEVICE. THIS DIAGNOSTIC CONSIST OF TWELVE TEST. TEST 1-9 ARE EXECUTED IN SEQUENCE. TEST 10-12 ARE STAND ALONE TEST WHICH ALLOW THE OPERATOR TO PERFORM SPECIFIC FUNCTIONAL TEST ON SCOPE LOGPS ON CERTAIN FUNCTIONS.

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

1.2 SYSTEM REQUIREMENTS

SBC-11/21, PROCESSOR AND MEMORY
CAUTION:DIAGNOSTIC REQUIRES 32K WORDS OF MEMORY
(28K USEABLE AND 4K RESERVED FOR I/O PAGE)
TSV05 MAGTAPE SUBSYSTEM (DRIVE AND CONTROLLER)
CONSOLE TERMINAL
PDP-11 DIAGNOSTIC SUPERVISOR (MSAAA.SYS VERSION 34 OR LATER)
PDP-11 DIAGNOSTIC LOADER/MONITOR (XXDP+)

1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

1. XXDP+ USERS GUIDE
2. TSV05 TRANSPORT SUBSYSTEM USER'S GUIDE
3. TSV05 TRANSPORT SUBSYSTEM TECHNICAL MANUAL
4. TSV05 TRANSPORT SUBSYSTEM INSTALLATION MANUAL

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

FUNCTIONAL SBC-11/21, CENTRAL PROCESSOR AND MEMORY
FUNCTIONAL CONSOLE TERMINAL
FUNCTIONAL STANDALONE DIAGNOSTIC SUPERVISOR
FUNCTIONAL DIAGNOSTIC LOADER/MONITOR (XXDP+)

1.5 ASSUMPTIONS

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

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

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

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

2.1.1 OPERATOR COMMANDS

THE TSV05 DIAGNOSTIC IS A SBC-11/21+ DIAGNOSTIC SUPERVISOR COMPATIBLE PROGRAM. ALL LOADING AND RUNTIME INSTRUCTIONS CAN BE REFERENCED IN THE XXDP+ USERS GUIDE. THE USER ENTRY IS IN QUOTES.

BOOT THE DIAGNOSTIC MEDIA

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

2.2 SWITCHES

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

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:DDDD	EXECUTE DDDDD PASSES (DDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED IN SECTION 2.3.
/EOP:DDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDD = 1 TO 64000)
/UNITS:LIST	TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED IN THE LIST. LIST EXAMPLE - /UNITS:0:5:10-12 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

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

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

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

	TESTS	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
ZFLAGS					
EXIT					

2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE

FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBR*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXE*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
PNT	PRINT TEST NUMBER AS TEST EXECUTES
BOE	"BELL" ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE
LOT	LOOP ON TEST

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

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

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

2.4 HARDWARE QUESTIONS

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

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

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

TSBA/TSDB = 176000, VECTOR = 224

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

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

UNIT 0

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

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

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

2.5 SOFTWARE QUESTIONS

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

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

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

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

2.6 EXTENDED P-TABLE DIALOGUE

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

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

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

* UNITS (D) ? 8<CR>

UNIT 1

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 0<CR>

Q-FACTOR (0) 0 ? 1<CR>

UNIT 2

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 1<CR>

Q-FACTOR (0) 1 ? 0<CR>

UNIT 3

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 2<CR>

Q-FACTOR (0) 0 ? <CR>

UNIT 4

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 3<CR>

Q-FACTOR (0) 0 ? <CR>

UNIT 5

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 4<CR>

Q-FACTOR (0) 0 ? <CR>

UNIT 6

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 5<CR>

Q-FACTOR (0) 0 ? <CR>

UNIT 7

CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 6<CR>

Q-FACTOR (0) 0 ? 1<CR>

UNIT 8

CSR ADDRESS (0) 160000<CR>

SUB-DEVICE # (0) ? 7<CR>

Q-FACTOR (0) 1 ? <CR>

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

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

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

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

♦ UNITS (D) ? 8<CR>

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

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

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

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

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

♦ UNITS (D) ? 8<CR>

UNIT 1
CSR ADDRESS (D) ? 160000<CR>
SUB-DEVICE ♦ (D) ? 0-7<CR>
Q-FACTOR (D) 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 "ST.RF"
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
CNTSB HRD ERR 01610 ON UNIT 00 TST 016 SUB 002 PC: 040624
FIFO STATUS (IN WORD 9) INCORRECT AFTER WRITE FIFO

TAPE BUS SIGNALS IN WORD #8: - DESIGNATOR <BIT #>
PARERR<15> IEOT <12> IFMK <9> IRDY<6> IRWD<2>
IRESV2<14> IIDENT<11> IHER <8> IONL<5> IFJY<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.

CNTSB HRD ERR 00159 ON UNIT 00 TST 001 SUB 005 PC: 026202
TSSR NOT CORRECT AFTER SPACE RECORDS COMMAND

TSSR = 100214

TSSR BITS SET: SC, SSR

TERMINATION CLASS CODE = UNRECOVERABLE ERROR

PACKET ADDRESS = 026420

PACKET WORD # = 140010

PACKET WORD # = 000010

PACKET WORD # = 000000

PACKET WORD # = 000024

ERROR MESSAGE EXAMPLE 3

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

CNTSB HRD 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 (SBC-11/21+)

DR>STA/FLA:PNT:HOE:UAM

UNITS (D) ? 1

UNIT 0

DEVICE ADDRESS (0) 176000 ? <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 FALCON + PROCESSOR.

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

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

Q.V.	2 MINS 15 SECONDS
DEFAULT	12 MINUTES

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

5.0 DEVICE INFORMATION TABLES

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

CHANGE HW (L) ?

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

UNIT 0

DEVICE ADDRESS (0) 176000 ? <ENTER THE ADDRESS OF THE
TSBA/TSDB REGISTER>

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

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

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

CHANGE SW (L) ?

6.0 TEST SUMMARIES

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

TEST DESCRIPTION:

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

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

THIS TEST VERIFIES THAT THE WRITE SUBSYSTEM MEMORY COMMAND WITH A 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 11/21+ SYSTEM WITH MORE THAN 128K WORDS OF MEMORY!

TEST 4: RAM EXERCISER TEST

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

TEST 5: EXTENDED FEATURES SWITCH AND TIMERS A,B

TEST DESCRIPTION:

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

TEST 6: FIFO EXERCISER

TEST DESCRIPTION:

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

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

TEST DESCRIPTION:

WRITE TO TSSR REGISTER TO SOFT INITIALIZE THE CONTROLLER
DO WRITE CHARACTERISTICS TO CHECK FOR EXTENDED FEATURES SWITCH
IF EXTENDED FEATURES HARDWARE SWITCH CLEAR THEN:
DO WRITE SUBSYSTEM WRITE MISCELLANEOUS TO SET EXTENDED FEATURES.
DO WRITE CHARACTERISTICS TO SELECT RESERVED INIT 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 TSDB HIGH BYTE WRITE ACCESS
- 5 TSDB LOW BYTE WRITE ACCESS
- 6 TSDB MAINTENANCE-MODE WORD WRITE ACCESS
- 7 TSDBX (TSSR HIGH BYTE) WRITE ACCESS
(EXTENDED FEATURES SWITCH MUST BE ON
TO USE SELECTION CODE 7)
- 8 EXIT (RETURN TO SUPERVISOR)

FOR SCOPE LOOPS THAT WRITE INTO REGISTERS, THE PROGRAM PROMPTS
THE OPERATOR FOR THE DATA TO BE WRITTEN, LIMITS ON THE DATA 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).

CVTSBBO => CNTSBAO

JAKI BERG

9-APR-1984

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

F2

SEQ 0018

```

2      .TITLE  TSV2 - PROGRAM HEADER
3      .SBTTL  PROGRAM HEADER
4
10     .MCALL  SVC
11 000000 SVC ; INITIALIZE SUPERVISOR MACROS
12     .ENABLE LC
13     .NLIST  BEX,CND
19 000000 .ENABL ABS,AMA
20     .-2000
21 002000 BGNMOD TSV2
    002000
22
23
24 ;++
25 ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
26 ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
27 ;--
28 002000 POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
29 002000 HEADER CNTSB,B,0,655..0
    002000
    002000 103 L$NAME:: ;DIAGNOSTIC NAME
    002001 116 .ASCII /C/
    002002 124 .ASCII /N/
    002003 123 .ASCII /T/
    002004 102 .ASCII /S/
    002005 000 .ASCII /B/
    002006 000 .BYTE 0
    002007 000 .BYTE 0
    002010 .BYTE 0
    002010 102 L$REV:: ;REVISION LEVEL
    002011 .ASCII /B/
    002011 060 L$DEPO:: ;0
    002012 .ASCII /O/
    002012 000000 L$UNIT:: ;NUMBER OF UNITS
    002014 .WORD 0
    002014 001217 L$TIML:: ;LONGEST TEST TIME
    002016 .WORD 655.
    002016 101412 L$HPCP:: ;PTR. TO H.W. QUES.
    002020 .WORD L$HARD
    002020 101544 L$SPCP:: ;PTR. TO S.W. QUES.
    002022 .WORD L$SOFT
    002022 002156 L$HPTP:: ;PTR. TO DEF. H.W. PTABLE
    002024 .WORD L$HW
    002024 002166 L$SPTP:: ;PTR. TO S.W. PTABLE
    002026 .WORD L$SW
    002026 102404 L$LADP:: ;DIAG. END ADDRESS
    002030 .WORD L$LAST
    002030 000000 L$STA:: ;RESERVED FOR APT STATS
    002032 .WORD 0
    002032 000000 L$CO::
    002034 .WORD 0
    002034 000000 L$DTYP:: ;DIAGNOSTIC TYPE
    002036 .WORD 0
    002036 000000 L$APT:: ;APT EXPANSION
    002040 .WORD 0
    002040 002124 L$DTP:: ;PTR. TO DISPATCH TABLE
    002042 .WORD L$DISPATCH
    L$PRIO:: ;DIAGNOSTIC RUN PRIORITY

```

G2

TSV2 - PROGRAM HEADER MACRO M1200 20-MAR-84 08:26 PAGE 7-1

SEQ 0019

PROGRAM HEADER

002042	000000			
002044		L\$ENVI::	.WORD 0	;FLAGS DESCRIBE HOW IT WAS SETUP
002044	000000		.WORD 0	
002046		L\$EXP1::	.WORD 0	;EXPANSION WORD
002046	000000		.WORD 0	
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::		
002056	000000		.WORD 0	
002060		L\$DEVP::		; POINTER TO DEVICE TYPE LIST
002060	003402		.WORD L\$DVTYP	
002062		L\$REPP::		;PTR. TO REPORT CODE
002062	022712		.WORD L\$RPT	
002064		L\$EXP4::		
002064	000000		.WORD 0	
002066		L\$EXP5::		
002066	000000		.WORD 0	
002070		L\$AUT::		;PTR. TO ADD UNIT CODE
002070	022400		.WORD L\$AU	
002072		L\$DUT::		;PTR. TO DROP UNIT CODE
002072	022476		.WORD L\$DU	
002074		L\$LUN::		;LUN FOR EXERCISERS TO FILL
002074	000000		.WORD 0	
002076		L\$DESP::		;POINTER TO DIAG. DESCRIPTION
002076	003410		.WORD L\$DESC	
002100		L\$LOAD::		;GENERATE SPECIAL AUTOLOAD EMT
002100	104035		EMT E\$LOAD	
002102		L\$ETP::		;POINTER TO ERRITBL
002102	000000		.WORD 0	
002104		L\$ICP::		;PTR. TO INIT CODE
002104	021556		.WORD L\$INIT	
002106		L\$CCP::		;PTR. TO CLEAN-UP CODE
002106	022664		.WORD L\$CLEAN	
002110		L\$ACP::		;PTR. TO AUTO CODE
002110	022604		.WORD L\$AUTO	
002112		L\$PRT::		;PTR. TO PROTECT TABLE
002112	021546		.WORD L\$PROT	
002114		L\$TEST::		;TEST NUMBER
002114	000000		.WORD 0	
002116		L\$DLY::		;DELAY COUNT
002116	000000		.WORD 0	
002120		L\$HIME::		;PTR. TO HIGH MEM
002120	000000		.WORD 0	

H2

TSV2 - PROGRAM HEADER MACRO M1200 20-MAR-84 08:26 PAGE 8

SEQ 0020

DISPATCH TABLE

31
32
33
34
35
36
37
38

002122
002122 000014
002124
002124 023474
002126 024456
002130 026450
002132 032044
002134 034634
002136 040426
002140 050540
002142 052020
002144 062646
002146 066716
002150 074560
002152 077732

```
.SBTTL DISPATCH TABLE

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

DISPATCH 12
.WORD 12
L$DISPATCH::
.WORD T1
.WORD T2
.WORD T3
.WORD T4
.WORD T5
.WORD T6
.WORD T7
.WORD T8
.WORD T9
.WORD T10
.WORD T11
.WORD T12
```

DEFAULT HARDWARE P-TABLE

```

40          .SBTTL  DEFAULT HARDWARE P-TABLE
41
42          ;
43          ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
44          ; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
45          ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
46          ;
47          002154      BGNHW      DFPTBL      ;DEFAULT HARD-P-TABLE
                     002154      000003      .WORD      L10000-L$HW/2
                     002156
                     002156      L$HW::
                     DFPTBL::
48
49          002156      176000      .WORD      176000      ; 1ST (OF 2) REGISTERS.
50          002160      000224      .WORD      224        ; INTERRUPT VECTOR
51          002162      000200      .WORD      PRI04       ; INTERRUPT PRIORITY.
52          002164
                     002164      ENDDW
                     L10000:

```

SOFTWARE P-TABLE

```

54          .SBTTL  SOFTWARE P-TABLE
55
56          ;++
57          ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
58          ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
59          ;--
60 002164          BGNSW  SFPTBL
   002164 000004      .WORD  L10001-L$SW/2
   002166
   002166  L$SW::
   002166  SFPTBL::
61
62 002166 000000      TRANSTST::      .WORD  0          ; ENABLE TEST OF TRANSPORT(S) IF =1
63 002170 000000      NOITS::          .WORD  0          ; INHIBIT ITERATION OPTION.
64                                     ; ... 0 = ITERATE.
65                                     ; ... NZ = INHIBIT ITERATE.
66 002172 000017      LERRMAX::          .WORD  15.        ; LOCAL (PER TEST) ERROR LIMIT
67 002174 000310      GERRMAX::          .WORD  200.       ; GLOBAL (PER UNIT) ERROR LIMIT
68 002176          ENDSW
   002176  L10001:
69
70 002176          ENDMOD

```

SOFTWARE P-TABLE

```

7          .TITLE  TSV3 - GLOBAL AREAS
8          .SBTTL  GLOBAL EQUATES SECTION
13
19
20 002176   BGNMOD  TSV3
    002176   TSV3::
21
22          .SBTTL  GLOBAL EQUATES SECTION
23
24          ;++
25          ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
26          ; ARE USED IN MORE THAN ONE TEST.
27          ;--
28
32 002176   EQUALS          ; GET STANDARD EQUATES.
          ;
          ; BIT DIFINITIONS
          ;
          100000   BIT15== 100000
          040000   BIT14== 40000
          020000   BIT13== 20000
          010000   BIT12== 10000
          004000   BIT11== 4000
          002000   BIT10== 2000
          001000   BIT09== 1000
          000400   BIT08== 400
          000200   BIT07== 200
          000100   BIT06== 100
          000040   BIT05== 40
          000020   BIT04== 20
          000010   BIT03== 10
          000004   BIT02== 4
          000002   BIT01== 2
          000001   BIT00== 1
          ;
          001000   BIT9==  BIT09
          000400   BIT8==  BIT08
          000200   BIT7==  BIT07
          000100   BIT6==  BIT06
          000040   BIT5==  BIT05
          000020   BIT4==  BIT04
          000010   BIT3==  BIT03
          000004   BIT2==  BIT02
          000002   BIT1==  BIT01
          000001   BIT0==  BIT00
          ;
          ; EVENT FLAG DEFINITIONS
          ; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
          ;
          ;
          000040   EF.START==      32.          ; BIT POSITION IN SECOND STATUS WORD
          000037   EF.RESTART==    31.          ; (100000) START COMMAND WAS ISSUED
          000036   EF.CONTINUE==   30.          ; (040000) RESTART COMMAND WAS ISSUED
          000035   EF.NEW==        29.          ; (020000) CONTINUE COMMAND WAS ISSUED
          000034   EF.PWR==        28.          ; (010000) A NEW PASS HAS BEEN STARTED
          ;                                     ; (004000) A POWER-FAIL/POWER-UP OCCURRED
          ;
          ;

```


MEMORY MANAGEMENT DEFINITIONS

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

MEMORY MANAGEMENT DEFINITIONS

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

B3

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 13

SEQ 0027

TSV05 REGISTER AND PACKET DEFINITIONS

```

39          .SDITL  TSV05 REGISTER AND PACKET DEFINITIONS
40
41          |
42          | SOME GENERAL EQUATES.
43          |
44
45          000004      ERRVEC==      4          | POINTER TO ERROR VECTOR FOR BUS TIME OUT.
46          000060      TTIVEC==     60          | INTERRUPT VECTOR FOR CONSOLE INPUT
47          177560      TTICSR==    177560        | BUS ADDRESS OF CONSOLE INPUT
48          177562      TTIBFR==    177562        | CONSOLE INPUT DATA BUFFER
49          177520      BDVPCR==    177520        | BDV11 PAGE CONTROL REGISTER
50
51          |*
52          |BIT DEFINITIONS FOR TSSR REGISTER
53          |*
54
55          100000      SC=      BIT15          |SPECIAL CONDITION
56          040000      BIE=     BIT14          |BUS INTERFACE ERROR
57          020000      SCE=     BIT13          |SANITY CHECK ERROR
58          010000      RMR=     BIT12          |MODIFICATION REFUSED
59          004000      NXM=     BIT11          |NONEXISTANT MEMORY ERROR
60          002000      NBA=     BIT10          |NEED BUFFER ADDRESS
61          001400      HIAADR=  BIT9:BIT8      |EXTENDED ADDRESS BITS
62          000200      SSR=     BIT7           |SUB SYSTEM READY
63          000100      OFL=     BIT6           |OFF LINE BIT
64          000060      FATERR=  BIT4:BIT5      |FATAL TERMINATION ERROR CODES
65          000016      TERCLS=  BIT3:BIT2:BIT1 |TERMINATION CODES
66
67          |*
68          |
69          |BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
70          |(XST0)
71          |
72          |*
73
74          100000      XSOTMK=  BIT15          |TAPE MARK DETECTED
75          040000      XSORLS=  BIT14          |RECORD LENGTH SHORT
76          020000      XSOLET=  BIT13          |LOGICAL END OF TAPE
77          010000      XSORLL=  BIT12          |RECORD LENGTH LONG
78          004000      XSOWLE=  BIT11          |WRITE LOCK ERROR
79          002000      XSONEF=  BIT10          |NON EXECUTABLE FUNCTION
80          001000      XS0ILC=  BIT9           |ILLEGAL COMMAND
81          000400      XS0ILA=  BIT8           |ILLEGAL ADDRESS
82          000200      XSOMOT=  BIT7           |TAPE IN MOTION
83          000100      XS0ONL=  BIT6           |TRANSPORT ON LINE
84          000040      XS0IE=   BIT5           |INTERRUPT ENABLE
85          000020      XS0VCK=  BIT4           |VOLUME CHECK BIT
86          000010      XS0PED=  BIT3           |PHASE ENCODED DRIVE
87          000004      XS0WLK=  BIT2           |WRITE LOCKED
88          000002      XS0BOT=  BIT1           |BEGINNING OF TAPE
89          000001      XS0EOT=  BIT0           |END OF TAPE
90
91          |*
92          |BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
93          |(XST1)
94          |
95          100000      X1ULT =  BIT15          |DATA LATE

```

C3

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 13-1

SEQ 0028

TSV05 REGISTER AND PACKET DEFINITIONS

```

96      040000      X1.SPARE = BIT14      ;NOT USED
97      020000      X1.COR  = BIT13      ;CORRECTABLE DATA ERROR
98      017375      X1.MBZ  = BIT12+BIT11+BIT10+BIT9+BIT7+BIT6+BIT5+BIT4+BIT3+BIT2+BIT0 ;ALWAYS 0
99      000400      X1.RBP  = BIT8       ;READ BUS PARITY ERROR
100     000002      X1.UNC  = BIT1       ;UNCORRECTABLE DATA OR HARD ERROR
101
102      ;
103      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
104      ;(XST2)
105      ;
106     100000      X2.OPM  = BIT15      ;OPERATION IN PROGRESS (TAPE MOVING)
107     040000      X2.RCE  = BIT14      ;RAM CHECKSUM ERROR
108     035400      X2.SPARE = BIT13+BIT12+BIT11+BIT9+BIT8 ;NOT USED BY TSV05 (ALWAYS=0)
109     002000      X2.WCF  = BIT10      ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
110     000200      X2.EXTF = BIT7       ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
111     000100      X2.BUFE = BIT6       ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
112     000077      X2.REV  = 000077    ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
113     000007      X2.UNIT = BIT2+BIT1+BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.
114
115      ;
116      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
117      ;(XST3)
118      ;
119     177400      X3.MDE  = 177400    ;MICRO-DIAGNOSTIC ERROR CODE
120     000200      X3.SPARE = BIT7      ;NOT USED BY TSV05
121     000100      X3.OPI  = BIT6      ;OPERATION INCOMPLETE
122     000040      X3.REV  = BIT5      ;REVERSE
123     000020      X3.TRF  = BIT4      ;TRANSPORT RESPONSE FAILURE
124     000010      X3.DCK  = BIT3      ;DENSITY CHECK
125     000006      X3.MBZ  = BIT2+BIT1 ;NOT USED ALWAYS 0
126     000001      X3.RIB  = BIT0      ;REVERSE INTO BOT
127
128      ;
129      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
130      ;(XST4)
131      ;
132     100000      X4.HSP  = BIT15      ;HIGH SPEED
133     040000      X4.RCE  = BIT14      ;RETRY COUNT EXCEEDED
134     020000      X4.TSM  = BIT13      ;TRANSPORT SPECIAL MCDE
135     017400      X4.MBZ  = BIT12+BIT11+BIT10+BIT9+BIT8 ;NOT USED ALWAYS 0
136     000377      X4.WRC  = 000377    ;WRITE RETRY COUNT FIELD
137
138      ;
139      ;
140      ;TSSR TERMINATION CODES (BIT 0-2)
141      ;
142      ;
143      ;
144     000006      TSREJ = 3+2          ;COMMAND REJECTED
145     000006      UNREC = 6           ;UNRECOVERABLE ERROR
146
147      ;
148      ;
149      ;DEVICE REGISTER OFFSETS
150      ;
151      ;
152      ;

```

D3

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 13-2

SEQ 0029

TSV05 REGISTER AND PACKET DEFINITIONS

```

153      000000      TSBA== 0
154      000000      TSDB== 0      ;TSDB/TSBA REGISTER
155      000001      TSBAH== 1
156      000001      TSDBH== 1      ;TSDB/TSBA REGISTER HIGH BYTE
157      000002      TSSR== 2      ;TSSR REGISTER
158      000003      TSSRH== 3      ;TSSR REGISTER HIGH BYTE
159
160      ;
161      ; TSDB ADDRESS BIT DEFINITIONS
162      ;
163      000003      A1716 = BIT1+BIT0      ;ADDRESS BITS 17:16 ARE IN 1:0
164
165      ;
166      ; COMMAND DEFINITIONS
167      ;
168      000017      P.GETSTAT = 17      ;GET STATUS
169      000013      P.INIT = 13      ;INITIALIZE
170      000012      P.CONTROL = 12      ;CONTROL COMMANDS
171      000011      P.FORMAT = 11      ;FORMAT
172      000010      P.POSITION = 10      ;POSITION
173      000006      P.WRTSUB = 6      ;SUBSYSTEM WRITE
174      000005      P.WRITE = 5      ;WRITE
175      000004      P.WRTCHAR = 4      ;WRITE CHARACTERISTICS
176      000001      P.READ = 1      ;READ
177
178      ;
179      ; COMMAND PACKET HEADER WORD BIT DEFINITIONS
180      ;
181      100000      P.ACK = BIT15      ;BUFFER AVAIL FOR CONTROLLER
182      040000      P.CVC = BIT14      ;CLEAR VOLUME CHECK
183      020000      P.OPP = BIT13      ;REVERSE SEQUENCE OF DATA BITS
184      010000      P.SWB = BIT12      ;SWAP BYTES IN MEMORY
185      007400      P.MODE = BIT11:BIT10:BIT9:BIT8 ;EXTENDED COMMAND MODE FIELD
186      000200      P.IE = BIT7      ;INTERRUPT ENABLE
187      000140      P.FMT = BIT6:BIT5      ;PACKET HEADER TYPE (ALWAYS=0)
188      000037      P.CMD = 37      ;MAJOR COMMAND FIELD
189
190      ;
191      ; CONTROL COMMAND MODE CODES
192      ;
193      000000      PC.RELEASE = 0*256.      ;RELEASE BUFFER
194      000400      PC.REWIND = 1*256.      ;REWIND
195      001000      PC.NOOP = 2*256.      ;NO-OP
196      002000      PC.IEREW = 4*256.      ;REWIND IMMEDIATE INTERRUPT
197      002400      PC.ERASE = 5*256.      ;SECURITY ERASE
198
199      ;
200      ; CONTROLLER RAM DEFINITIONS
201      ;
202      000167      RMCHBEG = 167      ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
203      000200      RMCHEND = 200      ;CHARACTERISTICS IO DATA END RAM ADDRESS
204      000201      RMPKTBEG = 201      ;COMMAND PACKET BEGIN RAM ADDRESS
205      000210      RMPKTEND = 210      ;COMMAND PACKET END RAM ADDRESS
206      000215      RMMSGBEG = 215      ;MESSAGE BUFFER BEGIN RAM ADDRESS
207      000234      RMMSGEND = 234      ;MESSAGE BUFFER END RAM ADDRESS
208
209      ;
210      ; REGISTER DEFINITIONS IN THE MESSAGE BUFFER

```

E3

SEQ 0030

TSV05 REGISTER AND PACKET DEFINITIONS

```

210      |
211      |
212      |
213      000006      XST0== 6      ;EXTENDED STATUS REGISTER 0 (WORD 4)
214      000010      XST1== 8      ;EXTENDED STATUS REGISTER 1 (WORD 5)
215      000012      XST2== 10     ;EXTENDED STATUS REGISTER 2 (WORD 6)
216      000014      XST3== 12     ;EXTENDED STATUS REGISTER 3 (WORD 7)
217      000016      XST4== 14     ;EXTENDED STATUS REGISTER 4 (WORD 8)
218
219      |
220      |
221      |OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
222      |
223      |
224
225      000002      PKLOW  = 2      LOW ORDER CHARACTERISTIC DATA POINTER
226      000004      PKHI   = 4      ;HIGH ORDER CHARACTERISTIC DATA POINTER
227      000006      PKBCNT = 6      ;NUMBER OF BYTES IN DATA PACKET
228
229      000010      EXBCNT=10      ;NUMBER OF BYTES IN EXTENDED DATA PACKET
230
231      |
232      |DATA PACKET OFFSETS FOR WRITE SUBSYSTEM COMMAND
233      |
234      000000      BSELO  = 0      ;BYTE 0
235      000001      BSEL1  = 1      ;BYTE 1
236      000002      SEL2   = 2      ;WORD 2
237      000004      SELDATA = 4      ;WORD 3
238
239      |
240      |BSELO SELECT CODES FOR WRITE SUBSYSTEM COMMAND
241      |
242      000000      PW.NOP      = 0      ;NO-OP
243      000001      PW.RDRAM    = 1      ;READ RAM
244      000002      PW.WTRAM    = 2      ;WRITE RAM
245      000003      PW.RFIFO    = 3      ;READ FIFO
246      000004      PW.WFIFO    = 4      ;WRITE FIFO
247      000005      PW.RDSTAT   = 5      ;READ STATUS
248      000006      PW.WCTL     = 6      ;WRITE TAPE CONTROL
249      000007      PW.WFMT     = 7      ;WRITE TAPE FORMAT
250      000010      PW.WMISC    = 10     ;WRITE MISCELLANEOUS
251      000011      PW.WNPR     = 11     ;WRITE NPR CONTROL
252      000020      PW.D22      = 20     ;DO MICROTEST 22
253      000021      PW.D11      = 21     ;DO MICROTEST 11
254      000022      PW.D13      = 22     ;DO MICROTEST 13
255      000023      PW.NO1311   = 23     ;DISABLE MICROTEST 11 AND 13
256      000024      PW.RDEXT     = 24     ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS)
257
258      |
259      |BSEL1 CODES FOR WRITE TAPE CONTROL
260      |
261      000200      WC.IFAD      = BIT7    ;IFAD - FORMATTER ADDRESS
262      000100      WC.IOTAD     = BIT6    ;ITADO - TRANSPORT ADDRESS BIT 0
263      000040      WC.I1TAD     = BIT5    ;ITAD1 - TRANSPORT ADDRESS BIT 1
264      000020      WC.ISRESV    = BIT4    ;IRESVS - RESERVED #5
265      000010      WC.IREW      = BIT3    ;IREW - REWIND
266      000004      WC.IRWU      = BIT2    ;IRWU - REWIND AND UNLOAD

```

F3

SEQ 0031

TSV05 REGISTER AND PACKET DEFINITIONS

```

267      000002      WC.IFEN      = BIT1      ;IFEN - FORMATTER ENABLE
268      000001      WC.IGO       = BIT0      ;GO
269
270      ;
271      ;BSEL1 CODES FOR WRITE FORMAT
272      ;
273      000200      WF.IHISP      = BIT7      ;IHISP - HIGH SPEED
274      000100      WF.IWRT      = BIT6      ;IWRT - WRITE
275      000040      WF.IREV      = BIT5      ;IREV - REVERSE
276      000020      WF.IWFM      = BIT4      ;IWFM - WRITE FILE MARK
277      000010      WF.IEDIT     = BIT3      ;IEDIT - EDIT
278      000004      WF.IERASE    = BIT2      ;IERASE - ERASE
279      000002      WF.I3RESV    = BIT1      ;IRESV3 - RESERVED #3
280      000001      WF.I4RESV    = BIT0      ;IRESV4 - RESERVED #4
281
282      ;
283      ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
284      ;
285      000200      MS.EXT        = BIT7      ;INVERT SENSE OF EXTENDED FEATURES SWITCH
286      000020      MS.RSFIFO     = BIT4      ;RESET FIFO AND INPUT PARITY ERRORR
287      000010      MS.RSTAPE     = BIT3      ;RESET TAPE STATUS IN 2 FLIF-FLOPS
288      000006      MS.ATTN       = BIT2:BIT1 ;ATTENTION TRIGGER FIELD
289      000001      MS.RSD        = BIT0      ;RESET TIMER A,B THEN DELAY TIMES IN SEL2
290
291      ; MS.ATTN SUBCODES
292      ;
293      000000      MSA.NOP = 0*2      ;NO-OP (NOTHING TRIGGERED)
294      000002      MSA.VOL = 1*2      ;SIMULATE ON-LINE/OFF-LINE TRANSITION
295      000004      MSA.NRAM = 2*2     ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
296      000006      MSA.FRAME = 3*2    ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
297
298      ;
299      ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
300      ;
301      000200      NP.IR         = BIT7      ;INTERRUPT REQUEST (0-1 TRANSITION)
302      000100      NP.OUT        = BIT6      ;TAPE DATA DIRECTION OUT (0= IN)
303      000040      NP.LOOP      = BIT5      ;ENABLE TRANSPORT LOOPBACK
304      000020      NP.WRP        = BIT4      ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
305
306      ;
307      ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
308      ;
309      000200      S2.DIM         = BIT7      ;WORD #9 BYTE 2 DATA IN MISS
310      000100      S2.ILW        = BIT6      ;ILW H
311      000040      S2.OUTRDY     = BIT5      ;OUT RDY H
312      000020      S2.INRDY      = BIT4      ;IN RDY H
313      000010      S2.ATIMR      = BIT3      ;TIMER A FLAG H
314      000004      S2.BTIMR      = BIT2      ;TIMER B FLAG H
315      000003      S2.UNDEF      = BIT1:BIT0 ; (UNDEFINED)
316      100000      S1.PARIN      = BIT15     ;WORD #8 BYTE 1 PARIN H
317      040000      S1.I2RESV     = BIT14     ;IRESV2
318      020000      S1.I1RESV     = BIT13     ;IRESV1
319      010000      S1.IEOT       = BIT12     ;IEOT L
320      004000      S1.IIDENT     = BIT11     ;IIDENT H
321      002000      S1.ICER       = BIT10     ;ICER H
322      001000      S1.IFMK       = BIT9      ;IFMK H
323      000400      S1.IHER       = BIT8      ;IHER H
324      000200      S0.ISPEED     = BIT7      ;WORD #8 BYTE 0 ISPEED H

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

324      000100      SO.IRDY      ▪ BIT6      ;      IRDY L
325      000040      SO.IONL      ▪ BIT5      ;      IONL L
326      000020      SO.ILDP      ▪ BIT4      ;      ILDP L
327      000010      SO.IDRY      ▪ BIT3      ;      IDBY L
328      000004      SO.IRWD      ▪ BIT2      ;      IRWD L
329      000002      SO.IFBY      ▪ BIT1      ;      IFBY L
330      000001      SO.IFPT      ▪ BIT0      ;      IFPT L
331                                     .SBITL  SPECIAL MACROS AND OPDEFS.
332
333      ;+
334      ;SAVE GENERAL REGS 1 TO 5
335      ;-
336
337      .MACRO  SAVREG
338      JSR     R5,REGSAV
339      .ENDM
340
341      ;+
342      ; MACRO TO FORCE AN ERROR
343      ;-
344      .MACRO  FORCERROR      TAG,NOTSSR
345      .NLIST
346      .IIF NDF LISTALL, .NLIST
347      .LIST
348      .IF B NOTSSR
349      MOV     TSSR(R5),R1      ;READ TSSR
350      .ENDC
351      MOV     FORCER,FORCER    ;IS FORCER SET? (LEAVE C BIT ALONE)
352      BNE     TAG              ;BR IF YES
353      .NLIST
354      .IIF NDF LISTALL, .LIST
355      .LIST
356      .ENDM
357
358      ;+
359      ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
360      ; WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
361      ; SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
362      ; FORCER TO 177777
363      ; TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
364      ;-
365      .MACRO  FORCEEXIT      TAG
366      .NLIST
367      .IIF NDF LISTALL, .NLIST
368      .LIST
369      MOV     FORCER,FORCER    ;IS FORCER NEGATIVE?
370      BMI     TAG              ;BR IF YES
371      .NLIST
372      .IIF NDF LISTALL, .LIST
373      .LIST
374      .ENDM
375      ;+
376      ; MACRO TO INCREMENT ERROR COUNTS
377      ;-
378      .MACRO  NEXT.ERRNO
379      .NLIST
380      ;.IIF NDF LISTALL, .NLIST

```

SPECIAL MACROS AND OPDEFS.

```

381          ERRNO=ERRNO+1
382      ***.IIF NDF LISTALL, .LIST
383      .LIST
384      .ENDM
385
386      ;+
387      ;MACRO TO PERFORM XOR
388      ;-
389
390          .MACRO XOR      A,B
391          MOV      A,-(SP)
392          BIC      B,(SP)
393          BIC      A,B
394          BIS      (SP)+,B
395          .ENDM
396
397          000000          EN=0          ; INITIALIZE ERROR NUMBER
398          .SBTTL  FORCER  - FORCE ERROR FLAG
399
400      ;
401      ; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
402      ; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
403      ;
404
405      002176  000000      FORCER::      0          ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED -
406          ; - BY THE MACRO "IFERROR"). AN ERROR NEED NOT -
407          ; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.
408          .SBTTL  GLOBAL DATA SECTION
409
410      ;++
411      ;THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
412      ;IN MORE THAN ONE TEST.
413      ;--
414
415      ;
416      ;THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
417      ;SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.
418      ;
419      002200  000000      EPRTSW::      .WORD      0          ;PRINT SWITCH
420      002202  000000      UNITN::      .WORD      0          ;UNIT # UNDER TEST.
421      002204  000000      QVP::      .WORD      0          ;QUICK VERIFY FLAG.
422      002206  000000      CSRADDR::    .WORD      0          ;ADDRESS OF CSR FOR CURRENT DEVICE
423      002210  000224      IVEC::      .WORD      224        ;INTERRUPT VECTOR
424      002212  000200      IPRI::      .WORD      PRI04      ;INTERRUPT PRIORITY.
425      002214  000000      TSTCNT::    .WORD      0          ;NUMBER OF TESTS RUN IN THIS PASS
426      002216  000000      LOOPCNT::   .WORD      0          ;REMAINING ITERATION COUNT FOR TEST
427      002220  000000      DEVCNT::    .WORD      0          ;NUMBER OF DEVICE UNDER TEST
428      002222  000000      FATFLG::    .WORD      0          ;SET IF FATAL ERROR IS DETECTED IN TEST
429      002224  000000      INTRECV::   .WORD      0          ;SET IF TAPE INTERRUPT WAS RECEIVED
430      002226  000000      EXTFEA::    .WORD      0          ;EXTENDED FEATURES SOFTWARE SW 0-OFF,1-ON
431      002230  000000      BENBSW::    .WORD      0          ;BUFFER ENABLE SWITCH SW 0-OFF,1-ON
432      002232  000000      EXPD::      .WORD      0          ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
433      002234  000000      RECV::      .WORD      0          ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
434      002236  000000      ERRHI::     .WORD      0          ;HIGH ADDRESS MEMORY ERROR
435      002240  000000      ERRLO::     .WORD      0          ;LOW ADDRESS MEMORY ERROR
436      002242          RAMDATA::      .BLKW      16.        ;DATA READ FROM RAM PACKET OF MESSAGE BUF AREA
437      002302  000000      RAMSIZ::    .WORD      0          ;RAM DATA SIZE FOR PRAMPKT ROUTINE

```

GLOBAL DATA SECTION

```

438 002304 000000      RCVHIADD::      .WORD    0      ;RECEIVED BUFFER HIGH ADDRESS
439 002306 000000      RCVLOADD::      .WORD    0      ;RECEIVED BUFFER LOW ADDRESS
440 002310 000000      COUNT::          .WORD    0      ;TEST COUNT PATTERN
441 002312 000000      DATA::          .WORD    0      ;TEST DATA
442 002314 000000      TSTFLAG::        .WORD    0      ;TEST FLAG WORD
443 002316 000000      TSTPTR::         .WORD    0      ;TSTBLK POINTER
444 002320 000000      PRMNO::           .WORD    0      ;PRINT ROUTINE TEMP
445 002322          EXPMSG::             .BLKB   100.    ;EXPECTED MESSAGE BUFFER DATA
446 002466          RECMMSG::            .BLKB   100.    ;RECEIVED MESSAGE BUFFER DATA
447 002632          TMPBFR::            .BLKB   80.     ;TEMPORARY STORAGE FOR PRINT
448                                     .SBTTL  TSTBLK  - TEST DATA TABLE
449
450
451
452      ;
453      ;THIS TABLE CONTAINS TEST DATA USED IN SEVERAL TESTS
454      ;
455      ;IN SEQUENCE THE DATA IS:
456      ;
457      ;      ALL ZEROS
458      ;      ALL ONES
459      ;      WALKING ONES
460      ;      WALKING ZEROS
461      ;      ALTERNATING ONES AND ZEROS
462      ;
463      ;-
464
464 002752          TSTBLK::
465 002752 000000      .WORD    0      ;ALL ZEROS
466 002754 177777      .WORD   177777    ;ALL ONES
467 002756 000001      .WORD   BIT0     ;DATA FOR WALKING ONES
468 002760 000002      .WORD   BIT1
469 002762 000004      .WORD   BIT2
470 002764 000010      .WORD   BIT3
471 002766 000020      .WORD   BIT4
472 002770 000040      .WORD   BIT5
473 002772 000100      .WORD   BIT6
474 002774 000200      .WORD   BIT7
475 002776 000400      .WORD   BIT8
476 003000 001000      .WORD   BIT9
477 003002 002000      .WORD  BIT10
478 003004 004000      .WORD  BIT11
479 003006 010000      .WORD  BIT12
480 003010 020000      .WORD  BIT13
481 003012 040000      .WORD  BIT14
482 003014 100000      .WORD  BIT15
483 003016 177776      .WORD  +CBIT0    ;DATA FOR WALKING ZEROS
484 003020 177775      .WORD  +CBIT1
485 003022 177773      .WORD  +CBIT2
486 003024 177767      .WORD  +CBIT3
487 003026 177757      .WORD  +CBIT4
488 003030 177737      .WORD  +CBIT5
489 003032 177677      .WORD  +CBIT6
490 003034 177577      .WORD  +CBIT7
491 003036 177377      .WORD  +CBIT8
492 003040 176777      .WORD  +CBIT9
493 003042 175777      .WORD  +CBIT10
494 003044 173777      .WORD  +CBIT11

```


TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 13-9

K3

SEQ 0036

GLOBAL ENVIRONMENT STORAGE

552 003376 000000

ERTABE: .WORD 0

553

554 003400 000000

SKIPT: .WORD 0

;1=SKIP SUBTEST 0=NO SKIP OF SUBTEST

GLOBAL TEXT MESSAGES

```

556                      .SBTTL  GLOBAL TEXT MESSAGES
557                      ;++
558                      ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
559                      ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
560                      ; MORE THAN ONE TEST.
561                      ;--
562
563                      ;+
564                      ; NAMES OF DEVICES SUPPORTED
565                      ;-
566
567 003402                DEVTP  <TSV05>
003402                L$DVTYP:
003402                .ASCIZ  *TSV05*
                    .EVEN
568
576                      ;+
577                      ; TEST DESCRIPTION
578                      ;-
579 003410                DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****>
003410                L$DESC:
003410                .ASCIZ  /**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****/
                    .EVEN
580
594
595                      ;+
596                      ; BIT TO ASCII CONVERSION FOR TSSR REGISTER
597                      ;-
598
599 003502 003542 003545 003551 TSSRBIT:
000 003522 003603 003607 003613 .WORD  1$,2$,3$,4$,5$,6$,7$,8$
601 003542      123      103      000 1$: .ASCIZ  'SC'
602 003545      102      111      105 2$: .ASCIZ  'BIE'
603 003551      123      103      105 3$: .ASCIZ  'SCE'
604 003555      122      115      122 4$: .ASCIZ  'RMR'
605 003561      116      130      115 5$: .ASCIZ  'NXM'
606 003565      116      102      101 6$: .ASCIZ  'NBA'
607 003571      102      111      124 7$: .ASCIZ  'BIT9'
608 003576      102      111      124 8$: .ASCIZ  'BIT8'
609 003603      123      123      122 9$: .ASCIZ  'SSR'
610 003607      117      106      114 10$: .ASCIZ  'OFL'
611 003613      102      111      124 11$: .ASCIZ  'BIT5'
612 003620      102      111      124 12$: .ASCIZ  'BIT4'
613 003625      102      111      124 13$: .ASCIZ  'BIT3'
614 003632      102      111      124 14$: .ASCIZ  'BIT2'
615 003637      102      111      124 15$: .ASCIZ  'BIT1'
616 003644      102      111      124 16$: .ASCIZ  'BIT0'
617                      .EVEN
618 003652      124      123      123 SFIERR: .ASCIZ  'TSSR ERROR AFTER SOFT INIT'
619 003705      124      123      123 SFHERR: .ASCIZ  'TSSR ERROR AFTER BUS RESET'
620 003740      040      040      116 NXR: .ASCIZ  / NON-EXISTANT DEVICE REGISTER/
621 003777      045      101      040 NXR: .ASCIZ  /#A ADDRESS: #06/
622 004020      045      101      040 TSSX: .ASCIZ  /#A TSBA,TSSR EXP'D: #06#A,#06#N/
623 004060      045      101      040 TSSX: .ASCIZ  /#A TSBA,TSSR REC'D: #06#A,#06/
624 004117      045      116      045 FUSI: .ASCIZ  /#N#A/
625 004123      040      040      125 USI: .ASCIZ  / UNEXPECTED INTERRUPT/
626 004152      040      040      111 NSI: .ASCIZ  / INTERRUPT EXPECTED, NOT RECEIVED/

```

GLOBAL TEXT MESSAGES

```

627 004215      045      116      045 FNINTR:      .ASCII /%N%A/
628 004221      040      040      116 NOINTR: .ASCIIZ / NO INTERRUPT WAS GENERATED/
629 004256      040      040      111 IFAULT: .ASCIIZ / INTERRUPT FAULT/
630 004300      045      101      040 INTX: .ASCIIZ /%A CPU PC: %06%A TSBA: %06/
631 004335      040      040      042 NOINIT: .ASCIIZ / "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
632 004407      040      040      042 NSINIT: .ASCIIZ / "SOI-T-INIT" DIDN'T INITIALIZE THE DPU/
633 004457      040      040      042 BRINIT: .ASCIIZ / "BUS-RESET" DIDN'T INITIALIZE THE DPU/
634
635 004527      000      116      000 NUL: .ASCIIZ //
636 004530      045      116      000 NULCR: .ASCIIZ /%N/
637 004533      045      101      040 EXPGOT: .ASCIIZ /%A EXP'D: %06%A, REC'D: %06/
638 004567      045      116      045 EXPGT2: .ASCIIZ /%N%A EXP'D: %06%A, %06%N%A REC'D: %0%A, %06/
639 004643      045      101      040 DUAD12: .ASCIIZ /%A REG(W) WRITTEN TO: %06%A REG(R) READ; EXP'D: %06%A, REC'D: %06/
640 004745      122      101      115 PKTRAM: .ASCIIZ 'RAM Contents Do Not Match Packet Sent'
641 005013      040      040      103 SCME: .ASCIIZ / CONFIG DOESN'T MATCH MFG. MASTER/
642 005056      127      127      111 WRTMSG: .ASCIIZ 'WRITE CHARACTERISTICS Failed'
643 005113      124      123      123 WRTERR: .ASCIIZ 'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
644 005206      124      123      123 RDERR: .ASCIIZ 'TSSR Incorrect After READ Command, More Bits Set Than SSR'
645 005300      106      101      124 SCHERR: .ASCIIZ 'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
646 005372      105      122      122 RETERR: .ASCIIZ 'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
647 005460      045      116      045 NOMEM: .ASCIIZ '%N%A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****N'
648 005554      045      116      045 M8186: .ASCIIZ '%N%A ***** 11/23A SYSTEM *****N'
649 005645      045      116      045 M8189: .ASCIIZ '%N%A ***** 11/23B SYSTEM *****N'
650
651
652
653
654
655
656
657
658
659 005736      000      116      000 BGNMSG: .ASCIIZ 'GLOBAL ERROR REPORT SECTION'
660 005736      013746 003114 000000 NXRERR: .ASCIIZ 'THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
005742 012746 003777 000000 CALLS THAT ARE USED IN MORE THAN ONE TEST.
005746 012746 000000 ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
005752 0106
005754 1044
005756 062703 000000
661 005762 004733 005736
662 005766
005766
005766 104423
663
664
665
666
667
668 005770 005727
669 005772 000000
670 005774 001402
671 005776 004777 177770
672 006002
006002 012746 004530
006006 012746 000000

```

N3

SEQ 0039

GLOBAL ERROR REPORT SECTION

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

PRITSSR - PRINT TSSR CONTENTS

```

675                                     .SBTTL  PRITSSR - PRINT TSSR CONTENTS
676
677
678
679                                     ;ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
680                                     ;THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
681                                     ;BY A MESSAGE PRINTING ROUTINE
682
683                                     ;INPUTS:
684                                     ;
685                                     ;      R1      CONTENTS OF TSSR
686
687                                     ;SUBORDINATE ROUTINES:
688                                     ;
689                                     ;      CHKAMB  CHECK FOR AMBIGUOUS CONTENTS
690
691                                     ;-
692
693 006024                                PRITSSR:
694 006024                                SAVREG                                ;SAVE GENERAL REGISTERS
695 006030 010104                        MOV      R1,R4                        ;SAVE THE TSSR CONTENTS
696 006032                                PRINTB  @TSSRFOR,R4                ;PRINT THE CONTENTS OF TSSR
697 006032 010446                        MOV      R4,-(SP)
698 006034 012746 006415                MOV      @TSSRFOR,-(SP)
699 006040 012746 000002                MOV      @2,-(SP)
700 006044 010600                        MOV      SP,R0
701 006046 104414                        TRAP    CIPNTB
702 006050 062706 000006                ADD      @6,SP
703 006054 010400                        MOV      R4,R0                                ;GET TSSR BACK FOR CHKAMB
704 006056 104737 016044                JSR      PC,CHKAMB                ;ARE CONTENTS AMBIGUOUS ?
705 006062 103410                        RCS      5:                            ;BRANCH IF NOT
706 006064 012746 006635                PRINTX  @AMBTSSR
707 006066 012746 000001                MOV      @AMBTSSR,-(SP)
708 006068 010600                        MOV      @1,-(SP)
709 006070 103415                        MOV      SP,R0
710 006100 062706 000004                TRAP    CIPNTX
711 006104 010403                        ADD      @4,SP
712 006106 042703 001476                MOV      R4,R3                                ;CONTENTS OF TSSR
713 006110 001434                        BIC      @HIADDR!FATERH!TERCLS,R3                ;CLEAR ALL MULTIPLE BIT FIELDS
714 006112 012702 002632                BEQ      20:                            ;NO BITS ARE SET
715 006114 012701 003502                MOV      @TMPBFR,R2                ;TEMPORARY ASCII BUFFER
716 006116 005703                        MOV      @TSSRBIT,R1                ;ASCII EQUIVALENT OF BITS
717 006118 001413                        TST      R3                                ;REMAINING BITS TO CONVERT
718 006120 000241                        BEQ      15:                            ;BRANCH WHEN ALL ARE DONE
719 006122 006103                        CLC                                ;CLEAR CARRY FOR SHIFT
720 006124 103076                        ROL      R3                                ;SHIFT NEXT BIT TO CARRY
721 006126 011100                        BCC      13:                            ;BRANCH IF BIT NOT SET
722 006128 112072                        MOV      (R1),R0                ;POINTER TO BIT DEFINITION
723 006130 001376                        MOV      (R0),-(R2)                ;MOVE ASCII TO BUFFER
724 006132 112762 000054 177777        BNE      11:                            ;MOVE ALL BITS
725 006134 005721                        MOV      @',,-1(R2)                ;INSERT A COMMA TO TERMINATE
726 006136 000763                        TST      (R1)                ;POINT TO NEXT DESCRIPTION
727 006138 105042                        BR      10:                            ;GET THE REMAINING BITS
728 006140 012746 002632                CLRB      -(R2)                ;TERMINATE THE LINE
729 006142 012746 006606                PRINTX  @TSSDEF,@TMPBFR                ;PRINT THE BIT DEFINITIONS
730 006144 012746 006606                MOV      @TMPBFR,-(SP)
731 006146 012746 006606                MOV      @TSSDEF,-(SP)

```

C4

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 15-1

SEQ 0041

PRITSSR - PRINT TSSR CONTENTS

006170	012746	000002		MOV	#2,-(SP)	
006174	010600			MOV	SP,R0	
006176	104415			TRAP	C:PNTX	
006200	062706	000006		ADD	#6,SP	
719						
720	006204	010403	201:	MOV	R4,R3	;GET THE TSSR CONTENTS
721	006206	042703		BIC	#1CTERCLS,R3	;CLEAR ALL BUT TERMINATION
722	006212	016303		MOV	TCCOD(R3),R3	;GET THE TERMINATION CODE MEANING
723	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	
724	006240	010403		MOV	R4,R3	;TSSR CONTENTS AGAIN
725	006242	042703		BIC	#1CFATERR,R3	;CLEAR ALL BUT FATAL TERMINATION
726	006246	001416		BEQ	251	;DON'T PRINT IF ZERO
727	006250	006203		ASR	R3	
728	006252	006203		ASR	R3	
729	006254	006203		ASR	R3	
730	006256	016303		MOV	TSFCOD(R3),R3	;ALINE TERMINATION CODE FOR INDEX
731	006262			PRINTX	#TFCASC,R3	;GET THE FATAL TERMINATION CODE
	006262	010346		MOV	R3,-(SP)	;PRINT THE FATAL TERMINATION CODE
	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	
732	006304	042704	251:	BIC	#1CHIADDR,R4	;CLEAR ALL BUT EXTENDED ADDRESS
733	006310	001411		BEQ	301	;DON'T PRINT IF ZERO
734	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	
735	006334	013703	301:	MOV	EPRTSW,R3	;PRINT MESSAGE BUFFER ADDRESS
736	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	
737	006356	000207		RTS	PC	;RETURN TO CALLER
738						
744	006360			EPRT2:		
745	006360	045	116	EPRT1:	.ASCIZ	'###A *****REPLACE M7196*****'
746						
756	006415	045	116	TSSRFOR:	.ASCIZ	'###A TSSR = #06'
757	006435	045	116	TEXASC:	.ASCIZ	'###A Extended Address Bits = #06'
758	006476	045	116	TCOASC:	.ASCIZ	'###A Termination Class Code = #T'
759	006537	045	116	TFCASC:	.ASCIZ	'###A Fatal Termination Class Code = #T'
760	006606	045	116	TSSDEF:	.ASCIZ	'###A TSSR Bits Set: #T'
761	006635	045	116	AMBTSSR:	.ASCIZ	'###A TSSR Contents Are Ambiguous'
762						

,EVEN

D4

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 15-2

SEQ 0042

PRITSSR - PRINT TSSR CONTENTS

```

763 006676 006716 006741 006767 TCOCOD: .WORD 1#,2#,3#,4#,5#,6#,7#,8#
764 006716      116      157      162 1#: .ASCIZ 'Normal Termination'
765 006741      124      145      162 2#: .ASCIZ 'Termination Condition'
766 006767      124      141      160 3#: .ASCIZ 'Tape Status Alert'
767 007011      106      165      156 4#: .ASCIZ 'Function Reject'
768 007031      122      145      143 5#: .ASCIZ 'Recoverable Error - Tape Position One Record Down'
769 007113      122      145      143 6#: .ASCIZ 'Recoverable Error - Tape Was Not Moved'
770 007162      125      156      162 7#: .ASCIZ 'Unrecoverable Error'
771 007206      106      141      164 8#: .ASCIZ 'Fatal Controller Error'
772      .EVEN
773
774 007236 007246 007302 007313 TSFCOD: .WORD 1#,2#,3#,4#
775 007246      111      156      164 1#: .ASCIZ 'Internal Diagnostic Failure'
776 007302      122      145      163 2#: .ASCIZ 'Reserved'
777 007313      102      165      163 3#: .ASCIZ 'Bus Interface or Sanity Check Error'
778 007357      122      145      163 4#: .ASCIZ 'Reserved'
779      .EVEN
780      .SBTTL    PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET
781
782
783      ;
784      ;THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
785      ;THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
786      ;
787      ;INPUT:
788      ;
789      ;        R0        NUMBER OF WORDS IN PACKET
790      ;        R3        HIGH ORDER COMMAND PACKET ADDRESS
791      ;        R4        ADDRESS OF COMMAND PACKET
792      ;
793      ;        NOTE:    R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
794      ;
795
796 007370
797 007370
798 007374 010005
799 007376 005737 003134
800 007402 001001
801 007404 005003
802 007406 010301
803 007410 010400
804 007412 006100
805 007414 006101
806 007416
807 007416 010446
808 007420 010146
809 007422 012746 007554
810 007426 012746 000003
811 007432 010600
812 007434 104414
813 007436 062706 000010
814 007442 010300
815 007444 001404
816 007446 010401
817 007450 004737 017316
818 007454 010004
819 007456 005001
820 007460 012402

```

```

PRIPKT::
    SAVREG
    MOV    R0,R5        ;SAVE THE REGISTERS
    TST    KTENABLE    ;SAVE NO. OF WORDS IN PACKET
    BNE    10#        ;ABOVE 28K UNDER TEST?
    CLR    R3        ;BR IF YES
    MOV    R3,R1       ;SET HIGH ORDER ADDRESS TO 0
    MOV    R4,R0       ;COPY HIGH ORDER ADDRESS
    ROL    R0        ;GET LOWER ADDRESS
    ROL    R1        ;SHIFT BIT 15 INTO C BIT
    PRINTB @PKTADD,R1,R4 ;AND INTO HIGH ORDER.
    MOV    R4,-(SP)    ;PRINT PACKET ADDRESS
    MOV    R1,-(SP)
    MOV    @PKTADD,-(SP)
    MOV    @3,-(SP)
    MOV    SP,R0
    TRAP   C#PNTB
    ADD    @10,SP
    MOV    R3,R0       ;GET HIGH ORDER ADDRESS
    BEQ    20#        ;BR IF NOT ABOVE 28K.
    MOV    R4,R1       ;GET LOW ORDER ADDRESS
    JSR    PC,SETMAP   ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
    MOV    R0,R4       ;GET RETURNED PAR6 ADDRESS BIAS
    CLR    R1        ;SAVE WORD NUMBER
    MOV    (R4),R2    ;GET PACKET CONTENTS
    10#:
    15#:
    20#:
    25#:

```

E4

SEQ 0043

PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

```

813 007462          PRINTB  PKTFRM,R1,R2  ;PRINT THE DATA
      007462 010246      MOV      R2,-(SP)
      007464 010146      MOV      R1,-(SP)
      007466 012746 007516  MOV      PKTFRM,-(SP)
      007472 012746 000003  MOV      #3,-(SP)
      007476 010600      MOV      SP,R0
      007500 104414      TRAP     C:PNTB
      007502 062706 000010  ADD      #10,SP
814 007506 005201      INC      R1          ;NEXT WORD NUMBER
815 007510 020105      CMP      R1,R5      ;DONE ALL PACKET WORDS?
816 007512 002762      BLT      25$      ;LOOP TILL ALL DONE
817 007514 000207      RTS      PC          ;RETURN
818
819 007516      045      116      045  PKTFRM: .ASCIZ  'NMA Packet Word #01#A = #06'
820 007554      045      116      045  PKTADD: .ASCIZ  'NMA Packet Address = #01#05'
821                                     .EVEN
822                                     .SBTTL  PRIBXOR - PRINT EXPD, RECV AND XOR BYTE
823
824
825
826
827 ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
828 ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
829
830 ;INPUTS:
831
832 ;      R1      RECEIVED DATA
833 ;      R2      EXPECTED DATA
834
835 ;OUTPUT:
836
837 ;      R0      XOR OF EXPECTED/RECEIVED DATA
838
839
840 PRIBXOR::
841 SAVREG          ;SAVE THE REGISTERS
842 MOV      R2,R3  ;EXPECTED DATA
843 XOR      R1,R3  ;FORM THE EXCLUSIVE OR
844 MOV      #C<377>,R0 ;BYTE MASK
845 BIC      R0,R1  ;SAVE LOW BYTE RECV
846 BIC      R0,R2  ;SAVE LOW BYTE EXPD
847 BIC      R0,R3  ;SAVE LOW BYTE XOR
848 PRINTB  XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
      007642 010346      MOV      R3,-(SP)
      007644 010146      MOV      R1,-(SP)
      007646 010246      MOV      R2,-(SP)
      007650 012746 007674  MOV      XORBFOR,-(SP)
      007654 012746 000004  MOV      #4,-(SP)
      007660 010600      MOV      SP,R0
      007662 104414      TRAP     C:PNTB
      007664 062706 000012  ADD      #12,SP
849 007670 010300      MOV      R3,R0      ;R0 HAS XOR ON RETURN
850 007672 000207      RTS      PC          ;RETURN TO CALLER
851
852 007674      045      116      045  XORBFOR: .ASCIZ  'NMA EXPD: #03#A RECV: #03#A XOR: #03'
853                                     .EVEN
854                                     .SBTTL  PRIXOR - PRINT EXPD, RECV AND XOR

```

PRI XOR - PRINT EXPD, RECV AND XOR

```

855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872 007742
873 007742
874 007746 010203
875 007750
876 007760
      007760 010346
      007762 010146
      007764 010200
      007766 012746 010012
      007772 012746 000004
      007776 010600
      010000 104414
      010002 062706 000012
877 010006 010300
878 010010 000207
879
880 010012 045 116 045 XORFOR: .ASCIZ '06 EXPD: 06 RECV: 06 XOR: 06'
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896 010060
897 010060
898 010064 000207
899
900
901
902
903

```

```

; *
; PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
; THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
;
; INPUTS:
;
;      R1      RECEIVED DATA
;      R2      EXPECTED DATA
;
; OUTPUT:
;
;      R0      XOR OF EXPECTED/RECEIVED DATA
;
; -
PRI XOR:
    SAVREG                ;SAVE THE REGISTERS
    MOV      R2,R3        ;EXPECTED DATA
    XOR      R1,R3        ;FORM THE EXCLUSIVE OR
    PRINTB   XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
    MOV      R3,-(SP)
    MOV      R1,-(SP)
    MOV      R2,-(SP)
    MOV      XORFOR,-(SP)
    MOV      04,-(SP)
    MOV      SP,R0
    TRAP     C1PNTB
    ADD      012,SP
    MOV      R3,R0        ;R0 HAS XOR ON RETURN
    RTS      PC           ;RETURN TO CALLER

045 XORFOR: .ASCIZ '06 EXPD: 06 RECV: 06 XOR: 06'
          .EVEN
          .SBTTL PRI EQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT

; *
; ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
; THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
;
; INPUTS:
;
;      R0      OCTAL VALUE TO CONVERT
;      R1      TABLE OF POINTERS TO ASCII EQUIVALENT
;
; -
PRI EQU:
    SAVREG                ;SAVE THE REGISTERS
    RTS      PC           ;RETURN TO CALLER

          .SBTTL PRI RAM - PRINT RAM ADDRESS

; *
; PRINT CONTROLLER RAM ADDRESS.

```

G4

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 15-5

SEQ 0045

PRIRAM - PRINT RAM ADDRESS

```

904      ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
905      ;
906      ;INPUTS:
907      ;
908      ;      R4      RAM ADDRESS
909      ;
910      ;-
911      PRIRAM:
912      SAVREG      ;SAVE R1-R5 UNTIL NEXT RETURN
913      PRINTB      RAMFOR,R4      ;PRINT RAM ADDRESS IN ERROR
914      MOV      R4,-(SP)
915      MOV      RAMFOR,-(SP)
916      MOV      R2,-(SP)
917      MOV      SP,R0
918      TRAP      C:PNTB
919      ADD      #6,SP
920      RTS      PC      ;RETURN
921
922      045  RAMFOR: .ASCIZ  '***A CONTROLLER RAM ADDRESS * #06'
923      .EVEN
924
925      .SBTTL  PRIADD - PRINT MEMORY ERROR ADDRESS
926      ;+
927      ;PRINT MEMORY ADDRESS
928      ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
929      ;
930      ;IMPLICIT INPUTS
931      ;
932      ;      ERRHI   - HIGH ORDER ADDRESS
933      ;      ERRLO   - LOW ORDER ADDRESS
934      ;
935      ;-
936      PRIADD:
937      SAVREG      ;SAVE R1-R5 UNTIL NEXT RETURN
938      MOV      ERRHI,R0      ;GET HIGH ADDRESS
939      MOV      ERRLO,R1      ;GET LOW ADDRESS
940      MOV      R1,R2      ;COPY LOW ADDRESS
941      ROL      R1      ;SHIFT BIT 15 TO C BIT
942      ROL      R0      ;SHIFT INTO HIGH ORDER
943      PRINTB      PRIA0,R0,R2      ;PRINT MEMORY ADDRESS IN ERROR
944      MOV      R2,-(SP)
945      MOV      R0,-(SP)
946      MOV      PRIA0,-(SP)
947      MOV      #3,-(SP)
948      MOV      SP,R0
949      TRAP      C:PNTB
950      ADD      #10,SP
951      RTS      PC      ;RETURN
952
953      045  PRIA0: .ASCIZ  '***A MEMORY ERROR ADDRESS * #01#05'
954      .EVEN
955
956      .SBTTL  PRITADD - PRINT MEMORY TEST ADDRESS
957      ;+
958      ;PRINT MEMORY ADDRESS

```

PRITADD - PRINT MEMORY TEST ADDRESS

```

948      ; THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
949      ;
950      ; IMPLICIT INPUTS
951      ;
952      ;     ERRHI   - HIGH ORDER ADDRESS
953      ;     ERRLO   - LOW ORDER ADDRESS
954      ;
955      ; -
956      PRITADD:
957      SAVREG      ;SAVE R1-R5 UNTIL NEXT RETURN
958      MOV     ERRHI,R2      ;GET HIGH ADDRESS
959      MOV     ERRLO,R1      ;GET LOW ADDRESS
960      ;MOV     R1,R2      ;COPY LOW ADDRESS
961      ;ROL     R1      ;SHIFT BIT 15 TO C BIT
962      ;ROL     R0      ;SHIFT INTO HIGH ORDER
963      PRINTB  #PRITO,R1      ;PRINT MEMORY ADDRESS LOW IN ERROR
964      MOV     R1,-(SP)
965      MOV     #PRITO,-(SP)
966      MOV     #2,-(SP)
967      MOV     SP,R0
968      TRAP    C#PNTB
969      ADD     #6,SP
970      PRINTB  #PRIT1,R2      ;PRINT MEMORY ADDRESS HIGH IN ERROR
971      MOV     R2,-(SP)
972      MOV     #PRIT1,-(SP)
973      MOV     #2,-(SP)
974      MOV     SP,R0
975      TRAP    C#PNTB
976      ADD     #6,SP
977      RTS     PC      ;RETURN
978
979      045 PRITO: .ASCIZ 'MMA MEMORY TEST ADDRESS LOW = #06'
980      045 PRIT1: .ASCIZ 'MMA MEMCRY TEST ADDRESS HIGH = #06'
981      .EVEN
982      .SBTTL  SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND
983
984      ; *
985      ; ROUTINE TO ISSUE A SPACE RECORDS
986      ; COMMAND (FORWARD OR REVERSE)
987      ;
988      ; INPUT:
989      ;
990      ;     R3      NUMBER OF RECORDS TO BE SPACED OVER
991      ;             BIT15 CONTROLS DIRECTION
992      ;             BIT15 = 0 IS FORWARD
993      ;             BIT15 = 1 IS REVERSE
994      ;     R5      FIRST DEVICE UNIBUS ADDRESS
995      ;
996      ; REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
997      ;
998      ; OUTPUT:
999      ;
1000     ;     CARRY   SET - SPACE RECORDS COMMAND OK
1001     ;             CLR - SPACE RECORDS FAILED
1002     ;

```

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

```

993      ;      RO      THE CONTENTS OF R4 IS MOVED TO RO
994      ;
995      ;
996      ;IMPLICIT OUTPUT:
997      ;
998      ;      TAPE HAS BEEN MOVED
999      ;
1000     ;SIDE EFFECTS:
1001     ;
1002     ;
1003     ;
1004     ;
1005     SPACE::
1006     010466      SAVREG
1007     010472      MOV      #500.,SDELAY      ;SAVE THE GENERAL REGISTERS
1008     010500      MOV      #140010,80$      ;SET UP DELAY
1009     010506      TST      R3                ;SET UP COMMAND, SPACE FORWARD
1010     010510      BMI      5$                ;CHECK FOR DIRECTION
1011     010512      MOV      R3,90$           ;BR, IF REVERSE INDICATED
1012     010516      BR       10$              ;LOAD UP NUMBER OF RECORDS TO SPACE
1013     010520      042703      100000      5$:  BIC      #BIT15,R3      ;GO DO COMMAND
1014     010524      010337      010652      MOV      R3,90$           ;CLEAR DIRECTION BIT
1015     010530      052737      000400      010650 10$: MOV      #BIT8,80$      ;LOAD UP NUMBER OF RECORDS TO SPACE
1016     010536      012704      010650      BIS      #80$,R4          ;SET REVERSE BIT IN COMMAND PACKET
1017     010540      010465      000000      15$: MOV      R4,TSD8(R5)    ;SET UP R4 WITH PACKET ADDRESS
1018     010544      010737      016250      MOV      R4,TSD8(R5)    ;SEND OUT COMMAND
1019     010548      0103420      BCS      20$      ;WAIT FOR SSR
1020     010552      DELAY      250              ;BR, IF SSR IS SET AND OK
1021     010556      012727      000250      MOV      #250,(PC)+      ;DELAY ABOUT .25 SECONDS
1022     010560      000000      .WORD      0
1023     010562      013727      002116      MOV      L$DLY,(PC)+
1024     010566      000000      .WORD      0
1025     010570      005367      177772      DEC      -6(PC)
1026     010574      001375      BNE      -4
1027     010576      005367      177756      DEC      -22(PC)
1028     010602      001367      BNE      -20
1029     010604      005337      010660      DEC      SDELAY
1030     010610      001356      BNE      15$
1031     010612      000411      BR       60$
1032     010614      016501      000002      20$: MOV      TSSR(R5),R1
1033     010620      012702      000200      MOV      #SSR,R2
1034     010624      020201      25$:  CMP      R2,R1
1035     010626      001401      BEQ      40$
1036     010630      000402      BR       60$
1037     010632      000261      40$:  SEC
1038     010634      000401      BR       70$
1039     010636      000241      60$:  CLC
1040     010640      010400      70$:  MOV      R4,RO
1041     010642      000207      RTS      PC

```

```

;PASS PACKET ADDRESS
;RETURN

```

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

```

1036      ;
1037      ;
1038      ;
1039      ; PACKET FOR SPACE COMMAND
1040      ;
1042      010650      ,=<.,+10>&177770
1044      ;
1045      ; COMMAND WORD
1046      010650      000000      80$: .WORD
1047      ; NUMBER OF RECORDS TO BE SPACED OVER WORD
1048      010652      000000      90$: .WORD
1049      010654      000000      .WORD
1050      010656      000000      .WORD
1051      010660      000000      SDELAY: .WORD 0 ; DELAY COUNTER
1052      ;
1053      ; SBTTL WRTCHR - WRITE CHARACTERISTICS COMMAND
1054      ;
1055      ;
1056      ;
1057      ; ROUTINE TO ISSUE A WRITE CHARACTERISTICS
1058      ; COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
1059      ;
1060      ; INPUT:
1061      ;
1062      ; R4 ADDRESS OF PACKET FROM TEST
1063      ; R5 FIRST DEVICE UNIBUS ADDRESS
1064      ; REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
1065      ;
1066      ; OUTPUT:
1067      ;
1068      ; R0 TSSR CONTENTS
1069      ; CARRY SET - WRITE CHARACTERISTICS COMMAND OK
1070      ; CLR - WRITE CHARACTERISTICS FAILED
1071      ;
1072      ; IMPLICIT OUTPUT:
1073      ;
1074      ; MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
1075      ; SOFTWARE SWITCHES SET AS FOLLOWS:
1076      ; EXTFEA = EXTENDED FEATURES PRESENT
1077      ; BENBSW = JUFFER ENABLE SWITCH ON OR OFF
1078      ;
1079      ;
1080      ; SIDE EFFECTS:
1081      ;
1082      ;
1083      ;
1084      ;
1085      010662      WRTCHR: ;
1086      010662      ; SAVREG ; SAVE THE GENERAL REGISTERS
1087      010666      005037 002230      CLR BENBSW ; CLEAR BUFFER ENABLE SWITCH
1088      010672      005037 002226      CLR EXTFEA ; CLEAR EXTENDED FEATURES SW SWITCH
1089      010676      010465 000000      10$: MOV R4,TSDB(R5) ; SEND OUT COMMAND
1090      010702      004737 016336      JSR PC,CHKTSSR ; WAIT FOR SSR
1091      010706      103401      BCS 20$ ; BR, IF SSR IS SET AND OK
1092      010710      000435      BR 60$ ; BR IF TROUBLE CARRY = CLEAR
1093      010712      016501 000002      20$: MOV TSSR(R5),R1 ; READ TSSR
1094      010716      012702 000200      MOV 4SSR,R2 ; SET UP EXPECTED

```

[illegible]

L4

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 16-2

SEQ 0050

REWIND - POSITION TAPE (REWIND) COMMAND

```

011042 012727 000372      MOV      #250.,(PC)+
011046 000000      .WORD      0
011050 013727 002116      MOV      L$DLY,(PC)+
011054 000000      .WORD      0
011056 005367 177772      DEC      -6(PC)
011062 001375      BNE      , -4
011064 005367 177756      DEC      -22(PC)
011070 001367      BNE      , -20
1152 011072 005303      DEC      R3          ;BUMP COUNTER DOWN
1153 011074 001357      BNE      10$         ;KEEP GOING
1154 011076 000241      CLC              ;CLEAR CARRY TO SET ERROR
1155 011100 010400      20$: MOV      R4,R0      ;PASS THE PACKET ADDRESS
1156 011102 000207      RTS       PC          ;RETURN
1157
1159          011110      RWPack: .*=.,+10>E177770
1161 011110          .WORD      102010          ;POSITION COMMAND (REWIND)
1162 011110 102010      .WORD      0          ;NOT USED
1163 011112 000000      .SBTTL   CKRAM      - COMPARE RAM TO I/O PACKET
1164
1165
1166      ;+
1167      ;
1168      ;ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
1169      ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
1170      ;
1171      ;INPUT:
1172      ;
1173      ;      R4      ADDRESS OF THE COMMAND PACKET
1174      ;      R5      FIRST DEVICE UNIBUS ADDRESS
1175      ;
1176      ;OUTPUT:
1177      ;
1178      ;      CARRY   SET - RAM MATCHES PACKET
1179      ;              CLR - RAM DOES NOT MATCH PACKET
1180      ;
1181      ;IMPLICIT OUTPUT:
1182      ;
1183      ;      THE TABLE RAMDATA IS FILLED WITH THE
1184      ;      DATA HELD IN RAM.
1185      ;      RAMSIZ IS SET TO 8. FOR PRAMPKT ROUTINE
1186      ;
1187      ;SIDE EFFECTS:
1188      ;
1189      ;      THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1190      ;
1191      ;-
1192
1193 011114      CKRAM:;
1194 011114      SAVREG
1195 011120 012701 002242      MOV      #RAMDATA,R1      ;SAVE THE GENERAL REGISTERS
1196 011124 012702 000201      MOV      #RMPKTBEGR,R2      ;ADDRESS TO SAVE THE RAM DATA
1197 011130 005003          CLR      R3          ;BYTE ADDRESS OF FIRST RAM DATA
1198 011132 004737 016336      JSR      PC,CHKTSSR      ;CLEAR THE ERROR FLAG
1199 011136 112765 000000 000000      MOVB   #0,TSDB(R5)      ;WAIT FOR SSR
1200 011144 004737 016336      10$: JSR      PC,CHKTSSR      ;SET MAINTENANCE MODE
1201 011150 010265 000000      MOV      R2,TSDB(R5)      ;WAIT FOR SSR TO SET
1202 011154 004737 016336      JSR      PC,CHKTSSR      ;SELECT NEXT RAM ADDRESS
                                ;WAIT FOR SSR TO SET

```

CKRAM - COMPARE RAM TO I/O PACKET

```

1203 011160 116511 000000      MOV8    TSBA(R5),(R1)      ;READ THE RAM DATA
1204 011164 122124      CMPB     (R1)+,(R4)+      ;COMPARE TO EXPECTED
1205 011166 001401      BEQ      20$             ;BRANCH IF OK
1206 011170 005203      INC      R3              ;SET ERROR FLAG
1207 011172 005202      20$: INC      R2          ;ADDRESS OF NEXT RAM LOCATION
1208 011174 020227 000210    CMP      R2,PRMPKTEND ;REACHED END YET ?
1209 011200 003761      BLE      10$             ;BRANCH TILL ALL READ
1210 011202 005703      TST      R3              ;WAS AN ERROR FOUND ?
1211 011204 001402      BEQ      30$             ;BRANCH IF NO
1212 011206 000241      CLC                      ;CLEAR CARRY TO SHOW ERROR
1213 011210 000401      BR       50$             ;AND EXIT
1214 011212 000261      30$: SEC                 ;SHOW GOOD COMPARE
1215 011214 012737 000010 002302 50$: MOV      #8, RAMSIZ ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
1216 011222 000207      RTS      PC              ;RETURN
1217                                     .SBTTL   CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA
1218                                     ;+
1219                                     ;
1220                                     ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
1221                                     ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
1222                                     ;
1223                                     ;INPUT:
1224                                     ;
1225                                     ;      R4      ADDRESS OF THE CHARACTERISTICS DATA
1226                                     ;      R5      FIRST DEVICE UNIBUS ADDRESS
1227                                     ;
1228                                     ;OUTPUT:
1229                                     ;
1230                                     ;      CARRY    SET - RAM MATCHES PACKET
1231                                     ;              CLR - RAM DOES NOT MATCH PACKET
1232                                     ;
1233                                     ;IMPLICIT OUTPUT:
1234                                     ;
1235                                     ;      THE TABLE RAMDATA IS FILLED WITH THE
1236                                     ;      DATA HELD IN RAM.
1237                                     ;      RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
1238                                     ;
1239                                     ;SIDE EFFECTS:
1240                                     ;
1241                                     ;      THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1242                                     ;
1243                                     ;-
1244
1245 011224      CKRAM2: SAVREG                      ;SAVE THE GENERAL REGISTERS
1246 011224      MOV      #RAMDATA,R1              ;ADDRESS TO SAVE THE RAM DATA
1247 011230 012701 002242    MOV      #RMCHBEG,R2   ;BYTE ADDRESS OF FIRST RAM DATA
1248 011234 012702 000167    CLR      R3           ;CLEAR THE ERROR FLAG
1249 011240 005003      JSR      PC,CHKTSSR         ;WAIT FOR SSR
1250 011242 004737 016336    MOV8     #0,TSDB(R5)  ;SET MAINTENANCE MODE
1251 011246 112765 000000 000000    JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
1252 011254 004737 016336 10$: MOV      R2,TSDB(R5) ;SELECT NEXT RAM ADDRESS
1253 011260 010265 000000    JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
1254 011264 004737 016336    MOV8     TSBA(R5),(R1) ;READ THE RAM DATA
1255 011270 116511 000000    CMPB     (R1)+,(R4)+ ;COMPARE TO EXPECTED
1256 011274 122124      BEQ      20$             ;BRANCH IF OK
1257 011276 001401      INC      R3              ;SET ERROR FLAG
1258 011300 005203      20$: INC      R2          ;ADDRESS OF NEXT RAM LOCATION
1259 011302 005202

```

CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA

Address	Offset	Hex	Symbol	Label	Comment
1260	011304	012737	000010	002302	MOV #8.,RAMSIZ ;ASSUME EXTFEA NOT SET
1261	011312	005737	002226		TST EXTFEA ;IS THE SOFTWARE EXTENDED FEATURES SET
1262	011316	001407			BEQ 25\$;BR, IF NOT SET
1263	011320	012737	000012	002302	MOV #10.,RAMSIZ ;SET RAMSIZ FOR EXTEND FEATURES
1264	011326	020227	000200		CMP R2,#RMCHEND ;AT END OF EXTENDED BUFFER
1265	011332	003750			BLE 10\$;BR, IF NOT AT END YET
1266	011334	000403			BR 27\$;AT END BRANCH
1267	011336	020227	000176	25\$:	CMP R2,#RMCHEND-2 ;REACHED END YET ?
1268	011342	003744			BLE 10\$;BRANCH TILL ALL READ
1269	011344	005703		27\$:	TST R3 ;WAS AN ERROR FOUND ?
1270	011346	001402			BEQ 30\$;BRANCH IF NOT
1271	011350	000241			CLC ;CLEAR CARRY TO SHOW ERROR
1272	011352	000401			BR 50\$;AND EXIT
1273	011354	000261		30\$:	SEC ;SHOW GOOD COMPARE
1274	011356	000207		50\$:	RTS PC ;RETURN
1275					.SBTTL CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS
1276					;
1277					;
1278					;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
1279					;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1280					;ERROR PRINT ROUTINES.
1281					;
1282					;INPUT:
1283					;
1284					R0 RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1285					R1 RECV MESSAGE BUFFER LOW ORDER ADDRESS
1286					R2 EXPD MESSAGE BUFFER ADDRESS
1287					;OUTPUT:
1288					;
1289					CARRY SET - MESSAGE BUFFERS MATCH
1290					CLR -MESSAGE BUFFERS DON'T MATCH
1291					;
1292					;IMPLICIT OUTPUT:
1293					;
1294					EXPMSG BUFFER IS SET TO EXPD DATA
1295					RECMMSG BUFFER IS SET TO RECV DATA
1296					RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
1297					RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
1298					;
1299					;-
1300	011360				CKMSG:;
1301	011360				SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1302	011364	010037	002304		MOV R0,RCVHIADD ;SAVE RECV HIGH ADDRESS
1303	011370	010137	002306		MOV R1,RCVLOADD ;SAVE RECV LOW ADDRESS
1304	011374	005737	003134		TST KTENABLE ;TESTING ABOVE 28K?
1305	011400	001403			BEQ 10\$;BR IF NO
1306	011402	004737	017316		JSR PC,SETMAP ;RETURN ADDRESS BIASED TO PAR6 IN R0
1307	011406	010001			MOV R0,R1 ;GET RETURNED ADDRESS BIASED TO PAR6
1308	011410	005004		10\$:	CLR R4 ;WORD IN BUFFER
1309	011412	005003			CLR R3 ;CLEAR ERROR SEEN FLAG
1310	011414	010205			MOV R2,R5 ;GET EXPD BUFFER ADDRESS
1311	011416	011264	002322	15\$:	MOV (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
1312	011422	011164	002466		MOV (R1),RECMMSG(R4) ;SAVE RECV FOR ERROR REPORT
1313	011426	0222			CMP (R2)+,(R1)+ ;EXPD EQUAL RECV?
1314	011430	001401			BEQ 25\$;BR IF YES
1315	011432	005203			INC R3 ;SET ERROR SEEN FLAG
1316	011434	002704	000002	25\$:	ADD #2,R4 ;POINT TO NEXT WORD ADDRESS

B5

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 16-5

SEQ 0053

CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS

```

1317 011440 020427 000014      CMP      R4,#14      ;DONE FIRST 7 WORDS?
1318 011444 003764      BLE      15#      ;BR IF NO
1319 011446 032765 000200 000012  BIT      #X2.EXTF,XST2(R5) ;IS EXTENDED FEATURES SET IN EXPD?
1320 011454 001403      BEQ      50#      ;BR IF NO
1321 011456 020427 000016      CMP      R4,#16      ;DONE EXTENDED FEATURES WORD?
1322 011462 003755      BLE      15#      ;BR IF NO
1323 011464 005703      50# :   TST      R3      ;ANY ERRORS SEEN?
1324 011466 001402      BEQ      55#      ;BR IF NO
1325 011470 000241      CLC      ;SET FAILURE
1326 011472 000401      BR      60#      ;
1327 011474 000261      55# :   SEC      ;SET SUCCESS
1328 011476 000207      60# :   RTS      PC      ;RETURN
1329      ;SBTTL CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS
1330      ;
1331      ;
1332      ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
1333      ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1334      ;ERROR PRINT ROUTINES.
1335      ;
1336      ;INPUT:
1337      ;
1338      ;      R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1339      ;      R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
1340      ;      R2      EXPD MESSAGE BUFFER ADDRESS
1341      ;      R3      NUMBER OF BYTES TO COMPARE
1342      ;
1343      ;OUTPUT:
1344      ;
1345      ;      CARRY   SET - MESSAGE BUFFERS MATCH
1346      ;             CLR - MESSAGE BUFFERS DON'T MATCH
1347      ;
1348      ;IMPLICIT OUTPUT:
1349      ;
1350      ;      EXPMSG   BUFFER IS SET TO EXPD DATA
1351      ;      RECVMSG  BUFFER IS SET TO RECV DATA
1352      ;      RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
1353      ;      RCVLOAD  SET TO LOW ORDER ADDRESS OF RECV
1354      ;
1355      ;-
1356 011500      CKMSG2:;
1357 011500      SAVREG      ;SAVE R1-R5 UNTIL NEXT RETURN
1358 011504 020327 000144      CMP      R3,#RECVMSG-EXPMSG ;IS COUNT ABOVE MAX ALLOWED?
1359 011510 003412      BLE      5#      ;BR IF NO
1360 011512 012703 000144      MOV      #RECVMSG-EXPMSG,R3 ;
1361 011516 012746 011632      PRINTF   #DEBUGMSG ;
1362      011516 012746 000001      MOV      #DEBUGMSG,-(SP)
1363      011522 012746 000001      MOV      #1,-(SP)
1364      011526 010600      MOV      SP,R0
1365      011530 104417      TRAP      C:PRINTF
1366      011532 062706 000004      ADD      #4,SP
1367 011536 010037 002304      5# :   MOV      R0,RCVHIADD ;SAVE RECV HIGH ADDRESS
1368 011542 010137 002306      MOV      R1,RCVLOAD ;SAVE RECV LOW ADDRESS
1369 011546 005737 003134      TST      KENABLE ;TESTING ABOVE 28K?
1370 011552 001403      BEQ      10#      ;BR IF NO
1371 011554 004737 017316      JSR      PC,SETMAP ;RETURN ADDRESS BIASED TO PAR6 IN R0
1372 011560 010001      MOV      R0,R1 ;GET RETURNED ADDRESS BIASED TO PAR6
1373 011562 005004      10# :   CLR      R4 ;WORD IN BUFFER

```

C5

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 16-6

SEQ 0054

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

1369 011564 005005          CLR      R5          ;CLEAR ERROR SEEN FLAG
1370 011566 111264 002322 15+:  MOVB    (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
1371 011572 111164 002466      MOVB    (R1),RECVMSG(R4) ;SAVE RECV FOR ERROR REPORT
1372 011576 122221          CMPB    (R2)+,(R1)+      ;EXPD EQUAL RECV?
1373 011600 001401          BEQ      25:            ;BR IF YES
1374 011602 005205          INC      R5              ;SET ERROR SEEN FLAG
1375 011604 062704 000001 25+:  ADD     #1,R4        ;POINT TO NEXT BYTE
1376 011610 020403          CMP      R4,R3           ;DONE ALL BYTES?
1377 011612 002001          BGE      50:            ;BR IF YES
1378 011614 000764          BR       15:            ;DO NEXT BYTE
1379 011616 005705          50+:  TST     R5          ;ANY ERRORS SEEN?
1380 011620 001402          BEQ      55:            ;BR IF NO
1381 011622 000241          CLC                     ;SET FAILURE
1382 011624 000401          BR       60:            ;
1383 011626 000261          55+:  SEC                     ;SET SUCCESS
1384 011630 000207          60+:  RTS      PC          ;RETURN
1385
1386 011632      120      122      117 DEBUGMSG:      .ASCIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-' ;000
1387 011722      045      116      045 FERCM: .ASCII /NMA ***/
1388 011733      040      040      124 ERCH:  .ASCIZ / TSSR ERROR CODE REC'D - /
1389 011766      056      056      056 SIMSG: .ASCIZ /... AFTER DOING SOFT INIT/
1390 012021      124      105      123 TINERR: .ASCIZ /TEST: .../
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407 012034          BGNMSG  SFIMSG
      012034          SFIMSG:: JSR      PC,PRI:TSSR      ;PRINT CONTENTS OF TSSR REGISTER
1408 012034 004737 006024      JSR      PC,CKDROP      ;DROP UNIT, IF ALLOWED
1409 012040 004737 017202      ENDMMSG
1410 012044          L10003: TRAP     C:MSG
      012044          104423
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422

```

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

1423 012046      BGNMSG  PKTSSR
      012046      PKTSSR:
1424 012046 004737 006024      JSR    PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1425 012052 012700 000004      MOV    #4,R0          ;NO. OF WORDS IN PACKET
1426 012056 004737 007370      JSR    PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
1427 012062      ENDMMSG
      012062      L10004:
      012062 104423      TRAP    C#MSG

1428
1429
1430      ;*
1431      ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1432      ;TSSR AND A GET STATUS COMMAND PACKET.
1433      ;
1434      ;INPUTS:
1435      ;
1436      ;      R1      TSSR CONTENTS
1437      ;      R4      ADDRESS OF COMMAND PACKET
1438      ;
1439      ;-
1440 012064      BGNMSG  PKTGETS
      012064      PKTGETS:
1441 012064 004737 006024      JSR    PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1442 012070 012700 000002      MOV    #2,R0          ;NO. OF WORDS IN GET STATUS PACKET
1443 012074 004737 007370      JSR    PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
1444 012100      ENDMMSG
      012100      L10005:
      012100 104423      TRAP    C#MSG

1445
1446
1447      ;*
1448      ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
1449      ;
1450      ;INPUTS:
1451      ;
1452      ;      R1      TSSR C 'NTENTS
1453      ;      R4      ADDRESS OF COMMAND PACKET
1454      ;
1455      ;-
1456 012102      BGNMSG  SFFMSG
      012102      SFFMSG:
1457 012102 004737 006024      JSR    PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
      012106      ENDMMSG
      012106      L10006:
      012106 104423      TRAP    C#MSG

1458      .SBTTL  PKTMES - PRINT TSSR AND MESSAGE BUFFER
1459
1460      ;*
1461      ;
1462      ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
1463      ;BUFFER FOR ERROR REPORTS
1464      ;
1465      ;INPUTS:
1466      ;
1467      ;      R1      CONTENTS OF TSSR
1468      ;      R2      LOW ORDER MESSAGE BUFFER
1469      ;      R3      HIGH ORDER MESSAGE BUFFER ADDRESS
1470      ;      NOTE: R3 IS IGNORED IF KENABLE FLAG IS CLEAR

```

E5

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 16-8

SEQ 0056

PKTMES - PRINT TSSR AND MESSAGE BUFFER

```

1471
1472 012110
1473 012110 004737 006024
1474 012114 010200
1475 012116 010301
1476 012120 004737 014242
1477 012124
1478 012124 104423
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490 012126
1491 012126 004737 010274
1492 012132 016501 000002
1493 012136 004737 006024
1494 012142
1495 012142 104423
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508 012144
1509 012144 012700 000007
1510 012150 005737 002226
1511 012154 001402
1512 012156 012700 000010
1513 012162 004737 014552
1514 012166
1515 012166 104423
1516
1517
1518

      BGNMSG PKTMES
PKTMES:
      JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR
      MOV     R2,R0           ;LOW ORDER ADDRESS
      MOV     R3,R1           ;HIGH ORDER ADDRESS
      JSR     PC,PRMESS      ;PRINT THE MESSAGE BUFFER
      ENDMSG
L10007:
      TRAP    CMSG
      .SBTTL  ADDSSR - PRINT TEST ADDRESS AND TSSR
      *
      ;PRINT ROUTINE TO PRINT THE CONTENTS OF
      ;TSSR AND A MEMORY TEST ADDRESS
      ;
      ;INPUTS:
      ;
      ;      R5      FIRST DEVICE UNIBUS ADDRESS
      ;      ERRHI   HIGH ORDER MEMORY TEST ADDRESS
      ;      ERRLO   LOW ORDER MEMORY TEST ADDRESS
      ;
      BGNMSG ADDSSR
ADDSSR:
      JSR     PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
      MOV     TSSR(R5),R1     ;GET CURRENT TSSR
      JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
      ENDMSG
L10010:
      TRAP    CMSG
      .SBTTL  MSGEXP - PRINT WRITE CHAR. EXPD-RECV MESSAGE BUFFERS
      *
      ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
      ;
      ;IMPLICIT INPUTS:
      ;
      ;      EXPMSG   - EXPECTED MESSAGE BUFFER
      ;      RECMMSG  - RECEIVED MESSAGE BUFFER
      ;      RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
      ;      RCVLOAD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
      ;
      BGNMSG MSGEXP
MSGEXP:
      MOV     #7,R0           ;ASSUME NO EXT FEATURES
      TST     EXTFEA          ;EXT FEATURES SET?
      BEQ     5$,             ;BR IF NO
      MOV     #8,R0           ;EXT FEATURE BUFFER IS 8 WORDS
      JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
      ENDMSG
5$:
L10011:
      TRAP    CMSG
      .SBTTL  FIFEXP - PRINT FIFO EXP/RECV DATA
      *
      ;PRINT ROUTINE TO PRINT FIFO EXP/RECV DATA

```

F5

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 16-9

SEQ 0057

FIFEXP - PRINT FIFO EXP/RECV DATA

```

1519
1520      R1      - BYTE COUNT
1521
1522      IMPLICIT INPUTS:
1523
1524      EXPMSG   - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
1525      RECMMSG  - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
1526
1527      BGNMSG   FIFEXP
1528      FIFEXP::
1529      PRINTX   #FIF1MSG,R1      ;PRINT BYTES TRANSFERRED
1530      MOV      R1,-(SP)
1531      MOV      #FIF1MSG,-(SP)
1532      MOV      #2,-(SP)
1533      MOV      SP,R0
1534      TRAP     C#PNTX
1535      ADD      #6,SP
1536      PRINTX   #FIF2MSG      ;PRINT HEADER MSG
1537      MOV      #FIF2MSG,-(SP)
1538      MOV      #1,-(SP)
1539      MOV      SP,R0
1540      TRAP     C#PNTX
1541      ADD      #4,SP
1542      MOV      R1,R0      ;GET BYTE COUNT
1543      JSR      PC,PRBYTEXP   ;PRINT FIFO BYTES IN ERROR
1544      ENDMSG
1545
1546      L10012:
1547      TRAP     C#MSG
1548      .ASCIZ   '***** NUMBER OF BYTES TRANSFERRED = *D2*'
1549      .ASCIZ   '***** FIFO DATA BYTES IN ERROR:'
1550      .EVEN
1551      .SBTTL   MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
1552
1553      PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
1554
1555      IMPLICIT INPUTS:
1556
1557      EXPMSG   - EXPECTED MESSAGE BUFFER
1558      RECMMSG  - RECEIVED MESSAGE BUFFER
1559      RCVHIADD - RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1560      RCVLOADD - RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1561
1562      BGNMSG   MSGSTAT
1563      MSGSTAT::
1564      MOV      #STATCOD,R1      ;ASCII ADDRESS TABLE
1565      MOV      (R1)+,R0      ;DONE ALL MSG LINES?
1566      BEQ      20#      ;BR IF YES
1567      PRINTX   R0      ;PRINT STATUS BIT NAMES
1568      MOV      R0,-(SP)
1569      MOV      #1,-(SP)
1570      MOV      SP,R0
1571      TRAP     C#PNTX
1572      ADD      #4,SP
1573      BR       10#      ;DO ANOTHER MSG LINE
1574      MOV      #10,,R0      ;NUMBER OF WORDS IN A READ STATUS BUFFER
1575
1519 012170
1520 012170
1521 012170 010146
1522 012172 012746 012242
1523 012176 012746 000002
1524 012202 010600
1525 012204 104415
1526 012206 062706 000006
1527 012212
1528 012212 012746 012311
1529 012216 012746 000001
1530 012222 010600
1531 012224 104415
1532 012226 062706 000004
1533 012232 010100
1534 012234 004737 015122
1535 012240
1536 012240 104423
1537 012242 045 116 045 FIF1MSG:
1538 012311 045 116 045 FIF2MSG:
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549 012350
1550 012350 012701 012412
1551 012354 012100
1552 012356 001410
1553 012360
1554 012360 010046
1555 012362 012746 000001
1556 012366 010600
1557 012370 104415
1558 012372 062706 000004
1559 012376 000766
1560 012400 012700 000012

```

MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS

```

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

```

H5

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 16-11

SEQ 0059

MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER

```

1603      ;
1604      ;
1605      ;IMPLICIT INPUTS:
1606      ;
1607      ;      EXPMSG - EXPECTED MESSAGE BUFFER
1608      ;      RECMMSG - RECEIVED MESSAGE BUFFER
1609      ;      PCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1610      ;      RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1611      ;
1612      ;      BGNMSG MSGSUB
1613      MSGSUB:  ;
1614      ;      MOV      #10.,R0      ;SIZE OF WRITE SUBSYSTEM BUFFER
1615      ;      JSR      PC,PRMSGEXP  ;PRINT EXPD/RCV MESSAGE BUFFERS
1616      ;      ENDMSG
1617      ;
1618      ;L10015:
1619      ;      TRAP      C#MSG
1620      ;
1621      ;      .SBTTL  MEMADD - PRINT MEMORY ADDRESS DATA ERROR
1622      ;
1623      ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
1624      ;
1625      ;IMPLICIT INPUTS:
1626      ;
1627      ;      ERRHI    - MEMORY ERROR HIGH ORDER ADDRESS
1628      ;      ERRLO    - MEMORY ERROR LOW ORDER ADDRESS
1629      ;      EXP      - EXPECTED DATA
1630      ;      RECV     - RECEIVED DATA
1631      ;
1632      ;      BGNMSG  MEMADD
1633      MEMADD:  ;
1634      ;      JSR      PC,PRIADD      ;PRINT MEMORY ADDRESS IN ERROR
1635      ;      MOV      EXPD,R1        ;GET EXPD DATA
1636      ;      MOV      RECV,R2        ;GET RECEIVED DATA
1637      ;      JSR      PC,PRIXOR      ;PRINT EXPD/RCV
1638      ;      ENDMSG
1639      ;
1640      ;L10016:
1641      ;      TRAP      C#MSG
1642      ;      .SBTTL  PRAMPKT - PRINT RAM AND PACKET DATA
1643      ;
1644      ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1645      ;WHEN THE RAM DATA DOES NOT MATCH.
1646      ;
1647      ;INPUTS:
1648      ;
1649      ;      R4        POINTER TO COMMAND PACKET
1650      ;
1651      ;IMPLICIT INPUTS:
1652      ;
1653      ;      RAMDATA   DATA AS READ FROM THE RAM
1654      ;      RAMSIZ    NUMBER OF BYTES IN PACKET
1655      ;                  IF RAMSIZ=0 THEN DEFAULT TO 8.
1656      ;
1657      ;IMPLICIT OUTPUTS:
1658      ;
1659      ;      RAMSIZ    SET TO 0

```

PRAMPKT - PRINT RAM AND PACKET DATA

```

1654
1655
1656 013776
1657 013776
1658 014002 012701 002242
1659 014006 005002
1660 014010 122124
1661 014012 001005
1662 014014
1663 014024 000436
1664 014026 116105 177777
1665 014032 116403 177777
1666 014036
1667 014046 042703 177400
1668 014052 116137 177777 002234
1669 014060 116437 177777 002232
1670 014066
      014066 010346
      014070 013746 002232
      014074 013746 002234
      014100 010246
      014102 012746 014156
      014106 012746 000005
      014112 010000
      014114 104414
      014116 062706 000014
1671 014122 005202
1672 014124 005737 002302
1673 014130 001404
1674 014132 020237 002302
1675 014136 003724
1676 014140 000403
1677 014142 020227 000010
1678 014146 002720
1679 014150 005037 002302
1680 014154 000207
1681
1682 014156 045 116 045 RAMASC: .ASCIZ '##A BYTE: #D2#A RAM: #03#A Packet: #03#A XOR:#03'
1683 .EVEN
1684 .SBTTL PRMESS - PRINT CONTENTS OF MESSAGE BUFFER
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701 014242

```

```

; -
PRAMPKT:
  SAVREG
  MOV  #RAMDATA,R1
  CLR  R2
  5$:  CMPB (R1)+,(R4)+
       BNE 7$,NOTSSR
       FORCERROR 7$,NOTSSR
  BR 10$
  7$:  MOVB -1(R1),R5
       MOVB -1(R4),R3
       XOR  R5,R3
       BIC  #177400,R3
       MOVB -1(R1),RECV
       MOVB -1(R4),EXPD
       PRINTB #RAMASC,R2,RECV,EXPD,R3
       MOV  R3,-(SP)
       MOV  EXPD,-(SP)
       MOV  RECV,-(SP)
       MOV  R2,-(SP)
       MOV  #RAMASC,-(SP)
       MOV  #5,-(SP)
       MOV  SP,R0
       TRAP C#PNTB
       ADD  #14,SP
  10$: INC  R2
       YST  RAMSIZ
       BEQ  15$
       CMP  R2,RAMSIZ
       BLE  5$
       BR 25$
  15$: CMP  R2,#8.
  20$: BLT  5$
  25$: CLR  RAMSIZ
       RTS  PC
;SAVE R1-R5 UNTIL NEXT RETURN
;DATA FROM THE RAM
;INIT BYTE NUMBER
;COMPARE EXPECTED, RECEIVED
;BR IF NO MATCH
;SSD
;GET RECV RAM DATA
;GET EXPD PACKET DATA
;XOR EXPD/RECV
;LOW BYTE ONLY
;GET RECEIVED RAM DATA
;GET EXPECTED RAM DATA
;UPDATE BYTE COUNT
;DEFAULT TO 8.?
;BR IF YES
;DONE ALL BYTES?
;BR IF NO
;
;DONE DEFAULT NUMBER OF BYTES?
;BR IF NO
;SET DEFAULT RAMSIZ
;RETURN

```

```

; *
; THIS ROUTINE PRINTS THE CONTENTS OF
; THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE
; TSV-05.
;
; INPUT:
;
;      R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
;      R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
;      NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
;
; THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
;
; -
PRMESS:

```

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```

1702 014242 SAVREG ;SAVE THE REGISTERS
1703 014246 010005 MOV R0,R5 ;SAVE LOW ORDER ADDRESS
1704 014250 005737 003134 TST KTENABLE ;ADDRESS ABOVE 28K?
1705 014254 001001 BNE 10$ ;BR IF YES
1706 014256 005001 CLR R1 ;SET HIGH ORDER ADDRESS TO 0
1707 014260 010103 10$: MOV R1,R3 ;SAVE HIGH ORDER ADDRESS
1708 014262 006100 ROL R0 ;SHIFT BIT15 TO C BIT
1709 014264 006101 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
1710 014266 PRINTX @PROASC,R1,R5 ;PRINT MESSAGE BUFFER ADDRESS
      014266 010546 MOV R5,-(SP)
      014270 010146 MOV R1,-(SP)
      014272 012746 014420 MOV @PROASC,-(SP)
      014276 012746 000003 MOV @3,-(SP)
      014302 010600 MOV SP,R0
      014304 104415 TRAP C:PNTX
      014306 062706 000010 ADD @10,SP
1711 014312 PRINTX @PRIASC ;PRINT HEADER FOR CONTENTS
      014312 012746 014465 MOV @PRIASC,-(SP)
      014316 012746 000001 MOV @1,-(SP)
      014322 010600 MOV SP,R0
      014324 104415 TRAP C:PNTX
      014326 062706 000004 ADD @4,SP
1712 014332 005004 CLR R4 ;NUMBER OF THE NEXT WORD
1713 014334 010501 MOV R5,R1 ;COPY LOW ORDER ADDRESS
1714 014336 010300 MOV R3,R0 ;COPY HIGH ORDER ADDRESS
1715 014340 001403 BEQ 20$ ;BR IF NOT ABOVE 28K
1716 014342 004737 017316 JSR PC,SETMAP ;SETUP PAR ADDRESS IN R0
1717 014346 010005 MOV R0,R5 ;GET PAR FORMAT ADDRESS ABOVE 28K
1718 014350 20$: PRINTX @PRASC,R4,(R5)+ ;PRINT THE CONTENTS OF MEMORY BUFFER
      014350 012546 MOV (R5),-(SP)
      014352 010446 MOV R4,-(SP)
      014354 012746 014523 MOV @PRASC,-(SP)
      014360 012746 000003 MOV @3,-(SP)
      014364 010600 MOV SP,R0
      014366 104415 TRAP C:PNTX
      014370 062706 000010 ADD @10,SP
1719 014374 005204 INC R4 ;NUMBER OF THE NEXT
1720 014376 020427 000007 CMP R4,@7 ;DONE ALL YET ?
1721 014402 003005 BGT 50$ ;BRANCH IF ALL DONE
1722 014404 002761 BLT 20$ ;PRINT FIRST 7 WORDS
1723 014406 032763 000200 000012 BIT @X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
1724 014414 001355 BNE 20$ ;PRINT EXTENDED STATUS WORD
1725 014416 000207 50$: RTS PC ;RETURN
1726
1727 014420 045 116 045 PROASC: .ASCIZ 'NMA Message Buffer Address = #01#05'
1728 014465 045 116 045 PRIASC: .ASCIZ 'NMA Message Buffer Contents:'
1729 014523 045 116 045 PRASC: .ASCIZ 'NMA Word#D1#A: #0'
1730 .EVEN
1731 .SBTTL PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
1732
1733
1734 ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
1735
1736 ; PO - NUMBER OF WORDS IN BUFFER
1737
1738 ;IMPLICIT INPUTS:
1739

```

K5

PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS

```

1740      ; EXPMSG - EXPECTED MESSAGE BUFFER
1741      ; RECMMSG - RECEIVED MESSAGE BUFFER
1742      ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1743      ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1744      ;
1745      PRMSGEXP::
1746      SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
1747      MOV R0,R5                            ;SAVE NUMBER OF WORDS
1748      MOV RCVLOADD,R0                      ;GET RECV LOW ADDRESS
1749      MOV R0,R4                            ;COPY LOW ADDRESS
1750      MOV RCVHIADD,R1                      ;GET RECV HIGH ADDRESS
1751      ROL R0                               ;SHIFT BIT15 TO C BIT
1752      ROL R1                               ;SHIFT TO HIGH ORDER FOR PRINTOUT
1753      PRINTX @PRMSG0,R1,R4                ;PRINT MESSAGE BUFFER ADDRESS
1754      MOV R4,-(SP)
1755      MOV R1,-(SP)
1756      MOV @PRMSG0,-(SP)
1757      MOV @3,-(SP)
1758      MOV SP,R0
1759      TRAP C:PNTX
1760      ADD @10,SP
1761      PRINTX @PRMSG1                      ;PRINT HEADER FOR CONTENTS
1762      MOV @PRMSG1,-(SP)
1763      MOV @1,-(SP)
1764      MOV SP,R0
1765      TRAP C:PNTX
1766      ADD @4,SP
1767      CLR R4                               ;NUMBER OF THE CURRENT WORD
1768      MOV @EXPMSG,R1                       ;GET EXPD BUFFER ADDRESS
1769      MOV @RECMMSG,R2                      ;GET RECV BUFFER ADDRESS
1770      MOV (R1),R0                          ;GET EXPD
1771      MOV (R2),R3                          ;GET RECV
1772      XOR R0,R3                            ;XOR EXPD/RECV
1773      PRINTX @PRMSG2,R4,(R1)+,(R2)+,R3
1774      MOV R3,-(SP)
1775      MOV (R2)+,-(SP)
1776      MOV (R1)+,-(SP)
1777      MOV R4,-(SP)
1778      MOV @PRMSG2,-(SP)
1779      MOV @5,-(SP)
1780      MOV SP,R0
1781      TRAP C:PNTX
1782      ADD @14,SP
1783      INC R4                               ;NUMBER OF THE NEXT
1784      CMP R4,R5                            ;DONE ALL YET?
1785      BGE 50$                              ;BR IF YES
1786      BR 20$                              ;DO ANOTHER
1787      RTS PC                              ;RETURN
1788      20$:
1789      50$:
1790      045 PRMSG0: .ASCIZ '##A Message Buffer Address - #01#05'
1791      045 PRMSG1: .ASCIZ '##A Message Buffer Contents:'
1792      045 PRMSG2: .ASCIZ '##A WORD #D2#A EXPD: #06#A RECV: #06#A XOR: #06'
1793      .EVEN
1794      .SBTTL PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
1795      ;*
1796      ;
1797      ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS

```

PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

```

1776      ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
1777
1778      RO      - NUMBER OF BYTES IN BUFFER
1779
1780      IMPLICIT INPUTS:
1781
1782      EXPMSG   - EXPECTED MESSAGE BUFFER
1783      RECMMSG  - RECEIVED MESSAGE BUFFER
1784
1785      PRBYTEXP::
1786      SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
1787      MOV      RO,R5                        ;SAVE NUMBER OF BYTES
1788      CLR      PRMNO                        ;INIT ERROR COUNT
1789      CLR      R4                          ;NUMBER OF THE CURRENT BYTE
1790      MOV      @EXPMSG,R1                   ;GET EXPD BUFFER ADDRESS
1791      MOV      @RECMMSG,R2                  ;GET RECV BUFFER ADDRESS
1792      MOV      (R1),R0                      ;GET EXPD BYTE
1793      BIC      @C<377>,R0                   ;CLEAR UPPER BYTE
1794      MOV      R0,PRBEXP                    ;SAVE FOR ERROR REPORT
1795      MOV      (R2),R3                      ;GET RECV BYTE
1796      BIC      @C<377>,R3                   ;CLEAR UPPER BYTE
1797      MOV      R3,PRBREC                    ;FOR ERROR REPORT
1798      XOR      R0,R3                        ;XOR EXPD/RECV
1799      CMP      (R1)+,(R2)+                  ;EXPD = RECV?
1800      BEQ      30$                          ;BR IF YES
1801      INC      PRMNO                        ;UPDATE ERROR COUNT
1802      CMP      PRMNO,@8.                    ;PRINTED 8?
1803      BHI      30$                          ;BR IF YES
1804      PRINTX   @PRBMSG,R4,PRBEXP,PRBREC,R3
1805      MOV      R3,-(SP)
1806      MOV      PRBREC,-(SP)
1807      MOV      PRBEXP,-(SP)
1808      MOV      R4,-(SP)
1809      MOV      @PRBMSG,-(SP)
1810      MOV      @5,-(SP)
1811      MOV      SP,R0
1812      TRAP     C#PNTX
1813      ADD      @14,SP
1814      FORCEEXIT 50$                          ;000
1815      BR       35$                          ;000
1816      30$:     FORCEERROR 27$,NOTSSR          ;000
1817      35$:     INC      R4                    ;NUMBER OF THE NEXT
1818      CMP      R4,R5                        ;DONE ALL YET?
1819      BGE      50$                          ;BR IF YES
1820      BR       20$                          ;DO ANOTHER
1821      50$:     PRINTX   @PRBTOT,PRMNO        ;PRINT TOTAL ERROR COUNT
1822      MOV      PRMNO,-(SP)
1823      MOV      @PRBTOT,-(SP)
1824      MOV      @2,-(SP)
1825      MOV      SP,R0
1826      TRAP     C#PNTX
1827      ADD      @6,SP
1828      RTS      PC                          ;RETURN
1829
1830      045      PRBMSG: .ASCIZ ' *N*A    BYTE *D2*A    EXPD: *03*A    RECV: *03*A    XOR: *03'

```

PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

```

1818 015423      045      116      045 PRBTOT: .ASCIZ  'NWA NUMBER OF BYTES IN ERROR = %D2'
1819                                     .EVEN
1820 015470      000000      PRBEXP: .WORD  0                      ;EXPD
1821 015472      000000      PRBREC: .WORD  0                      ;RECV
1822                                     .SBTTL  EXPREC - PRINT EXPD/RECV WORD DATA
1823                                     ;+
1824                                     ;
1825                                     ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1826                                     ;
1827                                     ;INPUTS:
1828                                     ;
1829                                     ;      R1      RECEIVED DATA
1830                                     ;      R2      EXPECTED DATA
1831                                     ;
1832                                     ;-
1833
1834 015474      BGNMSG  EXPREC
1835 015474      004737  007742  EXPREC:: JSR      PC,PRI XOR          ;PRINT THE DATA
1836 015500      ENDMMSG
1837 015500      L10017: TRAP      C#MSG
1838 015500      104423      .SBTTL  EXPBREC - PRINT EXPD/RECV BYTE DATA
1839                                     ;+
1840                                     ;
1841                                     ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
1842                                     ;
1843                                     ;INPUTS:
1844                                     ;
1845                                     ;      R1      RECEIVED DATA BYTE
1846                                     ;      R2      EXPECTED DATA BYTE
1847                                     ;
1848                                     ;-
1849
1850 015502      BGNMSG  EXPBREC
1851 015502      004737  007612  EXPBREC:: JSR      PC,PRI XOR          ;PRINT THE DATA
1852 015506      ENDMMSG
1853 015506      L10020: TRAP      C#MSG
1854 015506      104423
1855                                     .SBTTL  RAMERR - PRINT RAM AND PACKET DATA
1856                                     ;+
1857                                     ;
1858                                     ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1859                                     ;
1860                                     ;INPUTS:
1861                                     ;
1862                                     ;      R4      POINTER TO COMMAND PACKET
1863                                     ;
1864                                     ;IMPLICIT INPUTS:
1865                                     ;
1866                                     ;      RAMDATA  DATA AS READ FROM THE RAM
1867                                     ;      RAMSIZ   NUMBER OF BYTES IN PACKET
1868                                     ;              IF RAMSIZ=0 THEN DEFAULT TO 8.

```

RAMERR PRINT RAM AND PACKET DATA

```

1869
1870      ;IMPLICIT OUTPUTS:
1871      ;
1872      ;      RAMSIZ  SET TO 0
1873      ;
1874      ;
1875      015510      BGNMSG  RAMERR
1876      015510      004737  013776      RAMERR:;
1877      015514      JSR      PC,PRAMPKT      ;PRINT RAM/PACKET DATA
1878      015514      ENDMSG
1879      015514      L10021:
1880      015514      TRAP      C#MSG
1881      ;
1882      ;      .SBTTL  RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
1883      ;
1884      ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1885      ;
1886      ;INPUTS:
1887      ;
1888      ;      R4      POINTER TO COMMAND PACKET
1889      ;
1890      ;IMPLICIT INPUTS:
1891      ;
1892      ;      RAMDATA      DATA AS READ FROM THE RAM
1893      ;      RAMSIZ      NUMBER OF BYTES IN PACKET
1894      ;                  IF RAMSIZ=0 THEN DEFAULT TO 8.
1895      ;      ERRHI      HIGH ORDER TEST ADDRESS
1896      ;      ERRLO      LOW ORDER TEST ADDRESS
1897      ;
1898      ;IMPLICIT OUTPUTS:
1899      ;
1900      ;      RAMSIZ  SET TO 0
1901      ;
1902      015516      BGNMSG  RAMTADD
1903      015516      004737  010274      RAMTADD:;
1904      015522      JSR      PC,PRITADD      ;PRINT TEST ADDRESS
1905      015522      JSR      PC,PRAMPKT      ;PRINT RAM/PACKET DATA
1906      015526      ENDMSG
1907      015526      L10022:
1908      015526      TRAP      C#MSG
1909      ;
1910      ;      .SBTTL  RAMEXP - PRINT RAM EXPD/RECV DATA
1911      ;
1912      ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1913      ;
1914      ;INPUTS:
1915      ;
1916      ;      R1      RECEIVED DATA
1917      ;      R2      EXPECTED DATA
1918      ;      R4      CONTROLLER RAM ADDRESS
1919      ;
1920      015530      BGNMSG  RAMEXP
1921      015530      RAMEXP:;

```

B6

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 16-18

SEQ 0066

RAMEXP - PRINT RAM EXPD/RECV DATA

```

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

```

C6

SEQ 0067

BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

```

015722 104414
015724 062706 000006
1961 015730 012602
1962 015732 004737 006024
1963 015736
015736
015736 104423
1964 015740 045 116 045 XFERASC:
1965
1966
1967
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 015774
2000 015774
2001 016000 012765 000000 000002
2002 016006 004737 016250
2003 016012 016500 000002
2004 016016 010004
2005 016020 042704 176277
2006 016024 052704 002200
2007 016030 020400
2008 016032 001402
2009 016034 000241
2010 016036 000401
2011 016040 000261
2012 016042 000207
2013

TRAP C@PNT@
ADD @6,SP
MOV (SP),R2 ;RESTORE R2
JSR PC,PRITSSR ;DECODE TSSR CONTENTS
ENDMSG

L10025:
TRAP C@MSG
.ASCIZ 'DMA Data Transferred = %03'
.SBTTL GLOBAL SUBROUTINES SECTION

***
; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
; THAT ARE USED IN MORE THAN ONE TEST.
***
.SBTTL SOFINIT - SOFT INITIALIZE OF CONTROLLER

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

SOFINIT:
SAVREG ; SAVE THE REGISTERS
MOV @0,TSSR(R5) ; DO THE INIT.
JSR PC,WAITF ; WAIT FOR SSR
MOV TSSR(R5),R0 ;GET THE TSSR REGISTER
MOV R0,R4 ;TSSR CONTENTS
BIC @C<HIADDR!OFL>,R4
BIS @SSR!NBA,R4 ;R4 HAS EXPECTED CONTENTS
CMP R4,R0 ;ONLY EXPECTED BITS SET ?
BEQ 51 ;BRANCH IF OKAY
CLC ;CLEAR THE CARRY FOR ERROR
BR 101 ;GO TO EXIT
51: SEC ;SET THE CARRY BIT
101: RTS PC ;RETURN TO CALLER
.SBTTL CHKAMB - CHECK TSSR FOR AMBIGUITY

```

D6

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 16-20

SEQ 0068

CHKAMB - CHECK TSSR FOR AMBIGUITY

```

2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033 016044
2034 016044
2035 016050 010004
2036 016052 032700 100000
2037 016056 001004
2038 016060 032700 174077
2039 016064 001023
2040 016066 000424
2041 016070 032700 000200
2042 016074 001011
2043 016076 032700 000040
2044 016102 001414
2045 016104 042704 177761
2046 016110 020427 000016
2047 016114 001007
2048 016116 000410
2049 016120 032700 000040
2050 016124 001405
2051 016126 032700 000006
2052 016132 001002
2053 016134 000241
2054 016136 000401
2055 016140 000261
2056 016142 000207
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066 000200
2067 000001
2068
2069
2070 016144 000

;
;
; THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
; FOR AMBIGUITY
;
; INPUT:
;
; R0      CONTENTS OF TSSR
;
; OUTPUT:
;
; R0      CONTENTS OF TSSR
;
; CARRY   SET - NO AMBIGUITY
;         CLR - AMBIGUOUS CONTENTS
;
;
CHKAMB:
    SAVREG                ;SAVE THE GENERAL REGISTERS
    MOV     R0,R4          ;CONTENTS OF TSSR
    BIT     @SC,R0         ;IS BIT 15 SET ?
    BNE     50             ;BRANCH IF YES
    BIT     @C<NBA!OFL!SSR!HIADDR>,R0 ;ANY OTHER BITS SET ?
    BNE     40             ;MUST BE AN ERROR
    BR      45             ;RETURN WITH SUCCESS
    BIT     @SSR,R0        ;IS READY BIT SET ?
    BNE     10             ;BRANCH IF READY BIT IS SET.
    BIT     @BIT5,R0       ;IS FATAL ERROR BIT SET ?
    BEQ     40             ;ERROR IF NOT
    BIC     @CTERCLS,R4    ;CLEAR ALL BUT TERMINATION CODE
    CMP     R4,@16         ;ALL THREE BITS MUST BE SET
    BNE     40             ;ERROR IF NOT SET
    BR      45             ;OK IF ALL ARE SET
    BIT     @BIT5,R0       ;IS FATAL ERROR BIT SET ?
    BEQ     45             ;ERROR IF BIT IS SET WITH SSR
    BIT     @BIT2!BIT1,R0 ;IS THIS A FUNCTION REJECT
    BNE     45             ;BR. IF TSSR IS OK
    CLC                     ;AMBIGUOUS CONTENTS
    BR      50             ;
    SEC                     ;SHOW SUCCESS - NO AMBIGUITY
    RTS     PC              ;RETURN TO CALLER
    SBTTL   ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS
;
; DEFAULT DISPLAY INTERRUPT HANDLERS.
; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
;
; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
;
; IOKCKIN=BIT7    ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
; IOKSTP=BIT0     ; EXPECT "STOP" INTERRUPT.
;
; INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
INTMASK:         .BYTE 0

```

E6

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 16-21

SEQ 0069

ENAIN,DSBINT - ENABLE/DISABLE INTERRUPTS

```

2071                                ;INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
2072 016145      000                INTFLAG: .BYTE 0
2073
2074                                ;SAVED INTERRUPT VECTOR:
2075 016146      000000             INTVEC: .WORD 0
2076                                ;SAVE CPU PC
2077 016150      000000             INTCPC: .WORD 0
2078
2079                                ;SUBROUTINE TO ENABLE INTERRUPTS:
2080 016152      010046             ENAIN: MOV RO, -(SP) ;SAVE RO
2081 016154      013700      002210      MOV IVEC, RO ;GET POINTER TO VECTORS
2082 016160      012720      016216      MOV @INTR, (RO)+ ;SET UP INTERRUPT VECTOR
2083 016164      012720      000300      MOV @PRIORITY, (RO)+
2084 016170      012600      MOV (SP)+, RO ;RESTORE RO
2085 016172      011646      MOV (SP), -(SP)
2086 016174      012766      000000      000002      MOV @0,2(SP) ;SET CPU TO LEVEL 0
2087 016202      000002      RTI
2088
2089                                ;SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 6)
2090 016204      011646             DSBINT: MOV (SP), -(SP)
2091 016206      012766      000300      000002      MOV @PRIORITY,2(SP)
2092 015214      000002      RTI
2093                                .SBTTL INTR - INTERRUPT HANDLERS
2094
2095 016216      BGNSRV INTR ;DEFINE INTERRUPT ENTRY
2096 016216      INTR::
2097 016216      012737      000001      002224      MOV @1,INTRECV ;SET FLAG TO SHOW INTERRUPT RECEIVED
2098 016224      105037      016145      CLR B INTFLAG ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2099 016230      132737      000001      016144      BIT B @IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
2100 016236      001003      BNE 11 ;BR IF YES
2101 016240      152737      000001      016145      BIS B @IOKSTP,INTFLAG ;NO, SET THE ERROR FLAG.
2102
2103                                ;SAVE REGISTERS, MSG BUFFER, ETC.
2104 016246      11:
2105 016246      ENDSRV
2106 016246      L10026:
2107 016246      000002      RTI
2108                                .SBTTL WAITF - WAIT FOR SUBSYSTEM READY
2109
2110                                ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
2111                                ; INPUTS:
2112                                ; R5 ADDRESS OF FIRST DEVICE REGISTER
2113                                ; OUTPUTS:
2114                                ; R0 CONTENTS OF LAST TSSR READ
2115                                ; CARRY SET - READY BIT SET
2116                                ; CLR - TIMEOUT WAITING FOR READY
2117
2118                                ; WAITF:: BR 11 ;NOP WHEN SUPER FIXED
2119 016250      000401      BREAK ;DO A SUPVSR BREAK FIRST.
2120 016252      104422      TRAP C#BHK
2121 016254      012746      011000      11: MOV @11000, -(SP) ;25-APRIL-83 REV B - 1100 MSEC TIMER
2122 016260      016500      000002      21: MOV TSSR(R5), R0 ;READ THE TSSR REGISTER
2123 016264      105700      TST B R0 ;TEST FOR READY BIT SET

```

F6

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 16-22

SEQ 0070

WAITF - WAIT FOR SUBSYSTEM READY

```

2124
2125 016266 100420      BMI      3$          ; EXIT ON STOP FLAG.
2126 016270      DELAY    1              ; WAIT 100 USEC
      016270 012727 000001      MOV     $1,(PC)+
      016274 000000      .WORD    0
      016276 013727 002116      MOV     L$DLY,(PC)+
      016302 000000      .WORD    0
      016304 005367 177772      DEC     -6(PC)
      016310 001375      BNE     .-4
      016312 005367 177756      DEC     -22(PC)
      016316 001367      BNE     .-20
2127 016320 005316      DEC     (SP)      ;REDUCE DELAY COUNT
2128 016322 001356      BNE     2$      ;RETRY UNTIL TIMER EXPIRES
2129 016324 000241      CLC              ; C = 0, CONTROLLER STILL RUNNING...
2130 016326 000401      BR       4$      ;...OR HUNG-UP AFTER 300 MSEC.
2131 016330 000261      3$: SEC          ; C = 1, CONTROLLER IS STOPPED.
2132 016332 005326      4$: DEC     (SP)+ ;RESTORE STACK WITHOUT CHANGING CARRY BIT
2133 016334 000207      RTS     PC
2134      .SBTTL  CHKTTSSR - CHECK TSSR FOR READY
2135
2136      ;+
2137      ;
2138      ;THIS ROUTINE WAITS FOR READY IN THE TSSR
2139      ;AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
2140      ;
2141      ;INPUT:
2142      ;
2143      ;      R5      ADDRESS OF CSR REGISTERS
2144      ;
2145      ;OUTPUT:
2146      ;
2147      ;      R0      CONTENTS OF TSSR
2148      ;      CARRY   SET - OKAY
2149      ;              CLR - NOT READY AMBIGUOUS, OR SC SET
2150      ;
2151      ;-
2152
2153 016336      CHKTTSSR:
2154 016336 004737 016250      JSR     PC,WAITF      ;WAIT FOR READY
2155 016342 103014      BCC     20$      ;BRANCH IF TIME OUT
2156 016344 004737 016044      JSR     PC,CHKAMB     ;TSSR AMBIGUOUS?
2157 016350 103006      BCC     10$      ;BR IF YES
2158 016352 032700 100000      BIT     $SC,R0      ;SPECIAL CONDITION SET?
2159 016356 001405      BEQ     15$      ;BR IF NO
2160 016360 032700 074000      BIT     $<SCE!BIE!RMR!NXM>,R0 ;ANY ERROR BITS SET?
2161 016364 001402      BEQ     15$      ;BR IF NO
2162 016366 000241      10$: CLC              ;SET FAILURE
2163 016370 000401      BR       20$      ;
2164 016372 000261      15$: SEC              ;SET SUCCESS
2165 016374 000207      20$: RTS     PC      ;RETURN TO CALLER
2166      .SBTTL  XNXM      - CHECK FOR NONEXISTENT MEMORY
2167
2168      ;+
2169      ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
2170      ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
2171      ; "C" = 0, ALL ADDRESSES OK.
2172      ;
2173      ;CALL:  MOV ADR1,R1

```

G6

SEQ 0071

XNYM - CHECK FOR NONEXISTENT MEMORY

```

2173      |      MOV ADR2,R2
2174      |      JSR PC,NXM
2175      |      RETURN      |TEST "C" AND PROCEED.
2176      |
2177 016376 012737 016430 000004 XNXM: MOV    #21,804      | SET BUSERR VECTOR.
2178 016404 012737 000200 000006      MOV    #PRI04,806
2179 016412 005003      CLR     R3      | FLAG.
2180 016414 005711 1$: TST     (R1)      | TEST THE ADDRESS(ES).
2181      |      |IF ANY TRAP, CONTINUE AT 2$.
2182 016416 020102      CMP     R1,R2      | OTHERWISE, CONTINUE HERE.
2183 016420 001407      BEQ     3$      | BR IF FINISHED (NO NEXM'S).
2184 016422 062701 000002      ADD     #2,R1      | SET NEXT ADDRESS...
2185 016426 000772      BR      1$      | ...AND CONTINUE.
2186      |
2187 016430 005103 2$: COM     R3      | GOT ONE, SET FLAG...
2188 016432 012716 016440      MOV     #3$,(SP)
2189 016436 000002      RTI      | ...AND DISMISS INTERRUPT...
2190 016440 3$: CLRVEC  #4      | ...AND GIVE BACK THE VECTOR.
      016440 012700 000004      MOV     #4,R0
      016444 104436      TRAP    C#CVEC
2191 016446 005703      TST     R3      | DID WE CATCH ONE ??
2192 016450 001401      BEQ     .+4      | NO. "C" = 0, SKIP NEXT.
2193 016452 000261      SEC      | YES. "C" = 1, (R1) = NEXM ADDR.
2194 016454 000207      RTS     PC
2195
2196
2197      .SBTTL TSTLOOP - CHECK ITERATION COUNT
2198
2199      |*
2200      | SUBROUTINE TO EXECUTE TEST ITERATIONS.
2201      | EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
2202      | LOOP COUNTER IS SET BY "BEGIN,TEST" MACRO.
2203      |
2204      | CALL: LOOPTO ARG
2205      |
2206 016456 TSTLOOP:: TST     NOITS      | ITERATIONS INHIBITED?
2207 016462      BNE     1$      | YES.
2208 016464      TST     QVP      | NO.
2209 016470      BMI     1$      | LOOPS DISALLOWED IN QUICK PASS.
2210 016472      DEC     LOOPCNT  | BUMP LOOP COUNTER.
2211 016476      BNE     2$
2212 016500 1$: CLC      | LOOP DISALLOWED, OR DONE.
2213 016502      BR      3$
2214 016504 2$: SEC      | LOOP ENABLED.
2215 016506 3$: RTS     PC
2216
2217      .SBTTL TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
2218
2219      |*
2220      | PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
2221      | INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
2222      | IN THE CURRENT RUN SEQUENCE.
2223      | CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
2224      |
2225      | INPUT:
2226      |
2227      | R0      POINTER TO TEST ID ASCIZ STRING

```

H6

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 16-24

SEQ 0072

TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS

```

2228      ;OUTPUT:
2229      ;
2230      ;      R5      ADDRESS OF FIRST DEVICE REGISTER
2231      ;
2232      ;IMPLICIT OUTPUTS:
2233      ;
2234      ;      TSTCNT  UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
2235      ;
2236      ;SIDE EFFECTS:
2237      ;
2238      ;      INTERRUPT LEVEL IS RASIED TO LEVEL OF
2239      ;      THE DEVICE UNDER TEST
2240      ;
2241      ;-
2242
2243      TSTSETUP::
2244      016510      010046      MOV      RO,-(SP)      ;SAVE THE TEST ID MESSAGE
2245      016512      005037      003154      CLR      SIFLAG      ; CLEAR "SOFT INIT" FLAG
2246      016516      005037      016756      CLR      ERRK      ; CLEAR LOCAL ERROR COUNTER.
2247      016522      005037      005772      CLR      EXT A      ; CLEAR ERROR EXTENSION FLAG.
2248      016526      105037      016144      CLRB     INTMASK     ; CLEAR INTERRUPT MASK (CHECK ERROR)
2249      016532      013700      002202      MOV      UNITN,RO     ; GET THE UNIT NUMBER,
2250      016536      006300      ASL      RO      ; ... AND MAKE IT A WORD OFFSET.
2251      016540      005737      003114      TST      NODEV      ; DID STARTUP FIND THE DEVICE?
2252      016544      001450      BEQ      4$      ; BR IF YES
2253      016546      100010      BPL      3$      ; BR IF NOT IDLE
2254      016550      052760      160000      003176      BIS      0160000,ERTABL(RO) ; FLAG ERROR IN THE ERROR TABLE
2255      016556      ERRDF      1,NXR,NXRERR      ; NO DEVICE HERE -- PRINT IT
2256      016556      104455      TRAP      C#ERDF
2257      016560      000001      .WORD     1
2258      016562      003740      .WORD     NXR
2259      016564      005736      .WORD     NXRERR
2260      016566      000407      BR      2$
2261      016570      052760      160001      003176      3$: BIS      0160001,ERTABL(RO) ; FLAG ERROR IN THE ERROR TABLE
2262      016576      104455      ERRDF      2,NOINIT      ; DEVICE NOT IDLE
2263      016600      000002      TRAP      C#ERDF
2264      016602      004335      .WORD     2
2265      016604      000000      .WORD     NOINIT
2266      016606      012737      177777      003112      2$: MOV      0-1,DUFLG      ; DROP THE UNIT
2267      016614      013700      002202      DODU      UNITN
2268      016620      104451      MOV      UNITN,RO
2269      016622      DOCLN      TRAP      C#DODU
2270      016622      104444      DOCLN      ; ABORT THE PASS
2271      016624      000423      TRAP      C#DCLN
2272      016626      104421      BR      5$
2273      016630      032700      001000      4$: RFLAGS      RO      ; GET THE OPERATOR FLAGS.
2274      016634      001412      TRAP      C#RFLA
2275      016636      011600      BIT      0PNT,RO      ; PRINT THE TEST NUMBERS?
2276      016640      010046      BEQ      1$      ; BR IF NO
2277      016642      012746      016704      MOV      (SP),RO      ;GET THE ID MESSAGE
2278      016646      012746      000002      PRINTF      0TNAM,RO      ;DISPLAY THE TEST ID
2279      016652      010600      MOV      RO,-(SP)
2280      MOV      0TNAM,-(SP)
2281      MOV      02,-(SP)
2282      MOV      SP,RO

```

TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS

```

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

```

J6

SEQ 0074

INCERK - INCREMENT LOCAL ERROR COUNT

```

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

```

K6

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 16-27

SEQ 0075

KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT

```

2360      ; SUBROUTINE - DISABLE MEM MGT.
2361
2362 017274 005737 003132      KTOFF: TST      KTFLG      ; GOT KT11?
2363 017300 001405              BEQ      1$          ; NO.
2364 017302 000240              NOP
2365 017304 000240              NOP
2366 017306 012737 000000 177572      MOV      #0,SRO      ; DISABLE KT.
2367 017314 000207      1$: RTS      PC
2368      .SBTTL SETMAP - SETUP PAR6 MAPPING
2369
2370
2371      ;+
2372      ; THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
2373      ; AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
2374      ; IS RETURNED BIASED TO PAR6.
2375      ;
2376      ; INPUTS:
2377      ;
2378      ;      R0      HIGH ORDER ADDRESS BITS
2379      ;      R1      LOW ORDER ADDRESS BITS
2380      ;
2381      ; OUTPUTS:
2382      ;
2383      ;      R0      OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
2384      ;      CARRY   SET IF SUCCESS
2385      ;              CLR IF ERROR
2386      ;
2387      ; SETMAP:
2388      ; SAVREG
2389      ; TST      KTFLG      ; SAVE R1-R4 UNTIL NEXT RETURN
2390      ; BEQ      10$        ; SYSTEM HAVE ABOVE 28K?
2391      ; MOV      R1,R2      ; BR IF NO
2392      ; .REPT    6          ; SAVE LOW ORDER BITS
2393      ; ASR      R0
2394      ; ROR      R1
2395      ; .ENOR
2396      ; BIC      #177,R1    ; ALINE FOR LOWER 4K BOUNDARY
2397      ; CMP      R1,KTFLG   ; HIGHER THAN EXISTING MEMORY?
2398      ; BHIS     10$        ; BR IF YES
2399      ; MOV      R1,#KIPAR6 ; SETUP MAPPING REGISTER PAR6
2400      ; BIC      #160000,R2 ; SETUP DISPLACEMENT IN PAGE
2401      ; ADD      #140000,R2 ; ADD IN PAR6 BIAS
2402      ; MOV      R2,R0      ; RETURN IN R0
2403      ; SEC
2404      ; BR       15$        ; SET SUCCESS
2405      ; CLC
2406      ; RTS      PC        ; SET FAILURE
2407      ; .SBTTL FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN
2408      ;
2409      ; +
2410      ; FILL MEMORY WITH A BACKGROUND PATTERN
2411      ;
2412      ; INPUTS:
2413      ;
2414      ;      R0 - BACKGROUND PATTERN
2415      ;      FREE  - FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2416      ;      KTFLG - SET TO HIGHEST MEMORY LOCATION IF > 28K.

```

FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN

```

2417      ; OUTPUTS:
2418      ;
2419      ;      NONE
2420      ;
2421      ;
2422      ; FILLMEM:
2423      ; SAVREG
2424      ; JSR PC,KTOFF      ;SAVE R1-R5 UNTIL NEXT RETURN
2425      ; MOV R0,R3        ;DISABLE KT.
2426      ; MOV FREE,R1      ;COPY TEST PATTERN
2427      ; MOV FRESIZ,R2    ;GET FIRST FREE LOCATION
2428      ; MOV R3,(R1)+     ;SIZE OF FREE SPACE BELOW 28K.
2429      ; DEC R2           ;STORE A BACKGROUND WORD
2430      ; BGT 10$          ;DONE ALL MEMORY IN FREE SPACE?
2431      ; TST KTFLG        ;BR IF NO
2432      ; BEQ 55$          ; GOT KT?
2433      ; JSR PC,KTON      ; NO. GET OUT.
2434      ; CLR R0           ; YES. ENABLE KT.
2435      ; MOV PST32W,R1    ;HIGH ORDER ADDRESS START
2436      ; .REPT 6          ;GET >28K START ADDRESS (IN 32W BLOCKS)
2437      ; CLC              ;CLEAR C BIT
2438      ; ROL R1            ;CONVERT BLOCKS TO WORDS
2439      ; ROL R0            ;MAKE IT DOUBLE PRECISION
2440      ; .ENDR
2441      ; JSR PC,SETMAP    ;SETUP PAR6 MAPPING REGISTER
2442      ; MOV R3,(R0)+     ;STORE TEST PATTERN IN >28K ADDRESS
2443      ; CMP R0,#160000   ;END OF PAR6 MAPPING AREA?
2444      ; BLO 30$          ;BR IF NO
2445      ; SUB #20000,R0    ;BACKUP INTO PAR6 MAPPING BEGIN
2446      ; ADD #200,#KIPAR6 ;POINT TO NEXT 4K BLOCK >28K.
2447      ; CMP #KIPAR6,KTFLG ;END OF MEMORY?
2448      ; BEQ 50$          ;BR IF YES
2449      ; TST T23A        ;11/23A?
2450      ; BEQ 35$          ;NO KEEP GOING
2451      ; MOV SRO,R4        ;GET SRO CONTENTS
2452      ; BIC #177761,R4   ;CLEAR ALL BUT PAGE NUMBER
2453      ; CMP #16,R4       ;SEE IF PAGE 7
2454      ; BEQ 50$          ;EXIT IF THERE
2455      ; TST T23B        ;11/23B?
2456      ; BEQ 45$          ;NO KEEP GOING
2457      ; CMP #KIPAR6,#7600 ;REACHED 18 BITS?
2458      ; BHIS 40$         ;YES
2459      ; BR 45$           ;NO KEEP GOING
2460      ; MOV #20,SR3      ;SET 22 BIT RELOCATION
2461      ; JMP 30$          ;KEEP GOING ON ETC.
2462      ; JSR PC,KTOFF    ;DISABLE KT.
2463      ; RTS PC
2464      ; .SBTTL CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN
2465      ;
2466      ; COMPARE MEMORY WITH A BACKGROUND PATTERN
2467      ;
2468      ; INPUTS:
2469      ;
2470      ;      R0 = BACKGROUND PATTERN
2471      ;      FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2472      ;      KTFLG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
2473      ;

```

2474					; OUTPUTS:	
2475						
2476					CARRY - SET IF NO ERROR	
2477					CARRY - CLR IF ERROR	
2478						
2479					; IMPLICIT OUTPUTS:	
2480						
2481					ERRHI - ERROR HIGH ADDRESS	
2482					ERRLO - ERROR LOW ADDRESS	
2483					EXPD - EXPECTED DATA	
2484					RCV - RECEIVED DATA	
2485						
2486	017660				CMPMEM:	
2487	017660				SAVREG	;SAVE R1-R5 UNTIL NEXT RETURN
2488	017664	010003			MOV R0,R3	;COPY TEST PATTERN
2489	017666	004737	017274		JSR PC,KTOFF	;DISABLE KT.
2490	017672	013701	003124		MOV FREE,R1	;GET FIRST FREE LOCATION
2491	017676	013702	003126		MOV FPRESIZ,R2	;SIZE OF FREE SPACE BELOW 28K.
2492	017702	020311		10\$:	CMP R3,(R1)	;FREE SPACE LOCATION EQUAL TO EXPD?
2493	017702	001411			BEQ 15\$	;BR IF YES
2494	017706	010137	002240		MOV R1,ERRLO	;SAVE ADDRESS IN ERROR
2495	017712	005037	002236		CLR ERRHI	;NO HIGH ADDRESS
2496	017716	010337	002232		MOV R3,EXPD	;SAVE EXPD FOR ERROR REPORT
2497	017722	011137	002234		MOV (R1),RCV	;SAVE RCV FOR ERROR REPORT
2498	017726	000474			BR 50\$	
2499	017730	005721		15\$:	TST (R1)+	;POINT TO NEXT ADDRESS
2500	017732	005302			DEC R2	;DONE ALL MEMORY IN FREE SPACE?
2501	017734	003362			BGT 10\$	;BR IF NO
2502	017736	005737	003132		TST KTFLG	; GOT KT?
2503	017742	001472			BEQ 55\$	; NO. GET OUT.
2504	017744	004737	017256		JSR PC,KTON	; YES. ENABLE KT.
2505	017750	005000			CLP R0	;HIGH ORDER ADDRESS START
2506	017752	013701	003152		MOV PST32W,R1	;GET >28K START ADDRESS (IN 32W BLOCKS)
2507		000006			REPT 6	
2508					ROL R1	;CONVERT BLOCKS TO WORDS
2509					ROL R0	;MAKE IT DOUBLE PRECISION
2510					ENDR	
2511	020006	042701	000177		BIC #177,R1	;ALINE 4K BOUNDARY
2512	020012	010046			MOV R0,-(SP)	;SAVE HIGH ORDER
2513	020014	010146			MOV R1,-(SP)	;SAVE LOW ORDER
2514	020016	004737	017316		JSR PC,SETMAP	;SETUP PAR6 MAPPING REGISTER
2515	020022	010004			MOV R0,R4	;COPY ADDRESS BIASED TO PAR6
2516	020024	012601			MOV (SP)+,R1	;RESTORE LOW ORDER IN NON PAR6 FORMAT
2517	020026	012600			MOV (SP)+,R0	;RESTORE HIGH ORDER IN NON PAR6 FORMAT
2518	020030	020314		30\$:	CMP R3,(R4)	;ABOVE 28K LOCATION EQUAL EXPD?
2519	020032	001411			BEQ 32\$	;BR IF YES
2520	020034	010037	002236		MOV R0,ERRHI	;SAVE HIGH ORDER IN ERROR
2521	020040	010137	002240		MOV R1,ERRLO	;SAVE LOW ORDER IN ERROR
2522	020044	010337	002232		MOV R3,EXPD	;SAVE EXPD FOR ERROR REPORT
2523	020050	011437	002234		MOV (R4),RCV	;SAVE RCV FOR ERROR REPORT
2524	020054	000421			BR 50\$	
2525	020056	062701	000002	32\$:	ADD #2,R1	;UPDATE NON PAR6 ADDRESS
2526	020062	005500			ADC R0	;MAKE IT DOUBLE PRECISION ADD
2527	020064	062704	000002		ADD #2,R4	;UPDATE PAR6 FORMAT ADDRESS
2528	020070	020427	150000		CMP R4,#160000	;END OF PAR6 MAPPING AREA?
2529	020074	103755			BLO 30\$	;BR IF NO
2530	020076	162704	020000		SUB #20000,R4	;BACKUP INTO PAR6 MAPPING BEGIN

CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN

```

2531 020102 062737 000200 172354      ADD    #200,#KIPAR6      ;POINT TO NEXT 4K BLOCK >28K.
2532 020110 023737 172354 003132      CMP     @KIPAR6,KTFLG    ;END OF MEMORY?
2533 020116 101744                BLOS    30$              ;BR IF NO
2534 020120 004737 017274      50$:    JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
2535 020124 000241                CLC                     ;SET FAILURE
2536 020126 000403                BR      60$              ;
2537 020130 004737 017274      55$:    JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
2538 020134 000261                SEC                     ;SET SUCCESS
2539 020136 000207      60$:    RTS      PC
2540                                .SBTTL  REGSAV - SAVE R1-R5 ON STACK
2541                                ;+
2542                                ;
2543                                ;ROUTINE TO
2544                                ;SAVE R1 THROUGH R5 ON THE STACK
2545                                ;
2546                                ;CALLING SEQUENCE:
2547                                ;
2548                                ;      JSR      R5,REGSAV
2549                                ;
2550                                ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
2551                                ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
2552                                ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
2553                                ;REGISTERS.
2554                                ;
2555                                ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
2556                                ;CALLED VIA A JSR PC INSTRUCTION
2557                                ;
2558                                ;-
2559
2560 020140      REGSAV:
2561 020140 010446      MOV     R4,-(SP)
2562 020142 010346      MOV     R3,-(SP)
2563 020144 010246      MOV     R2,-(SP)
2564 020146 010146      MOV     R1,-(SP)
2565 020150 010546      MOV     R5,-(SP)
2566 020152 016605 00001?      MOV     10.(SP),R5
2567 020156 004736      JSR     PC,@(SP)+
2568 020160 012601      MOV     (SP)+,R1
2569 020162 012602      MOV     (SP)+,R2
2570 020164 012603      MOV     (SP)+,R3
2571 020166 012604      MOV     (SP)+,R4
2572 020170 012605      MOV     (SP)+,R5
2573 020172 000207      RTS      PC
2574                                .SBTTL  GETPAT - GET 8 BIT PATTERN FROM OPERATOR
2575                                ;+
2576                                ;
2577                                ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
2578                                ;
2579                                ;INPUTS:
2580                                ;
2581                                ;      NONE.
2582                                ;
2583                                ;OUTPUTS:
2584                                ;
2585                                ;      R0      OCTAL NUMBER FROM THE OPERATOR
2586                                ;
2587                                ;CALLING SEQUENCE:

```

B7

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 16-31

SEQ 0079

GETPAT - GET 8 BIT PATTERN FROM OPERATOR

```

2588
2589
2590
2591
2592
2593 020174
2594 020174
2595 020200
      020200 104443
      020202 000406
      020204 020230
      020206 000022
      020210 020232
      020212 000377
      020214 000000
      020216 000377
      020220
2596 020220
      020220 103367
2597 020222 013700 020230
2598 020226 000207
2599
2600
2601
2602
2603
2604 020230 000000
2605 020232 105 115 124
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624 020256
2625 020256
2626 020262 010002
2627 020264 010203
2628 020266 005713
2629 020270 001412
2630 020272
      020272 012346
      020274 012746 020442
      020300 012746 000002
      020304 010600

      JSR      PC,GETPAT

GETPAT::
1::  SAVREG          ;SAVE THE GENERAL REGISTERS
      GMANID DATASC,PATDAT,0,377,0,377,NO
      TRAP      CIGMAN
      BR        100001
      .WORD     PATDAT
      .WORD     TCODE
      .WORD     DATASC
      .WORD     377
      .WORD     TLOLIM
      .WORD     THILIM
100001:
      BNCOMPLETE 11      ;RETRY IF ERROR
      BCC      11
      MOV      PATDAT,R0      ;DATA PATTERN FROM OPERATOR
      RTS      PC            ;RETURN TO CALLER

      ;
      ;LOCAL DATA AREA
      ;
PATDAT: .WORD 0 ;TEMPORARY STORAGE FOR DATA
DATASC: .ASCIZ 'ENTER DATA PATTERN'
      .EVEN
      .SBTTL GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE
      ;
      ;ROUTINE TO ISSUE A MENU AND GET
      ;THE OPERATOR'S RESPONSE.
      ;
      ;INPUTS:
      ;
      ;      R0      ADDRESS OF ASCIZ STRING OF MENU
      ;      R1      MAXIMUM ALLOWABLE OPERATOR RESPONSE
      ;
      ;OUTPUTS:
      ;
      ;      R0      NUMBER OF THE OPERATOR'S SELECTION
      ;
      ;
GETSEL::
      SAVREG          ;SAVE GENERAL REGISTERS
      MOV      R0,R2    ;SAVE THE MENU ADDRESS
      MOV      R2,R3    ;START OF MENU STRING
2::  TST      (R3)      ;END OF ASCII ?
      BEQ      31      ;BRANCH IF ALL LINES DISPLAYED
      PRINTF   @SELASC,(R3),
      MOV      (R3),-(SP) ;DISPLAY THE MENU
      MOV      @SELASC, -(SP)
      MOV      @2, -(SP)
      MOV      SP,R0

```

C7

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 16-32

SEQ 0080

GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE

020306	104417			TRAP	C#PNTF	
020310	062706	000006		ADD	#6,SP	
2631 020314	000764			BR	2#	
2632 020316			3#:	G#MANID	MENASC,MENRES,D,-1,0,-1,NO	
020316	104443			TRAP	C#G#MAN	
020320	000406			BR	10001#	
020322	020476			.WORD	MENRES	
020324	000042			.WORD	T#CODE	
020326	020447			.WORD	MENASC	
020330	177777			.WORD	-1	
020332	000000			.WORD	T#LOLIM	
020334	177777			.WORD	T#HILIM	
020336			10001#:	BNCOMPLETE	1#	;RETRY IF ERROR
2633 020336				BCC	1#	
020336	103352			MOV	MENRES,R0	;GET THE OPERATOR'S REPLY
2634 020340	013700	020476		CMP	R0,R1	;COMPARE TO MAXIMUM ALLOWED
2635 020344	020001			BLOS	5#	;BRANCH IF OK
2636 020346	101411			PRINTF	#MENERR	;DISPLAY ERROR MESSAGE
2637 020350				MOV	#MENERR,-(SP)	
020350	012746	020374		MOV	#1,-(SP)	
020354	012746	000001		MOV	SP,R0	
020360	010600			TRAP	C#PNTF	
020362	104417			ADD	#4,SP	
020364	062706	000004		BR	1#	;RETRY
2638 020370	000735			RTS	PC	;RETURN TO CALLER
2639 020372	000207		5#:	MENERR: .ASCIZ	'###A *** Menu Selection Too Large ***'	
2640 020374	045	116	045	SELASC: .ASCIZ	'###T'	
2641 020442	045	116	045	MENASC: .ASCIZ	'Enter Menu Selection: '	
2642 020447	105	156	164	.EVEN		
2643				MENRES: .WORD	0	
2644 020476	000000			.SBTTL	CHKMAN - CHECK MANUAL INTERVENTION LEGALITY	
2645						
2646						
2647						
2648						
2649						
2650						
2651						
2652						
2653						
2654						
2655						
2656						
2657						
2658						
2659						
2660						
2661						
2662						
2663						
2664						
2665						
2666						
2667 020500						
2668 020500						
2669 020504						
020504	104450					

D7

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 16-33

SEQ 0081

CHKMAN - CHECK MANUAL INTERVENTION LEGALITY

```

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

```

E7

SEQ 0082

ENVIRN - SETUP FREE DIAGNOSTIC SPACE

```

2715 021032          PRINTF 0M8189          ;TELL THE SYSTEM TYPE
      021032 012746 005645      MOV 0M8189,-(SP)
      021036 012746 000001      MOV 01,-(SP)
      021042 010600          MOV SP,R0
      021044 104417          TRAP C$PNTF
      021046 062706 000004      ADD 04,SP
2716 021052 000207      40$: RTS PC          ;RETURN
2717          .SBTTL KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS
2718          ;
2719          ;
2720          ;ROUTINE TO INIT KT-11
2721          ;
2722          ;
2723          ;
2724 021054          KTINIT:
2725 021054 005037 003132      CLR KTFLG          ; INIT >28K MEMORY FLAG
2726 021060 005037 003134      CLR KTENABLE       ; INIT TEST >28K FLAG
2727 021064 023727 002120 001577  CMP L$HIME,01577 ; GOT ENOUGH MEMORY (>28K)?
2728 021072 101444          BLVS 9$              ; NO.
2729 021074 013700 000004      MOV 0$ERRVEC,R0    ; SAVE OLD ERR VEC PTR.
2730 021100 012737 021172 000004  MOV 02$,0$ERRVEC ; SET ERR VEC PTR.
2731 021106 005737 177572      TST 0$SRO         ; GOT KT11?
2732 021112 000240          NOP                  ; (TRAP IF NO).
2733 021114 013737 002120 003132  MOV L$HIME,KTFLG ; YES. SET KT FLAG.
2734 021122 042737 000177 003132  BIC 0177,KTFLG
2735 021130 010037 000004      MOV R0,0$ERRVEC   ; RESTORE OLD ERR VEC PTR.
2736 021134 005000          CLR R0              ; R0 = AR DATA.
2737 021136 012701 172340      MOV 0$KIPAR0,R1   ; R1 = KI REGS PTR.
2738 021142 012761 077406 177740 1$: MOV 077406,-40(R1) ; SET DESCRIPTOR REG.
2739 021150 010021          MOV R0,(R1)+        ; SET KIPAR REG.
2740 021152 062700 000200      ADD 0200,R0       ; BUMP AR DATA BY "4K".
2741 021156 020027 002000      CMP R0,02000     ; AT "I/O"?
2742 021162 001367          BNE 1$              ; NO.
2743 021164 012741 177600      MOV 0177600,-(R1) ; YES. SET KIPAR7 FOR I/O.
2744 021170 000405          BR 9$
2745          ;
2746 021172 012716 021200      2$: MOV 06$, (SP)   ; SET UP RETURN
2747 021176 000002          RTI                  ; RTI TO NEXT LOCATION
2748          ;
2749 021200 010037 000004      6$: MOV R0,0$ERRVEC ; RESTORE OLD ERR VEC PTR.
2750          ;
2751 021204 000207      9$: RTS PC
2752          ;
2753          ; SUBROUTINE TO SET EXTENDED FEATURES SWITCH
2754          ;
2755          ; Requires that SOFINIT and WRCHRR have been done previous to call.
2756          ;
2757          ;
2758          ; INPUTS:
2759          ; R5 CURRENT UNIT NUMBER
2760          ; OUTPUTS:
2761          ; The Extended Features Switch is set.
2762          ;
2763          ;
2764          ;
2765 021206          INVERT:
2766          ;

```

F7

TSV3 - GLOBAL AREAS MACRO M1200 20-MAR-84 08:26 PAGE 16-35

SEQ 0083

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

2767 021206 005737 002226      TST      EXTFEA      ; IS SWITCH SET?
2768 021212 001020      BNE      1$      ; YES,EXIT STAGE RIGHT!(or the next one outa town!)
2769 021214 012737 100206 021260      MOV      #100206,CMDPKT      ; WRT SUB-SYS MEM CMD
2770 021222 012737 021270 021262      MOV      #WSMBK,CMDPKT+2      ; MSG BUF ADDR
2771 021230 012737 000006 021266      MOV      #6,CMDPKT+6      ; BYTE COUNT
2772 021236 012737 100010 021270      MOV      #100010,WSMBK      ; INVERT THE SWITCH
2773 021244 012704 021260      MOV      #CMDPKT,R4      ; SET CMDPKT INTO R4
2774 021250 004737 010662      JSR      PC,WRTCHR      ; DO IT
2775 021254 000207      1$:      RTS      PC      ; RETURN
2776
2777      ;      COMMAND PACKET.
2778
2779      021260      .      -      <,+3>&177774      ;MUST BE ON MOD 4 BOUNDARY.
2780
2781 021260 000000      CMDPKT:: 0      ;1ST WORD IS TS05 COMMAND.
2782 021262 000000      0      ;2ND WORD IS THE BUFFER LOW ADDRESS.
2783 021264 000000      0      ;3RD WORD IS THE BUFFER HIGH ADDRESS.
2784 021266 000000      0      ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
2785
2786      ;      WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
2787
2788 021270 000000      WSMBK:: 0      ;1ST WORD:: SEL 0
2789 021272 000000      0      ;2ND WORD:: SEL 2
2790 021274 000000      0      ;3RD WORD:: SEL 4
2791      .EVEN
2792
2793      ;+      SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
2794      ;
2795      ;
2796      ;INPUTS:
2797      ;OUTPUTS:
2798      ;      The NXMFLG is set if we can test.
2799      ;      The NXMLO and NXMHI addresses are setup.
2800      ;-
2801
2802 021276      MEMCK::
2803
2804 021276      SAVREG      ;SAVE THE REGISTERS
2805 021302 005037 003136      CLR      NXMFLG      ;CLEAR THE FLAG
2806 021306 005037 003140      CLR      NXMLO      ;CLEAR THE TEST ADDRESS LO
2807 021312 005037 003142      CLR      NXMHI      ;CLEAR THE TEST ADDRESS HI
2808 021316 005737 003146      TST      T23B      ;IS IT A 11/23B?
2809 021322 001407      BEQ      1$      ;NO
2810 021324 023727 002120 007777      CMP      L#HIME,#7777      ; GREATER THAN 128K
2811 021332 103406      BLO      2$      ; NO
2812 021334 004737 021452      JSR      PC,NXMTST      ;SETUP THE ADDRESS
2813 021340 000427      BR      13$      ;SET THE FLAG AND EXIT
2814 021342 005737 003144      1$:      TST      T23A      ;IS IT A 11/23A?
2815 021346 001413      BEQ      4$      ;NO
2816 021350 023727 002120 005777 2$:      CMP      L#HIME,#5777      ;GREATER THAN 96K
2817 021356 101023      BHI      14$      ;YES,23A/23B WITH 128K MEMORY
2818 021360 023727 002120 003777      CMP      L#HIME,#3777      ;GREATER THAN 64K BUT LESS THAN 92K?
2819 021366 103403      BLO      4$      ;NO, CHECK 24K
2820 021370 004737 021452      JSR      PC,NXMTST      ;SETUP THE ADDRESS
2821 021374 000411      BR      13$      ;SET THE FLAG AND EXIT
2822 021376 023727 002120 001577 4$:      CMP      L#HIME,#1577      ;GREATER THAN 24K BUT LESS THAN 64K?
2823 021404 103410      BLO      14$      ;NO, TELL THEM AND EXIT WITH FLAG CLEAR

```

G7

SEQ 0084

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

2824 021406 004737 021452      JSR      PC,NXMTST      ;SETUP THE ADDRESS
2825 021412 062737 000077 003142  ADD      #77,NXMHI      ;FOOL THE 11/02 & 11/03
2826 021420 005237 003136      13$: INC      NXMFLG      ;SET THE FLAG
2827 021424 000411              BR       15$          ;EXIT
2828 021426 000410              14$: BR       15$          ;NOP FOR PRINTOUT
2829 021430              PRINTF  #NOMEM      ;TELL THEM & EXIT ***NO PRINT*****
      021430 012746 005460      MOV      #NOMEM,-(SP)
      021434 012746 000001      MOV      #1,-(SP)
      021440 010600              MOV      SP,R0
      021442 104417              TRAP     C#PNTF
      021444 062706 000004      ADD      #4,SP
2830 021450 000207      15$:  RTS      PC              ;RETURN
2831
2832      ;*
2833      ;      SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
2834      ;
2835      ;OUTPUTS: NXML0,NXMHI      ;SETUP WITH NXM ADDRESS
2836      ;
2837      ;-
2838
2839 021452 013701 002120      NXMTST: MOV      L#HIME,R1      ;GET TOP OF MEMORY
2840 021456 062701 000200      ADD      #200,R1      ;MAKE IT I/O BLOCK OR OTHER NXM
2841 021462 042701 000177      BIC      #177,R1
2842 021466 010102              MOV      R1,R2      ;RESAVE RESULTS
2843              000006      .REPT      6
2844              ASL      R1      ;PUT IN PLACE FOR XFER
2845              .ENDR
2846 021504 010137 003140      MOV      R1,NXML0      ;SAVE TEST ADDRESS LOW
2847              000012      .REPT      10
2848              ASR      R2      ;PUT IN PLACE FOR XFER
2849              .ENDR
2850 021534 042702 177700      BIC      #177700,R2      ;DON'T WANT ILA!
2851 021540 010237 003142      MOV      R2,NXMHI      ;SAVE TEST ADDRESS HIGH
2852 021544 000207      RTS      PC              ;RETURN
2853
2854
2855
2856 021546      ENDMOD

```

H7

TSV4 - MISCELLANEOUS SECTIONS MACRO M1200 20-MAR-84 08:26 PAGE 17

SEQ 0085

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

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

PROTECTION TABLE

18							.SBTTL	PROTECTION TABLE
19	021546						BGNPROT	
	021546						L\$PROT::	
20	021546	177777	177777	177777			.WORD	-1, -1, -1, -1
21	021556						ENDPROT	
22								

NO DEVICE PROTECTION REQUIRED.

J7

SEQ 0087

INITIALIZE SECTION

```

24          .SBTTL  INITIALIZE SECTION
25
26          ;++
27          ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
28          ;AT THE BEGINNING OF EACH PASS.
29          ;
30          ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
31          ;IF "CONTINUE", NOTHING IS REQUIRED.
32          ;
33          ;--
34          ;+
35          ;INSERT TEMPORARY JUMP TO ODT
36          ;-
37 021556      BGNINIT
38 021556      L$INIT::
39 021556      SETVEC  #140,#170000,#340      ;ODT ROM ADDRESS      ;JB REV A-C
021556 012746 000340      MOV  #340,-(SP)
021562 012746 170000      MOV  #170000,-(SP)
021566 012746 000140      MOV  #140,-(SP)
021572 012746 000003      MOV  #3,-(SP)
021576 104437      TRAP  C$SVEC
021600 062706 000010      ADD  #10,SP
40
41 021604 005037 002226      40$: CLR  EXTFEA
42 021610 005037 003136      CLR  NXMFLG
43 021614 012737 006360      MOV  #EPRT1,EPRTSW      ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
44 021622 005037 003154      CLR  SIFLAG      ;CLEAR "SOFT INIT" FLAG
45 021626 005037 003134      CLR  KTENABLE      ;CLEAR TEST ABOVE 28K FLAG
46 021632 005037 002302      CLR  RAMSIZ      ;CLEAR RAM SIZE FOR RAMERR ROUTINE
47 021636      READEF  #EF.CONTINUE
021636 012700 000036      MOV  #EF.CONTINUE,RO
021642 104447      TRAP  C$REFG
48 021644      BNCOMPLETE 1$
021644 103023      BCC  1$
49 021646 023737 002202 002012      CMP  UNITN,L$UNIT      ;UNIT IN RANGE?
50 021654 103070      BHIS  4$      ;BR IF NO.
51 021656 005737 003112      TST  DUFLG      ;DROPPED UNIT?
52 021662 100472      BMI  NXTU      ;BR IF YES
53 021664 013701 002202      MOV  UNITN,R1
54 021670 006301      ASL  R1
55 021672 005761 003176      TST  ERTABL(R1)
56 021676 001516      BEQ  SETU
57 021700 032761 040000 003176      BIT  #BIT14,ERTABL(R1)      ;DROPPED?
58 021706 001060      BNE  NXTU
59 021710      EXIT  INIT      ;DO NOTHING IF "CONTINUE".
021710 104432      TRAP  C$EXIT
021712 000416      .WORD  L10030
60 021714      READEF  #EF.NEW
021714 012700 000035      MOV  #EF.NEW,RO
021720 104447      TRAP  C$REFG
61 021722      BNCOMPLETE NXTU      ;TAKE NEXT UNIT IF NOT NEW PASS.
021722 103052      BCC  NXTU
62 021724      READEF  #EF.START
021724 012700 000040      MOV  #EF.START,RO
021730 104447      TRAP  C$REFG
63 021732      BCOMPLETE 2$

```

	021732	103404			BCS	2\$		
64	021734				READEP	DEF.RESTART		
	021734	012700	000037		MOV	DEF.RESTART,RO		
	021740	104447			TRAP	C\$REFG		
65	021742				BNCOMPLETE	31\$		
	021742	103031			BCC	31\$		
66	021744							;1ST PASS, BUS-INIT...
67	021744				BRESET			;BUS RESET.
	021744	104433			TRAP	C\$RESET		
68	021746	005037	002214		CLR	TSTCNT		;NUMBER OF TESTS RUN IN PASS
69	021752	005037	002222		CLR	FATFLG		;CLEAR FATAL ERROR COUNT
70	021756	005037	003144		CLR	T23A		;CLEAR 11/23A FLAG
71	021762	005037	003146		CLR	T23B		;CLEAR 11/23B FLAG
72					MOV	#340,-(SP)		
73					MOV	#201,-(SP)		;RETURN TO DEBUGGER
74					JMP	O.OOI		;ENTER THE DEBUGGER
75	021766	005037	003400		CLR	SKIPT		;CLEAR THE SUBTEST "SKIPPER"
76	021772							
77	021772	012737	177777	002204	MOV	#-1,QVP		;...QUICK VERIFY...
78	022000	004737	020630		JSR	PC,ENVIRN		;SET ENVIRONMENT.
79	022004	004737	021054		JSR	PC,KTINIT		;INITIALIZE KT MEMORY MANAGEMENT
80	022010	012700	003176		MOV	\$ERTABL,RO		
81	022014	005020			CLR	(RO),		;CLEAR THE ERROR TABLE
82	022016	020027	003376		CMP	RO,\$ERTABE		
83	022022	103774			BLO	30\$		
84	022024	000404			BR	4\$		
85	022026	005037	002204		CLR	QVP		
86	022032	000137	022102		JMP	PASRPT		;GO REPORT THE STATUS
87								
88	022036				4\$:			
89	022036	012737	177777	002202	NEWPAS:	MOV	#-1,UNITN	;INIT UNIT NUMBER...
90	022044	005037	002220			CLR	DEVcnt	;CLEAR COUNT OF DEVICES RUNNING
91	022050				NXTU;	BREAK		
	022050	104422			TRAP	C\$BRK		
92	022052	005237	002202		INC	UNITN		;...AND SET NEXT UNIT NUMBER.
93	022056	023737	002202	002012	CMP	UNITN,L\$UNIT		
94	022064	103423			BLO	SETU		
95	022066	012737	177777	003112	MOV	#-1,DUFFLG		
96	022074	000401			BR	11\$		
97	022076				DOCLN			;ABORT, NO MORE UNITS.
	022076	104444			TRAP	C\$DCLN		
98	022100	000240				NOP		
99	022102				11\$: PASRPT:			
100	022102	023727	002012	000001	CMP	L\$UNIT,#1		;HOW MANY UNITS SELECTED?
101	022110	101752			BLOS	NEWPAS		;BR IF ONLY 1
102	022112	005737	002220		TST	DEVcnt		;ARE ANY STILL RUNNING?
103	022116	001747			BEQ	NEWPAS		;BR IF NO
104	022120				RFLAGS	RO		
	022120	104421			TRAP	C\$RFLA		
105	022122	032700	000100		BIT	\$ISR,RO		;SHOULD WE PRINT STATISTICS
106	022126	001343			BNE	NEWPAS		;BR IF NO
107								
108	022130				DORPT			
	022130	104424			TRAP	C\$DRPT		
109	022132	000741			BR	NEWPAS		
110	022134				10\$:			
111								

INITIALIZE SECTION

```

112 022134      SETU:  GPHARD  UNITN,R0      ;GET UNIT N P-TABLE POINTER.
      022134 013700 002202      MOV      UNITN,R0
      022140 104442      TRAP      C#GPHRD
113 022142      BNCOMPLETE NXTU      ;BR IF UNIT NOT AVAILABLE.
      022142 103342      BCC      NXTU
114 022144 005037 003112      CLR      DUFLG      ;CLEAR "DROPPED" FLAG.
115 022150 005237 002220      INC      DEVCNT
116 022154 012001      MOV      (R0)+,R1      ;GET 1ST REGISTER ADDRESS.
117 022156 010137 002206      MOV      R1,CSRADDR      ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
118
119 022162 012001      MOV      (R0)+,R1      ;GET VECTOR ADDRESS.
120      ;MOV      (R0),R2      ;GET INTERRUPT PRIORITY
121      ;MOV      R2,IPRI      ;SET INTERRUPT PRIORITY.
122 022164 010137 002210      MOV      R1,IVEC      ;SET INTERRUPT VECTOR POINTER...
123 022170 012721 016216      MOV      @INTR,(R1)+      ;...VECTOR...
124 022174 013721 002212      MOV      IPRI,(R1)+      ;...AND PRIORITY.
125
126 022200      1$:
127      ;      TST      QVP      ;1ST PASS ??
128      ;      BEQ      5$      ;NO, SKIP THE PASS 1 STUFF.
129
130
131      ;
132      ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
133      ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
134 022200 013701 002202      ;
135 022204 006301      MOV      UNITN,R1
136 022206 052761 100000 003176      ASL      R1
137 022214 005037 005772      BIS      @BIT15,ERTABL(R1)      ;SAY DEVICE RUNNING
138 022220 023727 002012 000001      CLR      EXTA      ;CLEAR ERROR EXTENSION FLAG.
139 022226 101416      CMP      L#UNIT,@1      ;ARE WE TESTING MULTIPLE UNITS?
140 022230      BLOS      10$      ;BR IF NO.
141 022232 032700 001000      RFLAGS      R0      ;YES -- GET OPERATOR FLAGS.
142 022236 001412      TRAP      C#RFLA
143 022240      BIT      @PNT,R0      ;SHOULD WE PRINT UNIT #?
144 022240 013746 002202      BEQ      10$      ;BR IF NOT.
145 022244 012746 022332      PRINTF      @PUNIT,UNITN      ;PRINT THE UNIT #
146 022250 012746 000002      MOV      UNITN,-(SP)
147 022254 010600      MCV      @PUNIT,-(SP)
148 022256 104417      MOV      @2,-(SP)
149 022260 062706 000006      MOV      SP,R0
150      TRAP      C#PNTF
151      ADD      @6,SP
152
153      10$:
154      CLR      NODEV
155      MOV      CSRADDR,R1      ;ADDRESS OF FIRST REGISTER
156      MOV      R1,R2      ;START OF REGISTERS
157      ADD      @1SSR,R2      ;ADDRESS OF TSSR REGISTER
158      JSR      PC,XNXM      ;TEST BOTH CONTROLLER REGISTERS...
159      BCC      2$      ;...AND BR IF ALL OK.
160      MOV      R1,NODEV      ;FLAG DEVICE AS NON-EXISTENT
161      MOV      @-1,DUFLG      ;DROP THIS UNIT.
162
163      2$:
164      ;
165      ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
166      ;
167      5$:      SETPRI      @PRI00      ;ENABLE INTERRUPTS.
168      MOV      @PRI00,R0

```

M7

TSV4 - MISCELLANEOUS SECTIONS MACRO M1200 20-MAR-84 08:26 PAGE 19-3

SEQ 0090

INITIALIZE SECTION

```

    022326 104441          TRAP  C$SPRI
158 022330          ENDINIT
    022330          L10030:
    022330 104411          TRAP  C$INIT
159
160 022332      045      116      045 PUNIT: .ASCIZ /NNNNAAAAA TESTING UNIT *D2*A *****/
161      .EVEN
```

ADD AND DROP UNITS SECTIONS

```

163                                     .SBTTL  ADD AND DROP UNITS SECTIONS
164
165
166                                     ;**
167                                     ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
168                                     ; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
169                                     ; OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
170                                     ;--
171 022400                                BGNAU
172 022400                                L$AU::
173 022400 010001                                MOV     R0,R1                ; GET UNIT TO BE ADDED (R0)
174 022402 006301                                ASL      R1                    ; MAKE IT A WORD INDEX
175 022404 052761 100000 003176                BIS      #100000,ERTABL(R1)      ; SET THE "ACTIVE" BIT
176 022412 042761 040000 003176                BIC      #40000,ERTABL(R1)      ; CLEAR THE "DROPPED" BIT
177 022420                                PRINTF  #1$,R0
178 022420 010046                                MOV      R0,-(SP)
179 022422 012746 022446                                MOV      #1$,-(SP)
180 022426 012746 000002                                MOV      #2$,-(SP)
181 022432 010600                                MOV      SP,R0
182 022434 104417                                TRAP     C$PNTF
183 022436 062706 000006                                ADD      #6,SP
184 022442                                EXIT     AU
185 022442 000167                                .WORD    J$JMP
186 022444 000026                                .WORD    L10031-2-
187 022446 045 116 045 1$: .ASCIZ  /%N%A UNIT %D%A ADDED/
188                                     .EVEN
189
190                                     ENDAU                                ; UNUSED.
191
192 022474                                L10031:
193 022474 104452                                TRAP     C$AU
194
195                                     ;**
196                                     ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
197                                     ; TO BE REMOVED FROM THE TEST LIST.
198                                     ;
199                                     ; SUPVSK DOES THE "DROPPING". THIS IS JUST TO TELL THE MAN,
200                                     ; "DROPPED" UNITS ARE RE-SELECTED ON OPERATOR "STA" OR "ADD"
201                                     ; COMMAND, OTHERWISE REMAIN INACTIVE. THE "DISPLAY" COMMAND
202                                     ; WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
203                                     ; WHICH ARE STILL ACTIVE.
204                                     ; UPON ENTRY, R0 CONTAINS THE UNIT TO BE DROPPED.
205
206                                     BGNDU
207 022476                                L$DU::
208 022476 012737 177777 003112                MOV      #1,DUFLG
209 022504 010001                                MOV      R0,R1
210 022506 006301                                ASL      R1
211 022510 052761 140000 003176                BIS      #140000,ERTABL(R1)      ; SAY DROPPED
212 022516 000240 000240 000240                240,240,240                    ; ??????????
213 022524                                PRINTF  #1$,R0
214 022524 010046                                MOV      R0,-(SP)
215 022526 012746 022552                                MOV      #1$,-(SP)
216 022532 012746 000002                                MOV      #2$,-(SP)
217 022536 010600                                MOV      SP,R0
218 022540 104417                                TRAP     C$PNTF
219 022542 062706 000006                                ADD      #6,SP
220 022546                                EXIT     DU
221 022546 000167                                .WORD    J$JMP
222 022550 000030                                .WORD    L10032-2-

```

B8

TSV4 - MISCELLANEOUS SECTIONS MACRO M1200 20-MAR-84 08:26 PAGE 20-1

SEQ 0092

ADD AND DROP UNITS SECTIONS

200	022552	045	116	045	11:	.ASCIZ /WNA UNIT WDA DROPPED/	
201						.EVEN	
202	022602					ENDDU	
	022602			L10032:			
	022602	104453				TRAP C1DU	
203						---	
204						! AUTO-DROP CODE SECTION.	
205						---	
206	022604					BGNAUTO	
	022604			L1AUTO:			
207	022604	013705	002206			MOV CSRADDR,R5	!POINT TO DEVICE REGISTER
208	022610	012703	000550			MOV #360,R3	!ENOUGH TIME FOR 2400' REEL TO REWIND
209	022614	004737	016250	101:		JSR PC,WAITF	!WAIT FOR SSR TO SET
210	022620	103420				BCS 201	!LEAVE WHEN SSR IS SET
211	022622					DELAY 250.	!WAIT FOR .25 SECONDS
	022622	012727	000372			MOV #250.,(PC),	
	022626	000000				.WORD 0	
	022630	013727	002116			MOV L1DL Y,(PC),	
	022634	000000				.WORD 0	
	022636	005367	177772			DEC -6(PC)	
	022642	001375				BNE .-4	
	022644	005367	177756			DEC -22(PC)	
	022650	001367				BNE .-20	
212	022652	005303				DEC R3	!BUMP COUNTER DOWN
213	022654	001357				BNE 101	!KEEP GOING
214	022656	004737	017202			JSR PC,CKDROP	!TRY AND DROP UNIT
215	022662			201:			
216	022662					ENDAUTO	! UNUSED.
	022662			L10033:			
	022662	104461				TRAP C1AUTO	

D8

TSV4 - MISCELLANEOUS SECTIONS MACRO M1200 20-MAR-84 08:26 PAGE 21-1

SEQ 0094

CLEAN-UP AND REPORT CODING SECTIONS

```

023030 012746 000002      MOV     #2,-(SP)
023034 010600      MOV     SP,R0
023036 104416      TRAP    C1PNTS
023040 062706 000006      ADD     #6,SP
257 023044 000431      BR      41
258 023046 020227 160001      31:    CMP     R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
259 023052 001012      BNE     301      ; BR IF NO.
260 023054      PRINTS   #DEVNRD,R3
023054 010346      MOV     R3,-(SP)
023056 012746 023343      MOV     #DEVNRD,-(SP)
023062 012746 000002      MOV     #2,-(SP)
023066 010600      MOV     SP,R0
023070 104416      TRAP    C1PNTS
023072 062706 000006      ADD     #6,SP
261 023076 000414      BR      41
262 023100 042702 170000      301:   BIC     #1C7777,R2
263 023104      PRINTS   #DEVDRD,R3,R2
023104 010246      MOV     R2,-(SP)
023106 010346      MOV     R3,-(SP)
023110 012746 023424      MOV     #DEVDRD,-(SP)
023114 012746 000003      MOV     #3,-(SP)
023120 010600      MOV     SP,R0
023122 104416      TRAP    C1PNTS
023124 062706 000010      ADD     #10,SP
264 023130 062704 000002      41:    ADD     #2,R4
265 023134 005203      INC     R3
266 023136 020427 003376      CMP     R4,#ERTABE
267 023142 103701      BLO     11
268 023144 012604      MOV     (SP)+,R4
269 023146 012603      MOV     (SP)+,R3
270 023150 012602      MOV     (SP)+,R2
271 023152      ENDRPT      ; UNUSED.
023152      L10035:
023152 104425      TRAP    C1RPT
272
273
274 023154      045      116      045  DEVSUM: .ASCIZ  /#N#ADEVICE STATUS SUMMARY:#N/
275 023211      045      101      040  DEVONL: .ASCIZ  /#A UNIT #D3#A  ONLINE,  ERRORS = #D#N/
276 023261      045      101      040  DEVNXR: .ASCIZ  /#A UNIT #D3#A  DROPPED, NON-EXISTENT REGISTER#N/
277 023343      045      101      040  DEVNRD: .ASCIZ  /#A UNIT #D3#A  DROPPED, NOT READY AT STARTUP#N/
278 023424      045      101      040  DEVDRD: .ASCIZ  /#A UNIT #D3#A  DROPPED, ERRORS = #D#N/
279      .EVEN
280
281 023474      ENDMOD
282
283

```

E8

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 23

SEQ 0095

CLEAN-UP AND REPORT CODING SECTIONS

1
2
9
10
16
24

023474
023474

.TITLE TSV5 - HARDWARE TESTS

BGNMOD TSV5
TSV5::

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

```

26          .SBTTL TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS
27
28          ;*
29          ; TEST DESCRIPTION:
30          ;
31          ; This test verifies that a Hardware Initialize command
32          ; invoked after a Write Characteristics command sets up
33          ; the Command, Message and Characteristic image blocks
34          ; in the controller ram correctly.
35          ;
36          ; TEST STEPS:
37          ;
38          ; REPEAT FOR LOOPCNT
39          ; BEGIN
40          ; Do WRITE CHARACTERISTICS command.
41          ; If the NBA bit in the TSSR register is NOT=0 then Print Error.
42          ; Write to TSSR register to soft initialize the controller
43          ; If controller RAM 310-377 NOT=0 then Print Error
44          ; END
45          ;--
46
47          BGNTST
48          T1::
53          023474      012700      024132      MOV      #TST13ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
54          023500      004737      016510      JSR      PC,TSTSETUP      ;DO INITIAL TEST SETUP
55          023504      012737      000012      002216      MOV      #10,,LOOPCNT      ;PERFORM 10 ITERATIONS
56          023512      T13LOOP:
57          023512      004737      024406      JSR      PC,T13REST      ;SET PACKET TO START-UP VALUES
58
59          023516      012703      002764      MOV      #TSTBLK+10,,R3      ;START OF TEST DATA
60          023522      012704      024070      MOV      #T13PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
61          023526      012764      000010      000006      MOV      #8,,PKBCNT(R4)      ;START WITH MINIMUM ALLOWABLE VALUE
62          023534      5$:
63          023534      004737      015774      JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
64          023540      103405      BCS      10$      ;BR IF SOFT INIT OKAY
65          023542      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
66          023544      ERRDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
67          023544      104455      TRAP      C$ERDF
68          023546      000144      .WORD      100
69          023550      003652      .WORD      SFIERR
70          023552      012034      .WORD      SFIMSG
71
72          ;Do WRITE CHARACTERISTICS command.
73          10$:
74          023554      005037      002222      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
75          023560      010465      000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
76          023564      004737      016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
77          023570      FORCERROR      12$      ;GOODFORCE ERROR IF FORCER=1
78          023604      103407      BCS      15$      ;BR IF CARRY SET (GOOD RETURN)
79          023606      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
80          023610      NEXT.ERRNO
81          023610      104455      12$:      ERRDF      ERRNO,T13SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
82          023612      000145      TRAP      C$ERDF
83          023614      024317      .WORD      101
84          023616      012046      .WORD      T13SSR
85          023620      005237      002222      INC      FATFLG      ;SET FATAL ERROR FLAG
86          023624      15$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET

```

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

```

      023624 104406
79 023626 016501 000002      MOV    TSSR(R5),R1      ;GET THE CONTENTS OF TSSR      TRAP    C$CLP1
80 023632 012702 000200      MOV    $SSR,R2      ;EXPECTED CONTENTS OF TSSR
81 023636 032701 000100      BIT    $OFL,R1      ;IS OFF-LINE BIT SET ?
82 023642 001402      BEQ    25$      ;BRANCH IF NOT OFF-LINE
83 023644 052702 000100      BIS    $OFL,R2      ;SET OFF-LINE IN EXPECTED DATA
84
85      ;If the NBA bit in the TSSR register is NOT=0 then Print Error.
86 023650 25$:      FORCERROR      27$      ;880
87 023650      CMP    R2,R1      ;DOES EXPECTED MATCH RECEIVED ?
88 023664 020201      BEQ    30$      ;OKAY IF MATCH
89 023666 001404      NEXT,ERRNO
90 023670 27$:      ERRHRD  ERRNO,T13NBA,PKTSSR      ;NBA NOT ZERO
91 023670 104456      TRAP    C$ERHRD
      023672 000146      .WORD  102
      023674 024244      .WORD  T13NBA
      023676 012046      .WORD  PKTSSR
92 023700 30$:      CKLOOP      ;LOOP ON ERROR ?      TRAP    C$CLP1
      023700 104406
93
94      ;Write to TSSR register to soft initialize the controller
95 023702 40$:      JSR    PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
96 023702 004737 015774      FORCERROR      42$      ;880
97 023706      BCS    50$      ;BR IF SOFT INIT OKAY
98 023722 103405      MOV    R0,R1      ;SAVE CONTENTS OF TSSR
99 023724 010001      NEXT,ERRNO
100 023726 42$:      ERDF  ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
101 023726 104455      TRAP    C$ERDF
      023730 000147      .WORD  103
      023732 003652      .WORD  SFIERR
      023734 012034      .WORD  SFIMSG
102
103      ;If controller RAM 310-377 NOT=0 then Print Error
104 023736 012704 000310 50$:      MOV    $310,R4      ;START WITH LOC 310
105 023742 005002      CLR    R2      ;MEMORY EXPECTED SHOULD BE 000000
106 023744 105065 000000      CLRB   TSDB(R5)      ;SET MAINTENANCE MODE
107 023750 004737 016336      JSR    PC,CHKTSSR      ;WAIT FOR SSR READY
108 023754 010465 000000 60$:      MOV    R4,TSDB(R5)      ;SELECT RAM ADDRESS
109 023760 004737 016336      JSR    PC,CHKTSSR      ;WAIT FOR SSR READY
110 023764 116501 000000      MOVB   TSBA(R5),R1      ;READ LOC CONTENTS
111 023770      FORCERROR      62$,NOTSSR      ;880
112 024000 120102      CMPB   R1,R2      ;CHECK MEMORY FOR 000000
113 024002 001406      BEQ    70$      ;BRANCH IF DATA OKAY
114 024004      NEXT,ERRNO
115 024004 62$:      ERDF  ERRNO,T13MEM,RAMEXP      ;MEMORY NOT ZERO AFTER INIT.
      024004 104455      TRAP    C$ERDF
      024006 000150      .WORD  104
      024010 024205      .WORD  T13MEM
      024012 015530      .WORD  RAMEXP
116 024014 005237 002222 70$:      INC    FATFLG      ;SET THE FATAL ERROR FLAG
117 024020      CKLOOP
      024020 104406      TRAP    C$CLP1
118 024022      ESCAPE  TST      ;EXIT ON FATAL ERROR
      024022 104410      TRAP    C$ESCAPE
      024024 000430      .WORD  L10036-.

```

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

Line	Address	Hex	Hex	Label	Operation	Register	Comment
119	024026	005204			INC	R4	;LOOK AT NEXT RAM LOC.
121	024030	020427	000400		CMP	R4,#400	;AT TOP OF RAM ADDRESS SPACE
122	024034	001347			BNE	60H	;BRANCH TILL ALL MEMORY TESTED
123							
124							
125	024036	005737	002222		TST	FATFLG	;ANY FATAL ERRORS ?
126	024042	001402			BEQ	160H	;BRANCH IF NOT
127	024044	004737	017202		JSR	PC,CKDROP	;TRY TO DROP THE UNIT
128	024050	004737	016456		JSR	PC,TSTLOOP	;DONE ALL ITERATIONS?
129	024054	103002			BCC	165H	;BR IF YES
130	024056	000137	023512		JMP	T13LOOP	;LOOP UNTIL ITERATION COUNT DONE
131	024062						
132	024062				EXIT	TST	
	024062	104432					TRAP C#EXIT
	024064	000370					.WORD L10036-.
133							
134							
135							
136							
137							
138							
140		024070					
142	024070				T13PACKET:		;COMMAND PACKET FOR TEST
143	024070	100004			.WORD	100004	;WRITE CHARACTERISTICS COMMAND, WITH ACK
144	024072	024100			.WORD	T13DATA	;ADDRESS OF CHARACTERISTICS BLOCK
145	024074	000000			.WORD	0	
146	024076	000010			.WORD	8.	;STARTING VALUE OF BLOCK SIZE
147							
148	024100				T13DATA:		;CHARACTERISTICS DATA BLOCK
149	024100	024112			.WORD	T13BFR	;ADDRESS OF MESSAGE BUFFER
150	024102	000000			.WORD	0	
151	024104	000016			.WORD	14.	;LENGTH OF MESSAGE BUFFER
152	024106	000000	000000		.WORD	0,0	
153							
154	024112				T13BFR: .BLKW	8.	;MESSAGE BUFFER
155							
156							
157							
158	024132	111	156	151	TST13ID:	.ASCIZ	'Initialization After WRITE CHARACTERISTICS'
159	024205	111	156	143	T13MEM:	.ASCIZ	'Incorrect RAM Data After Init'
160					.EVEN		
161	024244	127	122	111	T13NBA:	.ASCIZ	'WRITE CHARACTERISTICS Command Not Accepted'
162	024317	103	157	156	T13SSR:	.ASCIZ	'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
163							
164							
165							
166							
167							
168							
169							
170							
171							
172							
173	024406				T13REST:		
174	024406				SAVREG		;SAVE THE REGISTERS
175	024412	012701	024070		MOV	#T13PACKET,R1	;START OF THE PACKET

Line	Address	Offset	Value	Label	Instruction	Comment
176	024416	012721	100004		MOV	#100004,(R1)+ ;WRITE CHARACTERISTICS WITH ACK
177	024422	012721	024100		MOV	#T13DATA,(R1)+ ;ADDRESS OF CHAR DATA BLOCK
178	024426	005021			CLR	(R1)+ ;EXTENDED ADDRESS
179	024430	012721	000010		MOV	#8,,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
180	024434	012721	024112		MOV	#T13BFR,(R1)+ ;ADDRESS OF MESSAGE BUFFER
181	024440	005021			CLR	(R1)+
182	024442	012721	000016		MOV	#14,,(R1)+ ;LENGTH OF MESSAGE BUFFER
183	024446	005021			CLR	(R1)+
184	024450	005011			CLR	(R1)
185	024452	000207			RTS	PC ;RETURN
186	024454				ENDTST	
	024454	104401				
187						
188						
189						
190						
191						
192						
193						
194						
195						
196						
197						
198	024456				BGNTST	
	024456					
203	024456	012700	026271		MOV	#T14ID,R0 ;ASCII MESSAGE TO IDENTIFY TEST
204	024462	004737	016510		JSR	PC,T14SETUP ;DO INITIAL TEST SETUP
205	024466	012737	000024	002216	MOV	#20,,LOOPCNT ;PERFORM 20 ITERATIONS
206	024474				T14LOOP:	
207	024474				BGNSUB	
	024474					
	024474	104402				
208	024476	004737	026336		JSR	PC,T14REST ;SET PACKET TO INITIAL VALUES
209	024502	004737	026374		JSR	PC,T14RST ;SET PACKET TO INITIAL VALUES
210	024506				SETPRI	#PRI00 ;LOWER PRIORITY TO ALLOW INTERRUPTS
	024506	012700	000000			
	024512	104441				
211	024514				5\$:	
212	024514				BGNSEG	
	024514	104404				
213	024516	004737	015774		JSR	PC,SQFINIT ;DO SOFT INIT OF CONTROLLER
214	024522	103405			BCS	10\$;BR IF SOFT INIT = OK
218	024524	010001			MOV	R0,R1 ;SAVE CONTENTS OF TSSR
219	024526				ERRDF	ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
	024526	104455				
	024530	000311				
	024532	003652				
	024534	012034				
220	024536				10\$:	
221	024536	012704	025630		MOV	#T14PK2,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
222	024542	004737	010662		JSR	PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
223	024546	103405			BCS	11\$;BR, IF COMMAND ISSUED OK
227	024550	010001			MOV	R0,R1 ;SAVE CONTENTS OF TSSR
228	024552				ERRHRD	ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
	024552	104456				
	024554	000312				

	024556	005056					.WORD	WRTMSG
	024560	012034					.WORD	SFIMSG
229	024562		11\$:					
230	024562	005037	002222	CLR	FATFLG	;CLEAR FATAL ERROR FLAG		
231	024566	005037	002224	CLR	INTRECV	;CLEAR INTERRUPT RECEIVED FLAG		
232	024572	012704	025210	MOV	#T14PACKET,R4	;SET UP NEW WRT. SUBSYS MEM. COMMAND		
233	024576	010465	000000	MOV	R4,TSD8(R5)	;SET THE PACKET ADDRESS		
234	024602	004737	016336	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET		
235	024606	103407		BCS	15\$	;BR IF CARRY SET (GOOD RETURN)		
236	024610	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR		
240	024612			ERRDF	ERRNO,T14SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET		
	024612	104455					TRAP	C\$ERDF
	024614	000313					.WORD	203
	024616	026023					.WORD	T14SSR
	024620	012046					.WORD	PKTSSR
241	024622	005237	002222	INC	FATFLG	;SET FATAL ERROR FLAG		
242	024626		15\$:	CKLOOP		;LOOP ON ERROR, IF FLAG SET		
	024626	104406					TRAP	C\$CLP1
243	024630			ESCAPE	SEG	;BY-PASS SUBTEST IF FATAL ERROR		
	024630	104410					TRAP	C\$ESCAPE
	024632	000074					.WORD	10000\$-
244	024634	005737	002224	TST	INTRECV	;DID AN INTERRUPT OCCUR ?		
245	024640	001004		BNE	22\$	;BRANCH IF YES		
249	024642			ERRHRD	ERRNO,T14NINT,PKTSSR			
	024642	104456					TRAP	C\$ERHRD
	024644	000314					.WORD	204
	024646	026113					.WORD	T14NINT
	024650	012046					.WORD	PKTSSR
250	024652	016501	000002	MOV	TSSR(R5),R1	;GET THE CONTENTS OF TSSR		
251	024656	012702	000200	MOV	#SSR,R2	;EXPECTED CONTENTS OF TSSR		
252	024662	032701	000100	BIT	#OFL,R1	;IS OFF-LINE BIT SET ?		
253	024666	001402		BEQ	25\$	;BRANCH IF NOT OFF-LINE		
254	024670	052702	000100	BIS	#OFL,R2	;SET OFF-LINE IN EXPECTED DATA		
255	024674	020201		25\$:	CMP	R2,R1	;DOES EXPECTED MATCH RECEIVED ?	
256	024676	001404		BEQ	30\$	;OKAY IF MATCH		
260	024700			ERRHRD	ERRNO,T14NBA,PKTSSR	;NBA NOT ZERO		
	024700	104456					TRAP	C\$ERHRD
	024702	000315					.WORD	205
	024704	025652					.WORD	T14NBA
	024706	012046					.WORD	PKTSSR
261	024710		30\$:					
262	024710	004737	011114	35\$:	JSR	PC,CKRAM	;CHECK RAM TO MEMORY	
263	024714	103405		BCS	59\$	;RAM OK GO ON		
267	024716			ERRHRD	ERRNO,PKTRAM,RAMERR	;THEY DON'T MATCH		
	024716	104456					TRAP	C\$ERHRD
	024720	000316					.WORD	206
	024722	004745					.WORD	PKTRAM
	024724	015510					.WORD	RAMERR
268	024726			ENDSEG		;***** END SEGMENT *****		
	024726					10000\$:		
	024726	104405					TRAP	C\$ESEG
269								
270								
271	024730		59\$:	ENDSUB		;***** END SUBTEST *****		
	024730					L10040:		
	024730	104403					TRAP	C\$ESUB
272								

273	024732	005737	002222	TST	FATFLG		;ANY FATAL ERRORS ?
274	024736	001402		BEQ	60\$		;BRANCH IF NOT
275	024740	004737	017202	JSR	PC,CKDROP		;TRY TO DROP THE UNIT
276	024744			60\$:			
277				;			
278				;			
279				TEST 2, SUBTEST 2			
280				;			
281				CHECK THAT NON-ZERO MODE BITS BEING SET CAUSES			
282				WRITE SUBSYSTEM MEMORY COMMAND TO BE REJECTED			
283				;			
284				-			
285							
286	024744			BGNSUB		/	////////// BEGIN SUBTEST //////////
	024744					T2,2:	
	024744	104402				TRAP	C#BSUB
287							
288	024746			SETPRI	#PRI00		;LOWER PRIORITY TO ALLOW INTERRUPTS
	024746	012700	000000			MOV	#PRI00,R0
	024752	104441				TRAP	C#SPRI
289	024754	012704	025630	5\$:	MOV	#T14PK2,R4	;GET THE ADDRESS OF COMMAND PACKET
290	024760	004737	026336		JSR	PC,T14REST	;RESTORE PACKET TO STARTING VALUES
291	024764	004737	026374		JSR	PC,T14RST	;RESTORE PACKET TO STARTING VALUES
292	024770				BGNSEG		;>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>
	024770	104404				TRAP	C#BSEG
293							
294	024772	004737	015774		JSR	PC,SOFINIT	;DO SOFT INIT OF CONTROLLER
295	024776	103405			BCS	10\$	;BR IF SOFT INIT = OK
299	025000	010001			MOV	R0,R1	;SAVE CONTENTS OF TSSR
300	025002				ERRDF	ERRNO,SFIERR,SFIMSG	;DEVICE FATAL ERROR DURING INIT
	025002	104455				TRAP	C#ERDF
	025004	000317				.WORD	207
	025006	003652				.WORD	SFIERR
	025010	012034				.WORD	SFIMSG
301	025012			10\$:			
302	025012	004737	010662		JSR	PC,WRTCHR	;ISSUE WRITE CHARACTERISTICS
303	025016	103405			BCS	11\$	;BR, IF COMMAND ISSUED OK
307	025020	010001			MOV	R0,R1	;SAVE CONTENTS OF TSSR
308	025022				ERRHRD	ERRNO,WRTMSG,SFIMSG	;WRITE SUBSYSTEM MEMORY FAILED
	025022	104456				TRAP	C#ERHRD
	025024	000320				.WORD	208
	025026	005056				.WORD	WRTMSG
	025030	012034				.WORD	SFIMSG
309	025032			11\$:			
310	025032	005737	002224		CLR	INTRECV	;CLEAR INTERRUPT RECEIVED FLAG
311	025036	012704	025210		MOV	#T14PACKET,R4	;SET-UP WITH WRT SUBSYS MEM PACKET
312	025042	052714	007000		BIS	#007000,(R4)	;NON-ZERO COMMAND MODE BITS
313	025046	010465	000000		MOV	R4,TSD8(R5)	;SET THE PACKET ADDRESS
314	025052	004737	016250		JSR	PC,WAITF	;WAIT FOR SSR TO SET
315	025056	103405			BCS	15\$	;BR IF CARRY SET (GOOD RETURN)
316	025060	010001			MOV	R0,R1	;SAVE CONTENTS OF TSSR
320	025062				ERRDF	ERRNO,T14SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	025062	104455				TRAP	C#ERDF
	025064	000321				.WORD	209
	025066	026023				.WORD	T14SSR
	025070	012046				.WORD	PKTSSR
321	025072			15\$:	CKLOOP		;LOOP ON ERROR, IF FLAG SET

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

```

369 025226          T14BFR: .BLKW    128.          ;MESSAGE BUFFER
370
371
373          025630
375 025630          T14PK2: .<.,+10>E177770
376 025630 100204    .WORD    100204              ;COMMAND PACKET FOR TEST
377 025632 025640    .WORD    T14DTA              ;WRITE CHARA. MEM. CMND.. WITH IE. ACK
378 025634 000000    .WORD    0                   ;ADDRESS OF SELECT DATA BLOCK
379 025636 000010    .WORD    8.                  ;STARTING VALUE OF BLOCK SIZE
380
381
382 025640          T14DTA:                      ;SELECT DATA BLOCK
383 025640 025226    .WORD    T14BFR              ;ADDRESS OF MESSAGE BUFFER
384 025642 000000    .WORD    0
385 025644 000400    .WORD    256.                ;LENGTH OF MESSAGE BUFFER
386 025646 000000 000000 .WORD    0,0
387
388
389
390          ;+
391          ;LOCAL TEXT MESSAGES FOR TEST
392          ;-
393 025652          127      122      111 T14NBA: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted'
394 025726          127      122      111 T142REJ: .ASCIZ 'WRITE SUBSYSTEM MEMORY Not Rejected with Non-Zero Mode Field'
395 026023          103      157      156 T14SSR: .ASCIZ 'Contents of ISSR Incorrect After WRITE SUBSYSTEM MEMORY'
396 026113          105      170      160 T14NINT: .ASCIZ 'Expected Interrupt Not Received On WRITE SUBSYSTEM MEMORY'
397 026205          111      156      143 T14TSBA: .ASCIZ 'Incorrect TSBA Address After WRITE SUBSYSTEM MEMORY'
398 026271          102      141      163 TST14ID: .ASCIZ 'Basic WRITE SUBSYSTEM MEMORY Command'
399          .EVEN
400
401
402
403          ;+
404          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
405          ;WRITE SUBSYSTEM MEMORY COMMAND
406          ;-
407
408
409 026336          T14REST:
410 026336          SAVREG
411 026342 012701 025210 MOV     #T14PACKET,R1          ;SAVE THE REGISTERS
412 026346 012721 100206 MOV     #100206,(R1)+        ;START OF THE PACKET
413 026352 012721 025220 MOV     #T14DATA,(R1)+        ;WRITE SUBSYSTEM MEM. WITH ACK, IE
414 026356 005021      CLR     (R1)+                ;ADDRESS OF DATA BLOCK
415 026360 012721 000006 MOV     #6.,(R1)+            ;EXTENDED ADDRESS
416 026364 005021      CLR     (R1)+                ;SIZE OF DATA BLOCK IN BYTES
417 026366 005021      CLR     (R1)+                ;CLEAR BSEL0 AND BSEL1
418 026370 005011      CLR     (R1)+                ;CLEAR SEL2
419 026372 000207      CLR     (R1)+                ;CLEAR DATA AREA
420          RTS     PC                               ;RETURN
421
422 026374          T14RST:
423 026374          SAVREG
424 026400 012701 025630 MOV     #T14PK2,R1          ;SAVE THE REGISTERS
425 026404 012721 100204 MOV     #100204,(R1)+        ;START OF THE PACKET
426 026410 012721 025640 MOV     #T14DTA,(R1)+        ;WRITE CHARA. WITH ACK, IE
427 026414 005021      CLR     (R1)+                ;ADDRESS OF CHARACTERISTICS DATA BLOCK
428          ;EXTENDED ADDRESS

```

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

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

L10037: TRAP C\$ETST

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```

438
439
440 .SBTTL TEST 3: DMA MEMORY ADDRESSING
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465 026450
026450
470 026450 012700 030520
471 026454 004737 016510
472 026460 012737 000012 002216
473 026466 005237 003150
474 026472 004737 021276
475
476 026476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497

      .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

```

C9

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 25-1

SEQ 0106

TEST 3: SUBTEST 1: GET STATUS SELECTED LOCATIONS

```

498      REPEAT FOR SELECTED VALID ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
499      BEGIN
500      Get a valid modulo-4 test address
501      Do a GET STATUS command from the test address
502      END
503      END
504      ---
505
506      BGNSUB                                ;//////////////// BEGIN SUBTEST //////////////////
507      T3.1:                                TRAP      C1BSUB
508
509      ;Write to TSSR to soft initialize
510      JSR      PC,SOFINIT                    ;DO SOFT INIT OF CONTROLLER
511      BCS      15:                          ;BR IF SOFT INIT = OK
512      NEXT,ERRNO
513      MOV      R0,R1
514      ERDF     ERRNO,SFIERR,SFIMSG          ;SAVE CONTENTS OF TSSR
515      ;DEVICE FATAL ERROR DURING INIT
516      TRAP     C1ERDF
517      .WORD    301
518      .WORD    SFIERR
519      .WORD    SFIMSG
520
521      ;Do a WRITE CHARACTERISTICS to setup a message buffer
522      15:
523      MOV      @T12PACKET,R4                ;GET THE ADDRESS OF COMMAND PACKET
524      JSR      PC,T12SWRT                  ;RESTORE PACKET TO STARTING VALUES
525      CLR      KTENABLE                    ;TURN OFF KT-11
526      MOV      R4,TSD8(R5)                ;SET THE PACKET ADDRESS
527      JSR      PC,CHKTSSR                  ;WAIT FOR SSR TO SET
528      FORCERROR 17:
529      BCS      20:                          ;BR IF SSR SET IN CHKTSSR
530      MOV      R0,R1
531      NEXT,ERRNO
532      17:   ERDF     ERRNO,T12WRTSSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
533      TRAP     C1ERDF
534      .WORD    302
535      .WORD    T12WRTSSR
536      .WORD    PKTSSR
537
538      ;Verify a Get Status can be fetched from each address
539      ;Get a valid modulo-4 test address
540      ;Do a GET STATUS command from the test address
541      20:   CLR      FATFLG                ;CLEAR FATAL ERROR FLAG
542      CLR      T12KT                      ;TEST ABOVE 28K SWITCH
543      MOV      @T12BLK,R2                ;POINT TO TEST PATTERN TABLE
544      T121LOOP:
545      CLR      KTENABLE                    ;TURN OFF ABOVE 28K TEST FLAG
546      MOV      (R2)+,R1                  ;GET TEST PATTERN ADDRESS
547      CLR      R0                          ;ASSUME NO TEST ABOVE 28K
548      TST      T12KT                      ;TEST ABOVE 28K THIS TIME?
549      BEQ      25:                        ;BR IF NO
550      MOV      -2(R2),R0                  ;GET TEST PATTERN AGAIN
551      BIC      @C<A1716>,R0              ;SAVE 18 BIT ADDRESS ONLY
552      MOV      @1,KTENABLE                ;TURN ON ABOVE 28K TEST FLAG
553      25:   JSR      PC,T12CONVERT         ;CONVERT TEST PATTERN TO TEST ADDRESS

```

TEST 3: SUBTEST 1: GET STATUS SELECTED LOCATIONS

```

545 026650 103034      BCC      65%      ;BR IF INVALID PACKET ADDRESS
546 026652 013704 030354  MOV      T12LOAD,R4      ;COPY CURRENT PACKET LOW ADDRESS
547 026656 013703 030352  MOV      T12HIADD,R3      ;COPY CURRENT PACKET HIGH ADDRESS
548 026662 004737 031736  JSR      PC,T12SETGET      ;SETUP CURRENT PACKET TO GET STATUS
549 026666 042703 177774  BIC      #C<A1716>,R3      ;SAVE ADDRESS BITS 17+16
550 026672 050304      BIS      R3,R4      ;SETUP 18 BIT PACKET ADDRESS
551 026674 004737 017274  JSR      PC,KTOFF      ;TURN OFF KT-11
552 026700 010465 000000  MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
553 026704 004737 016336  JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
554 026710      FORCERROR 32%
555 026724 103405      BCS      40%      ;BR IF SSR SET IN CHKTSSR
556 026726 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
557 026730      NEXT,ERRNO
558 026730 32%:      ERDIF  ERRNO,T12GETSSR,PKTGETS ;DEVICE FATAL SSR FAILED TO SET
                    TRAP      C$ERDIF
                    .WORD      303
                    .WORD      T12GETSSR
                    .WORD      PKTGETS
559 026740 40%:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                    TRAP      C$CLP1
560 026742 65%:
561 026742      FORCEEXIT 80%
562 026752 020227 030516  CMP      R2,#T12TBE      ;DONE ALL TSTBLK TEST PATTERNS?
563 026756 103002      BHIS      70%      ;BR IF YES
564 026760 000137 026610  JMP      T121LOOP      ;DO ANOTHER MODULO- 4 ADDRESS
565 026764 005737 030360  TST      T12KT      ;DONE ABOVE 28K TESTING TOO?
566 026770 003012      UGT      80%      ;BR IF YES
567 026772 005737 003132  TST      KTFLG      ;ANY MEMORY ABOVE 28K ON SYSTEM?
568 026776 001407      BEQ      80%      ;BR IF NO
569 027000 012737 000001 030360  MOV      #1,T12KT      ;SET SWITCH
570 027006 012702 030364  MOV      #T12BLK,R2      ;RESET TEST PATTERN TABLE
571 027012 000137 026610  JMP      T121LOOP      ;DO ABOVE 28K TESTING
572 027016 004737 017274  JSR      PC,KTOFF      ;TURN OFF KT11
573 027022 80%:      ENDSUB      ;////////////////// END SUBTEST ////////////////////
                    L10043:
574 027022 104403      TST      FATALFLG      ;ANY FATAL ERRORS ?
                    TRAP      C$ESUB
575 027030 005737 002222  BEQ      100%      ;BRANCH IF NOT
576 027032 004737 017202  JSR      PC,CKDROP      ;TRY TO DROP THE UNIT
577 027036 100%:

```

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

TEST 3: SUBTEST 2:

SUBTEST DESCRIPTION:

This subtest verifies the controller can deposit message packets to all available memory locations. Write Characteristics commands are executed with message buffer addresses set to various addresses in each available memory location. The various addresses are determined by floating a 1 then a 0 through the address bits.

TEST STEPS:

TEST 3: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS

```

595      BEGIN
596      ; Write to TSSR to soft initialize
597      ; Do a WRITE CHARACTERISTICS to setup a message buffer to compare
598
599      ; REPEAT FOR SELECTED ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
600      BEGIN
601      ; Get a valid modulo-4 test address
602      ; Set the packet message buffer to the TEST ADDRESS
603      ; Do a WRITE CHARACTERISTICS
604      ; Restore the test message buffer to background pattern
605      END
606      END
607
608
609      BGNSUB                                ;////////// BEGIN SUBTEST ///////////
                                           T3.2:
                                           TRAP    C1BSUB
610
611
612      ;Write to TSSR to soft initialize
613      JSR    PC,SOFINIT                    ;DO SOFT INIT OF CONTROLLER
614      BCS    15$                          ;BR IF SOFT INIT = OK
615      NEXT.ERRNO
616      MOV    R0,R1                        ;SAVE CONTENTS OF TSSR
617      ERDIF  ERRNO,SFIERR,SFIMSG          ;DEVICE FATAL ERROR DURING INIT
                                           TRAP    C1ERDF
                                           .WORD   304
                                           .WORD   SFIERR
                                           .WORD   SFIMSG
618
619      ;Do a WRITE CHARACTERISTICS to setup a message buffer to compare
620      15$:
621      MOV    #T12PACKET,R4                ;GET THE ADDRESS OF COMMAND PACKET
622      JSR    PC,T12SWRT                    ;SET PACKET TO WRITE CHARACTERISTICS
623      JSR    PC,KTOFF                      ;TURN OFF KT-11
624      MOV    R4,TSD8(R5)                  ;SET THE PACKET ADDRESS
625      JSR    PC,CHKTSSR                    ;WAIT FOR SSR TO SET
626      FORCERROR 17$
627      BCS    20$                          ;BR IF SSR SET IN CHKTSSR
628      MOV    R0,R1                        ;SAVE CONTENTS OF TSSR
629      NEXT.ERRNO
630      17$:
631      ERDIF  ERRNO,T12WRTSSR,PKTSSR        ;DEVICE FATAL SSR FAILED TO SET
                                           TRAP    C1ERDF
                                           .WORD   305
                                           .WORD   T12WRTSSR
                                           .WORD   PKTSSR
632
633      ;Get a valid modulo-4 test address
634      ;Set the packet message buffer to the test address
635      ;Do a WRITE CHARACTERISTICS
636      20$:
637      CLR    FATFLG                        ;CLEAR FATAL ERROR FLAG
638      MOV    #T12BLK,R3                    ;POINT TO TEST PATTERN TABLE
639      T122LOOP:
640      MOV    (R3),R1                      ;GET TEST PATTERN ADDRESS
641      MOV    R1,R0                        ;GET ADDRESS ALL "18 BITS"
642      BIC    #177774,R0                    ;LEAVE ONLY A17 AND A16
643      BIC    #3,R1                        ;GET RID OF A17 AND A16

```

TEST 3: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS

```

642 027160 004737 031366      JSR      PC,T12CONVERT      ;CONVERT TEST PATTERN TO TEST ADDRESS
643 027164 103402              BCS      25$              ;BR IF VALID MESSAGE BUFFER ADDRESS
644 027166 000137 027264      JMP      150$              ;GET ANOTHER TEST PATTERN TO TRY
645 027172 012704 030310      25$:  MOV      @T12PACKET,R4      ;SET THE COMMAND PACKET ADDRESS
646 027176 004737 031670      JSR      PC,T12SWRT      ;SETUP T12PACKET TO WRITE CHAR.
647 027202 013737 030354 030320  MOV      T12LOADD,T12DATA      ;SETUP LOW ORDER MESSAGE BUFFER ADD.
648 027210 013737 030352 030322  MOV      T12HIADD,T12DATA+2      ;SETUP HIGH ORDER MESSAGE BUFFER ADD.
649 027216 004737 017274      JSR      PC,KTOFF      ;TURN OFF KT-11
650 027222 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
651 027226 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
652 027232              FORCERROR      32$
653 027246 103405              BCS      50$              ;BR IF SSR SET IN CHKTSSR
654 027250 010001              MOV      R0,R1      ;SAVE CONTENTS OF TSSR
655 027252              NEXT.ERRNO
656 027252 32$:  ERROF      ERRNO,T12WRTSSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
        027252 104455              TRAP      C$ERDF
        027254 000462              .WORD      306
        027256 030622              .WORD      T12WRTSSR
        027260 012046              .WORD      PKTSSR
657 027262 50$:  CKLOOP              ;LOOP ON ERROR, IF FLAG SET
        027262 104406              TRAP      C$CLP1
658 027264 150$:
659 027264              FORCEXIT      160$
660 027274 020327 030516      CMP      R3,@T12TBE      ;DONE ALL TST12BLK TEST PATTERNS?
661 027300 103002              BHS      160$              ;BR IF YES
662 027302 000137 027144      JMP      T122LOOP      ;DO ANOTHER MODULO- 4 ADDRESS
663 027306 004737 017274      160$: JSR      PC,KTOFF      ;TURN OFF KT11
664 027312              ENDSUB      ;////////// END SUBTEST ////////////
        027312              L10044:
        027312 104403              TRAP      C$ESUB
665 027314 005737 002222      TST      FATFLG      ;ANY FATAL ERRORS ?
666 027320 001402              BEQ      180$              ;BRANCH IF NOT
667 027322 004737 017202      JCC     PC,CKDROP      ;TRY TO DROP THE UNIT
668 027326 180$:
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691

```

.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

```

692      BEGIN
693      ;
694      ;   Get a valid test address
695      ;   Set the test packet characteristics data pointer to the
696      ;   test address.
697      ;   Store expected characteristic data in test address block
698      ;   Do a WRITE CHARACTERISTIC command
699      ;
700      ;   END
701      ;--
702      BGNSUB                                ;////////// BEGIN SUBTEST //////////
703      ;                                     T3.3:
704      ;                                     TRAP    C$BSUB
705      ;
706      ;Write to TSSR to soft initialize
707      JSR    PC,SOFINIT                    ;DO SOFT INIT OF CONTROLLER
708      BCS    20$                          ;BR IF SOFT INIT = OK
709      NEXT.ERRNO
710      MOV     R0,R1                        ;SAVE CONTENTS OF TSSR
711      ERRDF   ERRNO,SFIERR,SFIMSG         ;DEVICE FATAL ERROR DURING INIT
712      ;                                     TRAP    C$ERDF
713      ;                                     .WORD   307
714      ;                                     .WORD   SFIERR
715      ;                                     .WORD   SFIMSG
716      ;
717      ;Get a valid test address
718      20$:  CLR    FATFLG                    ;CLEAR FATAL ERROR FLAG
719      CLR    T12KT                        ;TEST ABOVE 28K SWITCH
720      MOV     R1,R0                      ;POINT TO TEST PATTERN TABLE
721      T123LOOP:
722      CLR    KTENABLE                    ;TURN OFF ABOVE 28K TEST FLAG
723      MOV     (R3)+,R1                  ;GET TEST PATTERN ADDRESS
724      MOV     R1,R0                      ;GET ADDRESS ALL "18 BITS"
725      BIC     R1,R0                      ;LEAVE ONLY A17 AND A16
726      BIC     R1,R0                      ;GET RID OF A17 AND A16
727      TST     T12KT                      ;TEST ABOVE 28K THIS TIME?
728      BEQ     25$                        ;BR IF NO
729      MOV     -2(R3),R0                  ;GET TEST PATTERN AGAIN
730      BIC     R1,R0                      ;SAVE 18 BIT ADDRESS ONLY
731      MOV     R1,KTENABLE                ;TURN ON ABOVE 28K TEST FLAG
732      25$:  JSR    PC,T12CONVERT          ;CONVERT TEST PATTERN TO TEST ADDRESS
733      BCS     30$                        ;BR IF VALID TEST ADDRESS
734      JMP     60$                        ;GET NEXT TEST PATTERN
735      ;Set the test packet characteristics data pointer to the test address
736      30$:  MOV     R4,T12PACKET,R4        ;GET THE ADDRESS OF COMMAND PACKET
737      JSR     PC,T12SWRT                 ;RESTORE PACKET TO STARTING VALUES
738      MOV     R4,T12LOAD,PKLOW(R4)       ;STORE CHAR. DATA PTR LOW ADDRESS
739      MOV     R4,T12HIAUD,PKHI(R4)       ;STORE CHAR. DATA PTR HIGH ADDRESS
740      JSR     PC,T12CHAR                 ;STORE EXPECTED DATA IN DATA BLOCK
741      ;Do a WRITE CHARACTERISTIC command
742      JSR     PC,KTOFF                    ;TURN OFF KT-11
743      MOV     R4,TSDB(R5)                ;SET THE PACKET ADDRESS TO EXECUTE
744      JSR     PC,CHKTSSR                 ;WAIT FOR SSR TO SET
745      FORCERROR 32$                      ;
746      BCS     40$                        ;BR IF SSR SET IN CHKTSSR
747      MOV     R0,R1                      ;SAVE CONTENTS OF TSSR

```

TEST 3: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS

```

743 027526      NEXT.ERRNO
744 027526      32$:  ERRDF  ERRNO,T12WRTSSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
      027526 104455      TRAP  C$ERDF
      027530 000464      .WORD 308
      027532 030622      .WORD T12WRTSSR
      027534 012046      .WORD PKTSSR
745 027536      40$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      027536 104406      TRAP  C$CLP1
746 027540      60$:
747 027540 020327 030516      CMP  R3,#T12TBE      ;DONE ALL TSTBLK TEST PATTERNS?
748 027544 103002      BHIS  65$      ;BR IF YES
749 027546 000137 027364      JMP  T123LOOP      ;DO ANOTHER MODULO- 4 ADDRESS
750 027552 005737 030360      65$:  TST  T12KT      ;DONE ABOVE 28K TESTING TOO?
751 027556 003012      BGT  70$      ;BR IF YES
752 027560 005737 003132      TST  KTFLG      ;ANY MEMORY ABOVE 28K ON SYSTEM?
753 027564 001407      BEQ  70$      ;BR IF NO
754 027566 012737 000001 030360      MOV  #1,T12KT      ;SET SWITCH
755 027574 012703 030364      MOV  #T12BLK,R3      ;RESET TEST PATTERN TABLE
756 027600 000137 027364      JMP  T123LOOP      ;DO ABOVE 28K TESTING
757 027604 004737 017274      70$:  JSR  PC,KTOFF      ;TURN OFF KT11
758 027610      ENDSUB      ;///////////////// END SUBTEST ///////////////////
      027610      L10045:
      027610 104403      TRAP  C$ESUB
759 027612 005737 002222      TST  FATFLG      ;ANY FATAL ERRORS ?
760 027616 001402      BEQ  75$      ;BRANCH IF NOT
761 027620 004737 017202      JSR  PC,CKDROP      ;TRY TO DROP THE UNIT
762 027624      75$:

```

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

```

763
764
765 ;**
766 ; TEST 3: SUBTEST 4:

```

; SUBTEST DESCRIPTION:

```

767
768 ;
769 ; This subtest verifies the NXM error bit in the TSSR
770 ; register is set when attempting to fetch data (a characteristic
771 ; data block) from selected nonexistent locations.
772 ; If NXM fails to set it is likely that an LSI-11 Bus driver is
773 ; failing to assert an address line.
774 ; Addresses tested include all combinations of high-order address
775 ; bits (i.e bits 16-21).
776 ;

```

***** CAUTION *****

```

777
778 ;
779 ; The LSI BUS drivers for all available address lines(16-21)
780 ; are only checked when running on a 11/23B system with more than
781 ; 128K words of memory!
782 ;

```

; TEST STEPS:

; BEGIN

```

783
784 ; Write to TSSR to soft initialize
785 ; Do a write characteristic command
786 ; Invert the extended features switch
787 ;

```

```

788
789 ; REPEAT FOR SELECTED NON-EXISTENT MEMORY ADDRESSES
790
791
792

```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

793      BEGIN
794      :      Get an invalid test address
795      :      Set the test packet characteristics data pointer to the
796      :      test address.
797      :      Do a WRITE CHARACTERISTIC command
798      :      If TSSR register NXM bit not set then print error message
799      :      END
800      :      END
801      :--
802
803 027624      BGNSUB      ;//////////////// BEGIN SUBTEST //////////////////
      027624      T3.4:
      027624 104402      TRAP      C$BSUB
804
805
806 027626 005737 003144      TST      T23A      ;26-APR-83 REV B - CHK FOR 23A CPU
807 027632 001406      BEQ      5$      ;26-APR-83 REV B - BR, IF NOT 23A
808 027634 023727 002120 007777      CMP      L$HIME,07777      ;26-APR-83 REV B - CHK FOR > 256KB
809 027642 103402      BLO      5$      ;26-APR-83 REV B - BR, IF < 256KB
810 027644 000137 030236      JMP      NOEXTF      ;26-APR-83 REV B - JMP OVER 256KB
811 027650      5$:
812 027650 005737 003136      TST      NXMFLG      ;GOT ENOUGH MEMORY?
813 027654 001002      BNE      10$      ;IF SET STAY
814 027656 000137 030236      JMP      NOEXTF      ;LEAVE IF NOT SET
815
816      ;Write to TSSR to soft initialize
817
818 027662 004737 015774      10$:      JSR      PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
819 027666 103405      BCS      11$      ;BR IF SOFT INIT = OK
820 027670      NEXT,ERRNO
821 027670 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
822 027672      ERDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL ERROR DURING INIT
      027672 104455      TRAP      C$ERDF
      027674 000465      .WORD      309
      027676 003652      .WORD      SFIERR
      027700 012034      .WORD      SFIMSG
823
824      ;Do a WRITE CHARACTERISTIC command so to invert switch
825
826 027702      11$:      CKLOOP      ;LOOP IF SELECTED
      027702 104406      TRAP      C$CLP1
827 027704 012704 030310      MOV      @T12PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
828 027710 004737 031670      JSR      PC,T12SWRT      ;RESTORE PACKET TO STARTING VALUES
829 027714 005037 003134      CLR      KTENABLE      ;TURN OFF KT-11
830 027720 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS
831 027724 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
832 027730      FORCERROR      15$
833 027744 103405      BCS      17$      ;BR IF SSR SET IN CHKTSSR
834 027746 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
835 027750      NEXT,ERRNO
836 027750      15$:      ERDF      ERRNO,T12WRTSSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      027750 104455      TRAP      C$ERDF
      027752 000466      .WORD      310
      027754 030622      .WORD      T12WRTSSR
      027756 012046      .WORD      PKTSSR
837 027760      17$:      CKLOOP      ;LOOP IF SELECTED
      027760 104406      TRAP      C$CLP1

```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

838 027762 004737 021206          JSR      PC,INVERT          ;INVERT THE SWITCH
839
840          ;Get an invalid test address
841
842 027766 005037 002222          20$:   CLR      FATFLG          ;CLEAR FATAL ERROR FLAG
843 027772          25$:
844 027772 013737 003142 030352      MOV      NXMHI,T12HIADD      ;SAVE TEST ADDRESS HIGH
845 030000 013737 003140 030354      MOV      NXML0,T12LOADD      ;SAVE TEST ADDRESS LOW
846 030006          T124LOOP:
847
848          ;Set the test packet characteristics data pointer to the
849          ; test address.
850
851 030006 012704 030310          30$:   MOV      @T12PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
852 030012 004737 031670          JSR      PC,T12SWRT          ;RESTORE PACKET TO STARTING VALUES
853 030016 013764 030354 000002      MOV      T12LOADD,PKLOW(R4)      ;STORE CHAR. DATA PTR LOW ADDRESS
854 030024 013764 030352 000004      MOV      T12HIADD,PKHI(R4)      ;STORE CHAR. DATA PTR HIGH ADDRESS
855
856          ;Do a WRITE CHARACTERISTIC command
857 030032 004737 017274          JSR      PC,KTOFF          ;TURN OFF KT-11
858 030036 010465 000000          MOV      R4,TSD8(R5)          ;SET THE PACKET ADDRESS TO EXECUTE
859 030042 004737 016250          JSR      PC,WAITF          ;WAIT FOR SSR TO SET
860 030046          FORCERROR          32$:
861 030062 103407          BCS      40$          ;BR IF SSR SET IN CHKTSSR
862 030064 010001          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
863 030066          NEXT,ERRNO
864 030066          32$:   ERRDF      ERRNO,T12WRTSSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
865          030066 104455          TRAP      C$ERDF
866          030070 000467          .WORD      311
867          030072 030622          .WORD      T12WRTSSR
868          030074 012046          .WORD      PKTSSR
869
870 030076 005237 002222          40$:   INC      FATFLG          ;SET FATAL ERROR FLAG
871 030102          CKLOOP          ;LOOP ON ERROR, IF FLAG SET
872 030102 104406          TRAP      C$CLP1
873
874 030104          FORCERROR          45$:NOTSSR
875 030114          ESCAPE SUB          ;BY-PASS SUBTEST IF FATAL ERROR
876          030114 104410          TRAP      C$ESCAPE
877          030116 000124          .WORD      L10046-.
878
879          ;If TSSR register NXM bit not set then print error message
880          45$:
881          MOV      TSSR(R5),R1          ;GET TSSR CONTENTS
882          FORCERROR          52$:
883          BIT      @NXM,R1          ;NXM SET?
884          BNE      60$          ;BR IF YES
885          NEXT,ERRNO
886          52$:   MOV      T12LOADD,ERRLO          ;MEMORY TEST ADDRESS LOW
887          MOV      T12HIADD,ERRHI          ;MEMORY TEST ADDRESS HIGH
888          ERRL      ERRNO,T12NXM,ADDSR      ;REPORT ADDRESS AND TSSR ERROR
889          TRAP      C$ERHRO
890          .WORD      312
891          .WORD      T12NXM
892          .WORD      ADDSR
893
894          60$:   CKLOOP          ;LOOP ON ERROR, IF FLAG SET
895          TRAP      C$CLP1
896
897          FORCEXIT          90$
898          TST      T23A          ;IS IT A 11/23A?

```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

883 030210 001012      BNE      90$
884 030212 013700 030352      MOV      T12HIADD,R0
885 030216 005200      65$:      INC      R0
886 030220 020027 000077      CMP      R0,#77
887 030224 101004      BHI      90$
888 030226 010037 030352      75$:      MOV      R0,T12HIADD
889 030232 000137 030006      JMP      T124LOOP
890 030236
891 030236      90$:
      NOEXTF:
892 030236 004737 017274      JSR      PC,KTOFF
893 030242      ENDSUB
      030242
      030242 104403
894 030244 005737 002222      TST      FATFLG
895 030250 001402      BEQ      100$
896 030252 004737 017202      JSR      PC,CKDROP
897 030256 004737 016456      100$:      JSR      PC,TSTLOOP
898 030262 103002      BCC      105$
899 030264 000137 026476      JMP      T12LOOP
900 030270
901 030270 004737 017274      105$:      JSR      PC,KTOFF
902 030274 005037 003150      CLR      T3BFLG
903 030300      EXIT      TST
      030300 104432
      030302 001540
904
905
906
907
908
909
911      030310
913 030310      T12PACKET:
914 030310 100004      .WORD      100004
915 030312 030320      .WORD      T12DATA
916 030314 000000      .WORD      0
917 030316 000010      .WORD      8.
918
919 030320      T12DATA:
920 030320 030332      .WORD      T12BFR
921 030322 000000      .WORD      0
922 030324 000016      .WORD      14.
923 030326 000000 000000      .WORD      0,0
924
925 030332      T12BFR: .BLKW      8.
926
927 030352 000000      T12HIADD: .WORD      0
928 030354 000000      T12LOADD: .WORD      0
929 030356 000000      T12PAR6:  .WORD      0
930 030360 000000      T12KT:   .WORD      0
931 030362 000000      T124TST: .WORD      0
932
933
934      ;
935      ;TABLE OF ADDRESSES
936      ;
937 030364 000001      T12BLK: .WORD      000001

```

; YES WERE DONE
 ; GET CURRENT HIGH ADDRESS
 ; GET NEXT ADDRESS
 ; DONE A21-A16?
 ; BR IF YES
 ; SETUP NEW HIGH ORDER ADDRESS
 ; DO ANOTHER NON-EXISTENT ADDRESS

 ; 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-.

;+
 ; LOCAL STORAGE FOR THIS TEST
 ;-

 .<+.10>E177770
 T12PACKET:
 .WORD 100004
 .WORD T12DATA
 .WORD 0
 .WORD 8.

 T12DATA:
 .WORD T12BFR
 .WORD 0
 .WORD 14.
 .WORD 0,0

 T12BFR: .BLKW 8.

 T12HIADD: .WORD 0
 T12LOADD: .WORD 0
 T12PAR6: .WORD 0
 T12KT: .WORD 0
 T124TST: .WORD 0
 ;+
 ;
 ; TABLE OF ADDRESSES
 ;
 ;-
 T12BLK: .WORD 000001

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

938	030366	000002	.WORD	000002
939	030370	000003	.WORD	000003
940	030372	000005	.WORD	000005
941	030374	000006	.WORD	000006
942	030376	000007	.WORD	000007
943	030400	000011	.WORD	000011
944	030402	000012	.WORD	000012
945	030404	000013	.WORD	000013
946	030406	000021	.WORD	000021
947	030410	000022	.WORD	000022
948	030412	000023	.WORD	000023
949	030414	000041	.WORD	000041
950	030416	000042	.WORD	000042
951	030420	000043	.WORD	000043
952	030422	000101	.WORD	000101
953	030424	000102	.WORD	000102
954	030426	000103	.WORD	000103
955	030430	000201	.WORD	000201
956	030432	000202	.WORD	000202
957	030434	000203	.WORD	000203
958	030436	000401	.WORD	000401
959	030440	000402	.WORD	000402
960	030442	000403	.WORD	000403
961	030444	001001	.WORD	001001
962	030446	001002	.WORD	001002
963	030450	001003	.WORD	001003
964	030452	002001	.WORD	002001
965	030454	002002	.WORD	002002
966	030456	002003	.WORD	002003
967	030460	004001	.WORD	004001
968	030462	004002	.WORD	004002
969	030464	004003	.WORD	004003
970	030466	010001	.WORD	010001
971	030470	010002	.WORD	010002
972	030472	010003	.WORD	010003
973	030474	020001	.WORD	020001
974	030476	020002	.WORD	020002
975	030500	020003	.WORD	020003
976	030502	040001	.WORD	040001
977	030504	040002	.WORD	040002
978	030506	040003	.WORD	040003
979	030510	100001	.WORD	100001
980	030512	100002	.WORD	100002
981	030514	100003	.WORD	100003
982	030516	177777	T12TBE: .WORD	177777

```

;+
;LOCAL TEXT MESSAGES FOR TEST
;-

```

987	030520	104	115	101	T12TID:	.ASCIIZ	'DMA Memory Addressing'
988	030546	103	157	156	T12GETSSR:	.ASCIIZ	'Contents of TSSR Incorrect After GET STATUS'
989	030622	103	157	156	T12WRTSSR:	.ASCIIZ	'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
990	030711	115	145	163	T12MSGBUF:	.ASCIIZ	'Message Buffer Contents Incorrect After WRITE CHARACTERISTICS'
991	031007	102	141	143	T12BKGN0:	.ASCIIZ	'Background Pattern Disturbed By WRITE CHARACTERISTICS'
992	031075	105	170	160	T12NINT:	.ASCIIZ	'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
993	031166	127	162	151	T12DPR:	.ASCIIZ	'Write Characteristic data in ram does not match expected'
994	031257	124	123	123	T12NXM:	.ASCIIZ	'TSSR NXM bit failed to set when non-existent memory address specifi

ed'

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

995                                     .EVEN
996
997
998
999
1000 ROUTINE TO CONVERT A TEST PATTERN TO A VALID ADDRESS IN DIAGNOSTIC FREE SPACE
1001 :   DIAGNOSTIC FREE SPACE IS BETWEEN THE END OF THE DIAGNOSTIC AND THE
1002 :   BEGINNING OF THE SUPERVISOR. THIS IS ALWAYS BELOW 24K.
1003 :   IF MEMORY ABOVE 28K SPECIFIED (VIA R1) THEN PAR 6 IS SET
1004 :   TO THE RELOCATION BASE.
1005
1006 :   INPUTS:
1007 :       R0       HIGH ORDER ADDRESS BITS
1008 :       R1       LOW ORDER ADDRESS BITS
1009
1010 :   OUPUTS:
1011 :       T12PAR6 = ADDRESS BIASED TO PAR6 IF >28K UNDER TEST
1012 :       T12HIADD = HIGH ORDER ADDRESS IN NON PAR6 FORMAT
1013 :       T12LOADD = LOW ORDER ADDRESS IN NON PAR6 FORMAT
1014 :       C BIT = 1 IF GOOD ADDRESS RETURNED
1015 :       C BIT = 0 IF TEST PATTERN DID NOT YIELD A VALID ADDRESS
1016
1017 031366 T12CONVERT:
1018 031366 SAVREG
1019 031372 005037 030354 CLR T12LOADD ;SAVE R1-R5 UNTIL NEXT RETURN
1020 031376 005037 030352 CLR T12HIADD ;CLEAR LOW ADDRESS
1021 031402 005037 030356 CLR T12PAR6 ;CLEAR HIGH ADDRESS
1022 031406 042701 170000 BIC #C<7777>,R1 ;CLEAR PAR6 BIASED ADDRESS
1023 031412 010005 MOV R0,R5 ;FORCE TO LOWER 12 BITS OF ADDRESS
1024 031414 004737 017274 JSR PC,KTOFF ;SAVE HIGH ORDER ADDRESS BITS
1025 031420 013702 003124 MOV FREE,R2 ;SHUTOFF MEMORY MANAGEMENT
1026 031424 062702 000020 ADD #16.,R2 ;GET FIRST FREE ADDRESS
1027 031430 060102 ADD R1,R2 ;IN CASE TEST PATTERN=0
1028 031432 042702 000003 BIC #3,R2 ;ADD IN TEST PATTERN
1029 031436 013703 003130 25$: MOV FREEHI,R3 ;MAKE IT MODULO-4
1030 031442 162703 000020 SUB #16.,R3 ;GET LAST FREE ADDRESS
1031 031445 010237 030354 MOV R2,T12LOADD ;SAVE AT LEAST 8 WORDS (IN CASE MESSAGE BUFFER)
1032 031452 010237 030356 MOV R2,T12PAR6 ;SAVE POSSIBLE LOW ADDRESS
1033 031456 020203 CMP R2,R3 ;SAVE IT IN PAR6 BIASED TOO
1034 031460 101007 BHI 35$ ;IS THIS ADDRESS ABOVE FREE SPACE?
1035 031462 020237 003124 CMP R2,FREE ;BR IF YES
1036 031466 103007 BHIS 50$ ;IS IT IN FREE SPACE?
1037 031470 005737 003134 TST KTENABLE ;BR IF YES- ITS GOOD
1038 031474 001004 BNE 50$ ;TESTING ABOVE 28K?
1039 031476 000424 BR 90$ ;BR IF YES
1040 031500 162702 000020 35$: SUB #16.,R2 ;BR IF NOT IN FREE SPACE
1041 031504 000754 BR 25$ ;FORCE FIT THE TEST PATTERN
1042 031506 50$: TST KTENABLE ;TRY THIS TEST PATTERN ADDRESS
1043 031506 005737 003134 TST KTENABLE ;TESTING ABOVE 28K?
1044 031512 001420 BEQ 100$ ;BR IF NO
1045 031514 005737 003132 TST KIFLG ;ANY MEMORY ABOVE 28K?
1046 031520 001413 BEQ 90$ ;BR IF NO
1047 031522 004737 017256 JSR PC,KTON ;TURN ON MEMORY MANAGEMENT
1048 031526 010500 MOV R5,R0 ;GET HIGH ORDER ADDRESS
1049 031530 010037 030352 MOV R0,T12HIADD ;SAVE POSSIBLE HIGH ADDRESS
1050 031534 010201 MOV R2,R1 ;GET COMPUTED LOW ORDER ADDRESS
1051 031536 004737 017316 JSR PC,SETMAP ;RETURN PAR6 BIASED ADDRESS IN R0

```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

Address	Hex Address	Hex Data	Hex Data	Label	Operation	Comment
1052	031542	010037	030356		MOV R0,T12PAR6	;COPY PAR6 BIASED ADDRESS
1053	031546	103403			BCS 105\$	;BR IF VALID ADDRESS
1054	031550	000241		90\$:	CLC	;CLR C BIT FOR FAILURE
1055	031552	000401			BR 105\$	
1056	031554	000261		100\$:	SEC	;SET SUCCESS
1057	031556	000207		105\$:	RTS	;RETURN
1058						
1059						
1060						
1061						
1062						
1063						
1064						
1065						
1066						
1067						
1068						
1069						
1070						
1071						
1072						
1073						
1074						
1075						
1076						
1077						
1078						
1079						
1080						
1081						
1082						
1083						
1084						
1085						
1086	031560					
1087	031560					
1088	031564	017701	002242		SAVREG	;SAVE THE GENERAL REGISTERS
1089	031570	012702	000201		MOV #RAMDATA,R1	;ADDRESS TO SAVE THE RAM DATA
1090	031574	005003			MOV #RMPKTBEG,R2	;BYTE ADDRESS OF FIRST RAM DATA
1091	031576	004737	016336		CLR R3	;CLEAR THE ERROR FLAG
1092	031602	112765	000000	000000	JSR PC,CHKTSSR	;WAIT FOR SSR
1093	031610	004737	016336		MOVB #0,TSDB(R5)	;SET MAINTENANCE MODE
1094	031614	010265	000000	10\$:	JSR PC,CHKTSSR	;WAIT FOR SSR TO SET
1095	031620	004737	016336		MOV R2,TSDB(R5)	;SELECT NEXT RAM ADDRESS
1096	031624	116511	000000		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET
1097	031630	122124			MOVB TSBA(R5),(R1)	;READ THE RAM DATA
1098	031632	001401			CMPB (R1)+,(R4)+	;COMPARE TO EXPECTED
1099	031634	005203			BEQ 20\$	;BRANCH IF OK
1100	031636	005202		20\$:	INC R3	;SET ERROR FLAG
1101	031640	020227	000203		INC R2	;ADDRESS OF NEXT RAM LOCATION
1102	031644	002761			CMP R2,#RMPKTBEG+2	;DONE 2 BYTES?
1103	031646	005703			BLT 10\$	;BR IF NO
1104	031650	001402			TST R3	;WAS AN ERROR FOUND ?
1105	031652	000241			BEQ 30\$	;BRANCH IF NOT
1106	031654	000401			CLC	;CLEAR CARRY TO SHOW ERROR
1107	031656	000261		30\$:	BR 50\$	;AND EXIT
1108	031660	012737	000002	002302	SEC	;SHOW GOOD COMPARE
				50\$:	MOV #2,RAMSIZ	;SETUP RAMSIZ

B10

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 25-13

SEQ 0118

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

1109 031666 000207          RTS      PC              ;RETURN
1110
1111
1112
1113
1114
1115
1116 031670          T12SWRT;
1117 031670          SAVREG          ;SAVE THE REGISTERS
1118 031674 012701 030310      MOV      #T12PACKET,R1      ;START OF THE PACKET
1119 031700 012721 100004      MOV      #100004,(R1);      ;WRITE CHARACTERISTICS WITH ACK
1120 031704 012721 030320      MOV      #T12DATA,(R1);      ;ADDRESS OF CHAR DATA BLOCK
1121 031710 005021            CLR      (R1);                ;EXTENDED ADDRESS
1122 031712 012721 000010      MOV      #8,(R1);            ;SIZE OF DATA BLOCK IN BYTES
1123 031716 012721 030332      MOV      #T12BFR,(R1);        ;ADDRESS OF MESSAGE BUFFER
1124 031722 005021            CLR      (R1);
1125 031724 012721 000016      MOV      #14,(R1);            ;LENGTH OF MESSAGE BUFFER
1126 031730 005021            CLR      (R1);
1127 031732 005011            CLR      (R1);
1128 031734 000207          RTS      PC              ;RETURN
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139 031736          T12SETGET;
1140 031736          SAVREG          ;SAVE THE REGISTERS
1141 031742 010401            MOV      R4,R1                ;GET LOW ORDER ADDRESS
1142 031744 005737 003134      TST      KTENABLE          ;TESTING ABOVE 28K?
1143 031750 001404            BEQ      101;                ;BR IF NO
1144 031752 010300            MOV      R3,R0                ;GET HIGH ORDER ADDRESS
1145 031754 004737 017316      JSR      PC,SETHAP          ;RETURN ADDRESS BIASED TO PAR6 IN R0
1146 031760 010001            MOV      R0,R1                ;GET ADDRESS
1147 031762 012700 000017      101:  MOV      #P.GETSTATUS,R0 ;GET STATUS COMMAND CODE NO IE
1148 031766 052700 100000      BIS      #P.ACK,R0          ;SET ACK
1149 031772 010021            MOV      R0,(R1);            ;STORE GET STATUS IN PACKET
1150 031774 005021            CLR      (R1);                ;CLEAR UNUSED WORD
1151 031776 000207          RTS      PC              ;RETURN
1152
1153
1154
1155
1156
1157
1158
1159 032000          T12CHAR;
1160 032000          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
1161 032004 012700 030320      MOV      #T12DATA,R0        ;GET T12PACKET DATA POINTER
1162 032010 013701 030354      MOV      T12LOAD,R1        ;ASSUME NOT ABOVE 28K
1163 032014 005737 003134      TST      KTENABLE          ;TESTING ABOVE 28K?
1164 032020 001402            BEQ      101;                ;BR IF NO
1165 032022 013701 030356      MOV      T12PAR6,R1         ;SET TEST ADDRESS ABOVE 28K

```

C10

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 25-14

SEQ 0119

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
1166 032026 012021      104:  MOV      (R0)+,(R1)+      ;STORE DATA WORD 1
1167 032030 012021      MOV      (R0)+,(R1)+      ;STORE DATA WORD 2
1168 032032 012021      MOV      (R0)+,(R1)+      ;STORE DATA WORD 3
1169 032034 012021      MOV      (R0)+,(R1)+      ;STORE DATA WORD 4
1170 032036 012021      MOV      (R0)+,(R1)+      ;STORE DATA WORD 5
1171 032040 000207      RTS      PC              ;RETURN
1172
1173 032042      ENDTST
      032042
      032042 104401
```

L10042: TRAP C1ETST

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

1175
1176
1177
1178
1179
1180
1181
1182
1183
1184 032044
1185 032044
1186
1191 032044 005737 002214
1192 032050 001402
1193 032052 005237 003400
1194 032056 012700 034503
1195 032062 004737 016510
1196 032066 012737 000005 002216
1197 032074
1198
1199
1200
1201
1202
1203
1204
1205
1206 032074
1207 032074 104402
1208 032076
1209 032076 012700 000000
1210 032102 104441
1211 032104 005737 003400
1212 032110 001402
1213 032112 000137 032374
1214 032116 004737 034522
1215 032122 004737 034574
1216 032126 004737 015774
1217 032132 103405
1218 032134 010001
1219 032136
1220 032136 104455
1221 032140 000621
1222 032142 003652
1223 032144 012034
1224 032146
1225 032146 012704 033420
1226 032152 004737 010662
1227 032156 103405
1228 032160 010001
1229 032162
1230 032162 104456
1231 032164 000622
1232 032166 005056
1233 032170 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
; 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
;
; T4.1:
;
; SETPRI #PRI00 ;LOWER PRIORITY TO ALLOW INTERRUPTS
;
; TRAP C$BSUB
; MOV #PRI00,R0
; TRAP C$SPRI
;
; TST SKIPT ;SHOULD WE SKIP THIS SUBTEST
; BEQ 10$ ;BR, IF NOW SKIP REQUIRED
; JMP 50$ ;SKIP SUBTEST
; 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 ERNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
;
; TRAP C$ERDF
; .WORD 401
; .WORD SFIERR
; .WORD SFIMSG
;
; 20$:
;
; 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 ERNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
;
; TRAP C$ERHRD
; .WORD 402
; .WORD WRTMSG
; .WORD SFIMSG

```

E10

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 26-1

SEQ 0121

TEST 4: RAM EXERCISER TEST

1229	032172	012703	000400	23:	MOV	#256, R3	; STARTING ADDRESS FOR RAM WRITE
1230	032176	112737	000001		MOVB	#1, T158S1	; SIZE OF TRANSFER
1231	032204	112737	000002		MOVB	#2, T158S0	; WRITE RAM "COMMAND"
1232	032212			25:			
1233	032212	010337	034132		MOV	R3, T15S2	; ADDRESS FOR RAM
1234	032216	012704	034120		MOV	#T15PK2, R4	; WRITE SUBSYS MEM PACKET
1235	032222	110337	034134		MOVB	R3, T15S3	; DATA FOR WRITE (ADDRESS)
1236	032226	010465	000030		MOV	R4, TSDB(R5)	; ISSUE COMMAND
1237	032232	004737	016336		JSR	PC, CHKTSSR	; WAIT FOR SSR
1238	032236	103407			BCS	30:	; BR, IF NO ERROR
1239	032240	010001			MOV	R0, R1	; ERROR, SAVE TSSR
1243	032242				ERRHRD	ERRNO, T15SSR, PKTSSR	; TSSR NOT CORRECT AFTER WRITE SUB MEM
	032242	104456					TRAP C\$ERHRD
	032244	000623					.WORD 403
	032246	034136					.WORD T15SSR
	032250	012046					.WORD PKTSSR
1244	032252				ESCAPE	SUB	; DON'T CONTINUE IF ERROR ON WRITE
	032252	104410					TRAP C\$ESCAPE
	032254	000122					.WORD L10050-
1245	032256			30:	CKLOOP		; SCOPE LOOP
	032256	104406					TRAP C\$CLP1
1246							
1247							
1248	032260	005203			INC	R3	; NEXT ADDRESS
1249	032262	020327	010000		CMP	R3, #10000	; END OF RAM MEMORY CHECK
1250	032266	001351			BNE	25:	; LOOP TILL ALL RAM WRITTEN
1251	032270	005002			CLR	R2	; CLEAR OUT R2 HIGH BITS
1252	032272	005303			DEC	R3	; SET BACK TO 7777
1253	032274	110337	034134	40:	MOVB	R3, T15S3	; GET DATA PATTERN BACK IN SHAPE
1254	032300	010337	034132		MOV	R3, T15S2	; ADDRESS FOR RAM READ
1255	032304	112737	000001		MOVB	#1, T158S0	; READ RAM COMMAND
1256	032312	010465	000000		MOV	R4, TSDB(R5)	; SEND OUT PACKET ADDRESS TO CONTR.
1257	032316	004737	016336		JSR	PC, CHKTSSR	; WAIT FOR READY, NON-AMBIGUOUS
1258	032322	103405			BCS	43:	; BR, IF NO PROBLEM
1259	032324	010001			MOV	R0, R1	; SAVE TSSR
1263	032326				ERRDF	ERRNO, T15SSR, PKTSSR	; TSSR NOT CORRECT
	032326	104455					TRAP C\$ERDF
	032330	000624					.WORD 404
	032332	034136					.WORD T15SSR
	032334	012046					.WORD PKTSSR
1264	032336			43:	CKLOOP		; SCOPE LOOP
	032336	104406					TRAP C\$CLP1
1265	032340	013701	033462		MOV	T15BFR+20, R1	; GET RAM READ DATA
1266	032344	010302			MOV	R3, R2	; SET UP FOR COMPARE
1267	032346	120102			CMPB	R1, R2	; CHECK WITH DATA WRITTEN
1268	032350	001404			BEG	45:	; BR IF OK, DATA IN = DATA OUT
1272	032352				ERRHRD	ERRNO, T15AM4, EXPBREC	; WRITTEN DATA NOT = TO READ
	032352	104456					TRAP C\$ERHRD
	032354	000625					.WORD 405
	032356	034417					.WORD T15AM4
	032360	015502					.WORD EXPBREC
1273	032362			45:	CKLOOP		; SCOPE LOOP
	032362	104406					TRAP C\$CLP1
1274	032364	005303			DEC	R3	; DROP DATA COUNTER (PATTERN)
1275	032366	020327	000377		CMP	R3, #255,	; AT BOTTOM YET
1276	032372	001340			BNE	40:	; BR, IF MORE TO CHECK
1277	032374			50:	CKLOOP		; SCOPE LOOP

TEST 4: RAM EXERCISER TEST

Line	Address	Hex	Label	Code	Comment	Trap	Code
1278	032374	104406		ENDSUB	////////// END SUBTEST //////////	TRAP	C\$CLP1
	032376				L10050;		
	032376	104403				TRAP	C\$ESUB
1279							
1280							
1281	032400		BGNSUB		////////// BEGIN SUBTEST //////////		
	032400				T4.2;		
	032400	104402				TRAP	C\$BSUB
1282							
1283							
1284							
1285							
1286							
1287							
1288							
1289							
1290							
1291	032402	004737	034522	JSR	PC,T15REST		
1292	032406	004737	034574	JSR	PC,T15RT2		
1293	032412	004737	015774	JSR	PC,SOFINIT		
1294	032416	103405		BCS	20\$		
1298	032420	010001		MOV	R0,R1		
1299	032422			ERRDF	ERRNO,SFIERR,SFIMSG		
	032422	104455					
	032424	000626				TRAP	C\$ERDF
	032426	003652				.WORD	406
	032430	012034				.WORD	SFIERR
						.WORD	SFIMSG
1300	032432			20\$:			
1301	032432	012704	033420	MOV	#T15PACKET,R4		
1302	032436	004737	010662	JSR	PC,WRTCHR		
1303	032442	103405		BCS	25\$		
1307	032444	010001		MOV	R0,R1		
1308	032446			ERRHRD	ERRNO,WRTMSG,SFIMSG		
	032446	104456					
	032450	000627				TRAP	C\$ERHRD
	032452	005056				.WORD	407
	032454	012034				.WORD	WRTMSG
						.WORD	SFIMSG
1309	032456			25\$:			
1310	032456	112737	000001	MOVB	#1,T15BS1		
1311	032464	012704	034120	MOV	#T15PK2,R4		
1312	032470	012703	000400	MOV	#256.,R3		
1313	032474	112737	000002	MOVB	#2,T15BS0		
1314	032502	105037	034134	CLRB	T15S3		
1315	032506	010337	034132	30\$:			
1316	032512	010465	000000	MOV	R3,T15S2		
1317	032516	004737	016336	MOV	R4,T15S2(R5)		
1318	032522	103405		JSR	PC,CHKTSSR		
1319	032524	010001		BCS	33\$		
1323	032526			MOV	R0,R1		
	032526	104456		ERRHRD	ERRNO,T15SSR,PKTSSR		
	032530	000630				TRAP	C\$ERHRD
	032532	034136				.WORD	408
	032534	012046				.WORD	T15SSR
						.WORD	PKTSSR
1324	032536			33\$:	CKLOOP		
	032536	104406					
1325						TRAP	C\$CLP1

TEST 4: RAM EXERCISER TEST

```

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

```

H10

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 26-4

SEQ 0124

TEST 4: RAM EXERCISER TEST

```

032746 104456
032750 000635
032752 034212
032754 015502
1379 032756 104406
032756 104406
1380 032760 005303
1381 032762 020327 000377
1382 032766 001271
1383
1384 032770
032770
032770 104403
1385
1386 032772
032772
032772 104402
1387
1388
1389
1390
1391
1392
1393
1394
1395 032774 005737 003400
1396 033000 001402
1397 033002 000137 033376
1398 033006 004737 034522
1399 033012 004737 034574
1400 033016 004737 015774
1401 033022 103405
1405 033024 010001
1406 033026
033026 104455
033030 000636
033032 003652
033034 012034
1407 033036
1408 033036 012704 033420
1409 033042 004737 010662
1410 033046 103405
1414 033050 010001
1415 033052
033052 104456
033054 000637
033056 005056
033060 012034
1416 033062
1417 033062 112737 000001 034131
1418 033070 012704 034120
1419 033074 012703 000400
1420 033100 112737 000002 034130
1421 033106 112737 000377 034134
1422 033114 010337 034132
1423 033120 010465 000000
1424 033124 004737 016336

45$: CKLOOP
DEC R3
CMP R3, #255.
BNE 40$
ENDSUB
BGNSUB

10$: JSR PC, T15REST
JSR PC, T15RT2
JSR PC, SFINIT
BCS 20$
MOV R0, R1
ERRDF ERRNO, SFIERR, SFIMSG

20$: MOV #T15PACKET, R4
JSR PC, WRTCHR
BCS 25$
MOV R0, R1
ERRHRD ERRNO, WRTMSG, SFIMSG

25$: MOV #1, T15BS1
MOV #T15PK2, R4
MOV #256., R3
MOV #2, T15BS0
MOV #377, T15S3
MOV R3, T15S2
MOV R4, T5DB(R5)
JSR PC, CHKTSR

;SCOPE LOOP
;DROP RAM ADDRESS POINTER
;AT START YET
;BR, IF MORE RAM TO CHECK
;////////////////////// END SUBTEST ////////////////////////
L10051:
TRAP C$CLP1
;////////////////////// BEGIN SUBTEST ////////////////////////
T4.3:
TRAP C$BSUB

;+
;
;TEST 4, SUBTEST 3
;
;
;THIS SUBTEST WRITES RAM WITH ALL ONES
;THEN WALKS AN ALL ZEROS WORD DOWN THROUGH MEMORY
;
;CHECK RUN MODE
;BR, IF NO SKIP
;SKIP SUBTEST
;RESTORE PACKET FOR WRITE CHARA
;RESTORE PACKET FOR WRT SUB SYS MEM
;DO INITIALIZE ON CONTROLLER
;BR IF INIT WAS OK
;CONTENTS OF TSSR REGISTER
;FATAL ERROR TSSR WAS NOT OK
TRAP C$ERDF
;WORD 414
;WORD SFIERR
;WORD SFIMSG

;SUBROUTINE NEEDS PACKET ADDRESS
;ISSUE WRITE CHARACTERISTICS
;BR, IF COMMAND ISSUED OK
;SAVE CONTENTS OF TSSR
;WRITE CHARACTERISTICS FAILED
TRAP C$ERHRD
;WORD 415
;WORD WRTMSG
;WORD SFIMSG

```

TEST 4: RAM EXERCISER TEST

1425	033130	103405				BCS	33:				;BR, IF NO PROBLEM		
1426	033132	010001				MOV		R0,R1			;SAVE TSSR		
1430	033134					ERRHRD		ERRNO,T15SSR,PKTSSR			;TSSR NOT CORRECT		
	033134	104456										TRAP	C\$ERHRD
	033136	000640										.WORD	416
	033140	034136										.WORD	T15SSR
	033142	012046										.WORD	PKTSSR
1431	033144						33:	CKLOOP			;SCOPE LOOP		
	033144	104406										TRAP	C\$CLP1
1432													
1433													
1434	033146	005203				INC		R3			;NEXT ADDRESS		
1435	033150	020327	010000			CMP		R3,#10000			;END OF RAM MEMORY CHECK		
1436	033154	001357				BNE		30:			;BR, MORE RAM TO GO		
1437	033156	005303				DEC		R3			;SET BACK TO 7777		
1438	033160	112702	000377			MOVB		#377,R2			;SET TO ALL ONES		
1439	033164	112737	000001	034130		MOVB		#1,T15B50			;READ RAM COMMAND		
1440	033172	010337	034132			MOV		R3,T15S2			;ADDRESS TO BE READ TO PACKET DATA		
1441	033176	010465	000000			MOV		R4,TSD8(R5)			;SEND OUT PACKET ADDRESS		
1442	033202	004737	016336			JSR		PC,CHKTSSR			;WAIT FOR SSR TO SET		
1443	033206	103405				BCS		41:			;BR, IF ALL IS WELL		
1444	033210	010001				MOV		R0,R1			;SAVE TSSR		
1448	033212					ERRHRD		ERRNO,T15SSR,PKTSSR			;TSSR NOT CORRECT		
	033212	104456										TRAP	C\$ERHRD
	033214	000641										.WORD	417
	033216	034136										.WORD	T15SSR
	033220	012046										.WORD	PKTSSR
1449	033222						41:	CKLOOP			;SCOPE LOOP		
	033222	104406										TRAP	C\$CLP1
1450	033224	013701	033462			MOV		T15BFR+20,R1			;PICK UP READ DATA		
1451	033230	120102				CMPB		R1,R2			;BOTH SHOULD BE 11111111 BINARY		
1452	033232	001404				BEQ		42:			;BR, IF DATA IS GOOD		
1456	033234					ERRHRD		ERRNO,T15AM3,EXPBREC			;CHARACTERISTICS DATA NOT CORRECT		
	033234	104456										TRAP	C\$ERHRD
	033236	000642										.WORD	418
	033240	034313										.WORD	T15AM3
	033242	015502										.WORD	EXPBREC
1457	033244	012702	000377			MOV		#000377,R2			;SET ALL ONES WORD		
1458	033250	012737	000002	034130		MOV		#2,T15B50			;WRITE RAM COMMAND		
1459	033256	112737	000377	034134		MOVB		#000377,T15S3			;ALL ONES PATTERN		
1460	033264	010465	000000			MOV		R4,TSD8(R5)			;PASS PACKET ADDRESS TO CONTR.		
1461	033270	004737	016336			JSR		PC,CHKTSSR			;WAIT FOR SSR		
1462	033274	103405				BCS		43:			;BR, IF OK (NO ERROR)		
1463	033276	010001				MOV		R0,R1			;SAVE TSSR		
1464	033300					ERRHRD		ERRNO,T15SSR,PKTSSR					

J10

SEQ 0126

TEST 4: RAM EXERCISER TEST

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

TEST 4: RAM EXERCISER TEST

```

1529 034130      000      T15BS0: .BYTE 0      ;BSEL0 AREA
1530 034131      000      T15BS1: .BYTE 0      ;BSEL1 AREA
1531 034132 000000      T15S2:  .WORD 0      ;SEL 2 AREA
1532 034134 000000      T15S3:  .WORD 0      ;DATA AREA
1533
1534
1535
1536
1537
1538      ;*
1539      ;LOCAL TEXT MESSAGES FOR TEST
1540      ;-
1541 034136      127      122      111 T15SR: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted'
1542 034212      127      122      111 T15AM2: .ASCIZ 'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Ones Word Read Back'
1543 034313      127      122      111 T15AM3: .ASCIZ 'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Zeros Word Read Back'
1544 034415      127      122      111 T15AM4: .ASCIZ 'WRITE SUBSYSTEM MEMORY COMMAND Failed On Address Test'
1545 034503      122      101      115 TST15ID: .ASCIZ 'RAM Exerciser'
1546
1547      ;*
1548      ;
1549      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
1550      ;WRITE SUBSYSTEM MEMORY COMMAND
1551      ;
1552      ;-
1553
1554 034522      T15REST:
1555 034522      SAVREG
1556 034526 012701 033420      MOV      #T15PACKET,R1      ;SAVE THE REGISTERS
1557 034532 012721 100204      MOV      #100204,(R1)+      ;START OF THE PACKET
1558 034536 012721 033430      MOV      #T15DATA,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK, IE
1559 034542 005021      CLR      (R1)+      ;ADDRESS OF CHARAISTICS DATA BLOCK
1560 034544 012721 000010      MOV      #8,(R1)+      ;EXTENDED ADDRESS
1561 034550 012721 033442      MOV      #T15BFR,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
1562 034554 005021      CLR      (R1)+      ;ADDRESS OF MESSAGE BUFFER
1563 034556 012721 000400      MOV      #256,(R1)+      ;LENGTH OF MESSAGE BUFFER
1564 034562 005021      CLR      (R1)+
1565 034564 005011      CLR      (R1)
1566 034566 005037 033442      CLR      T15BFR      ;CLEAR 1ST LOC IN MESSAGE BUFFER
1567 034572 000207      RTS      PC      ;RETURN
1568
1569
1570 034574      T15RT2:
1571 034574      SAVREG
1572 034600 012701 034120      MOV      #T15PK2,R1      ;SAVE THE REGISTERS
1573 034604 012721 100206      MOV      #100206,(R1)+      ;START OF THE PACKET
1574 034610 012721 034130      MOV      #T15BF2,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK, IE
1575 034614 005021      CLR      (R1)+      ;ADDRESS OF DATA BLOCK
1576 034616 012721 000006      MOV      #6,(R1)+      ;EXTENDED ADDRESS
1577 034622 005021      CLR      (R1)+      ;SIZE OF DATA BLOCK IN BYTES
1578 034624 005021      CLR      (R1)+
1579 034626 005011      CLR      (R1)
1580 034630 000207      RTS      PC      ;RETURN
1581 034632      ENDTST
      034632
      034632 104401

```

L10047: TRAP C\$ETST

TEST 5: EXTENDED FEATURES SWITCH AND TIMERS A,B

```

1583 .SBTTL TEST 5: EXTENDED FEATURES SWITCH AND TIMERS A,B
1584
1585 ;**
1586 ; TEST DESCRIPTION:
1587 ;
1588 ; This test verifies the Invert Extended Features function
1589 ; can logically invert the Extended features switch and
1590 ; that the internal timers A and B operate correctly.
1591 ;
1592 ; TEST STEPS:
1593 ;
1594 ; REPEAT FOR LOOPCNT
1595 ; BEGIN
1596 ; Do Subtest 1 - Verify Extended Features Switch
1597 ; Do Subtest 2 - Verify Timers A,B
1598 ; END
1599 ;--
1600
1601 034634 BGNTST
1602 034634
1603 034634 T5::
1604 034634 012700 036712 MOV #TST16ID,R0 ;ASCII MESSAGE TO IDENTIFY TEST
1605 034640 004737 016510 JSR PC,TSTSETUP ;DO INITIAL TEST SETUP
1606 034644 012737 000012 002216 MOV #10,,LOOPCNT ;PERFORM 10 ITERATIONS
1607 034652 T16LOOP;
1608
1609 .SBTTL TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST
1610
1611 ;**
1612 ; TEST 5: SUBTEST 1:
1613 ;
1614 ; SUBTEST DESCRIPTION:
1615 ;
1616 ; This subtest verifies that the Invert Sense of Extended features
1617 ; Switch function (Write Subsystem Memory,Write Misc command)
1618 ; operates properly.
1619 ; First the state of the Extended Features switch is read in the
1620 ; message packet supplied by the write characteristics command.
1621 ; Then, the sense of the switch is logically inverted.
1622 ; A Write characteristics command is executed and it is verified
1623 ; that the Extended status register (XST4) is returned when
1624 ; in Extended mode, and not returned if not in extended mode.
1625 ; The subtest also verifies that specifying a Message Buffer
1626 ; address with any of bits 21-19 ,set will cause the command to
1627 ; be rejected.
1628 ;
1629 ; TEST STEPS:
1630 ;
1631 ; BEGIN
1632 ; Write to TSSR register to soft initialize the controller
1633 ; Do WRITE CHARACTERISTICS to check for Extended Features Switch
1634 ; IF Extended Features Hardware Switch CLEAR
1635 ; THEN
1636 ; (* Verify Extended Features switch can be Inverted to SET *)
1637 ; Do Write Subsystem Write Miscellaneous to SET Extended Features.
1638 ; DO a WRITE CHARACTERISTICS with an extended characteristic word
1639 ; Compare the controller ram to the extended characteristic word
1640
1641
1642

```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```

1643      ;      If Data word in controller ram NOT= to word sent Then Print Error
1644      ;      If Message Buffer Data Length NOT= 12. Then Print Error
1645      ;      ELSE
1646      ;      (* Verify Extended Features switch can be Inverted to CLEAR *)
1647      ;      Do Write Subsystem Write Miscellaneous to CLEAR Extended Features.
1648      ;      Do a WRITE CHARACTERISTICS without an extended characteristic word
1649      ;      If Message Buffer Data Length NOT= 10. Then Print Error
1650      ;      END-IF
1651      ;      (* Verify Function Reject when Message Buffer 21-19 are non-zero *)
1652      ;      Write to TSSR register to soft initialize the controller
1653      ;      REPEAT FOR MESSAGE BUFFER ADDRESS bits <21:19> FROM 0 TO 7
1654      ;      DO a WRITE CHARACTERISTICS with a message address bit<21:19> non-zero
1655      ;      If TSSR termination code NOT= Function Reject Then Print Error
1656      ;      END-REPEAT
1657      ;      END
1658      ;      --
1659      034652      BGNSUB                                ;////////// BEGIN SUBTEST //////////
1660      034652      104402                                T5.1:
1661      034652      TRAP      C$BSUB
1662      034654      5$:
1663      ;      Write to TSSR register to soft initialize the controller
1664      034654      004737      015774      JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
1665      034660      103405      BCS      10$      ;BR IF SOFT INIT OKAY
1666      034662      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
1667      034664      010001      ERDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
1668      034664      104455      TRAP      C$ERDF
1669      034666      000764      .WORD      500
1670      034670      003652      .WORD      SFIERR
1671      034672      012034      .WORD      SFIMSG
1672      10$:      Do WRITE CHARACTERISTICS to check for Extended Features Switch
1673      034674      004737      040060      JSR      PC,T16REST      ;RESTORE PACKET DEFAULTS
1674      034700      005037      002222      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
1675      034704      012704      040240      MOV      @T16PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
1676      034710      004737      010662      JSR      PC,WRTCHR      ;DO WRITE CHARACTERISTICS COMMAND
1677      034714      103407      FORCERROR      12$      ;DO FORCE ERROR IF FORCER=1
1678      034730      103407      BCS      15$      ;BR IF CARRY SET (GOOD RETURN)
1679      034732      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
1680      034734      12$:      NEXT,ERRNO
1681      034734      104455      ERDF      ERRNO,T16SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
1682      034736      000765      TRAP      C$ERDF
1683      034740      036762      .WORD      501
1684      034742      012046      .WORD      T16SSR
1685      034744      005237      002222      INC      FATFLG      ;SET FATAL ERROR FLAG
1686      034750      104406      15$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
1687      034750      104406      TRAP      C$CLP1
1688      ;      If Extended Features Hardware Switch Clear then:
1689      ;      (* Verify Extended Features switch can be Inverted to SET *)
1690      ;      REPEAT FOR TEST PATTERNS IN TSTBLK TABLE
1691      034752      012701      040262      MOV      @T16BFR,R1      ;MESSAGE BUFFER ADDRESS
1692      034756      032761      000200      000012      BIT      @X2.EXTF,XST2(R1)      ;EXTENDED FEATURES SWITCH CLEAR?
1693      034764      001402      BEQ      20$      ;BR IF YES
1694      034766      000137      035336      JMP      200$
1695      034772      012703      002764      20$:      MOV      @TSTBLK+10.,R3      ;START OF TEST DATA

```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```

1689          ; Do Write Subsystem Write Miscellaneous to SET Extended Features.
1690
1691 034776 004737 040220      JSR      PC,T16SEXT      ;SETUP PACKET FOR WRITE MISC INVERT
1692 035002 012704 040310      MOV      #T16PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
1693 035006 010465 000000      MOV      R4,TSD8(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
1694 035012 004737 016336      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
1695 035016                      FORCERROR 32$          ;DO FORCE ERROR IF FORCER=1
1696 035032 103407          BCS      40$              ;BR IF CARRY SET (GOOD RETURN)
1697 035034 010001          MOV      R0,R1              ;SAVE CONTENTS OF TSSR
1698 035036                      NEXT.ERRNO
1699 035036 32$:          ERRDF  ERRNO,T162SSR,PKTSSR    ;DEVICE FAT/ SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD    502
                                .WORD    T162SSR
                                .WORD    PKTSSR
                                035036 104455
                                035040 000766
                                035042 037017
                                035044 012046
1700 035046 005237 002222      INC      FATFLG          ;SET FATAL ERROR FLAG
1701 035052 40$:          CK! OOP                      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                035052 104406
1702
1703          ; DO a WRITE CHARACTERISTICS with an extended characteristic word
1704 035054 012737 125252 002312  MOV      #125252,DATA    ;SETUP TEST DATA FOR EXTENDED WORD
1705 035062 012704 040240      MOV      #T16PACKET,R4    ;GET THE ADDRESS OF COMMAND PACKET
1706 035066 012764 000020 000006  MOV      #16.,PKBCNT(R4) ;STORE MESSAGE PACKET SIZE
1707 035074 013737 002312 040260  MOV      DATA,T16DATA+10 ;STORE TEST DATA IN EXTENDED WORD
1708 035102 004737 010662      JSR      PC,WRTCKR      ;DO WRITE CHARACTERISTICS COMMAND
1709 035106                      FORCERROR 42$          ;DO FORCE ERROR IF FORCER=1
1710 035122 103407          BCS      50$              ;BR IF CARRY SET (GOOD RETURN)
1711 035124 010001          MOV      R0,R1              ;SAVE CONTENTS OF TSSR
1712 035126                      NEXT.ERRNO
1713 035126 42$:          ERRDF  ERRNO,T16SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD    503
                                .WORD    T16SSR
                                .WORD    PKTSSR
                                035126 104455
                                035130 000767
                                035132 036762
                                035134 012046
1714 035136 005237 002222      INC      FATFLG          ;SET FATAL ERROR FLAG
1715 035142 50$:          CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                035142 104406
1716          ; If the TSBA Address Register NOT= Expected Then Print Error
1717 035144 016501 000000      MOV      TSBA(R5),R1      ;GET TSBA REGISTER CONTENTS
1718 035150 012702 040262      MOV      #T16BFR,R2      ;START OF THE DATA BUFFER
1719 035154 062702 000020 62$:  ADD      #16.,R2          ;EXPECTED CONTENTS OF TSBA
1720 035160                      FORCERROR 72$,NOTSSR    ;DO FORCE ERROR IF FORCER=1
1721 035170 020102          CMP      R1,R2              ;COMPARE EXPECTED TO RECEIVED
1722 035172 001404          BEQ      80$              ;ERROR IF NOT EQUAL
1723 035174                      NEXT.ERRNO
1724 035174 72$:          ERRHRD  ERRNO,T16TSBA,EXPREC    ;PRINT THE ERROR & EXPD/RECV
                                TRAP      C$ERHRD
                                .WORD    504
                                .WORD    T16TSBA
                                .WORD    EXPREC
                                035174 104456
                                035176 000770
                                035200 037130
                                035202 015474
1725 035204 80$:          CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                035204 104406
1726          ; Compare the controller ram to the extended characteristic word
1727          ; If Data word in controller ram NOT= to word sent Then Print Error
1728 035206 012704 040250      MOV      #T16DATA,R4      ;GET CHARACTERISTIC DATA ADDRESS
1729 035212 004737 011224      JSR      PC,CKRAM2        ;DOES RAM DATA EQUAL DATA SENT?
1730 035216                      FORCERROR 92$          ;DO FORCE ERROR IF FORCER=1

```

B11.

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 27.3

SEQ 0131

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```

1731 035232 103404          BCS      100#          ;BR IF YES
1732 035234          NEXT.ERRNO
1733 035234          92# :  ERRHRD  ERRNO,PKTRAM,RAMERR  ;REPORT THE RAM ERROR(S)
                                TRAP      C1ERRHD
                                .WORD     505
                                .WORD     PKTRAM
                                .WORD     RAMERR
1734 035244          100# :  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C1CLP1
1735 035244          104406          ;
                                ; If Message Buffer Data Length NOT= 12, Then Print Error
1736 035246          012702 040262      MOV      #T16BFR,R2  ;GET MESSAGE BUFFER ADDRESS
1737 035252          016201 000002      MOV      2(R2),R1    ;GET RECV DATA FIELD LENGTH
1738 035256          012702 000014      MOV      #12.,R2     ;GET EXPD DATA FIELD LENGTH
1739 035262          020102          FORCERROR 112#,NOTSSR    ;DO FORCE ERROR IF FORCER=1
1740 035272          001404          CMP      R1,R2          ;COMPARE EXPECTED TO RECEIVED
1741 035274          001404          BEQ      120#          ;ERROR IF NOT EQUAL
1742 035276          112# :  NEXT.ERRNO
1743 035276          112# :  ERRHRD  ERRNO,T16LEN,EXPREC  ;PRINT THE ERROR & EXPD/RECV
                                TRAP      C1ERRHD
                                .WORD     506
                                .WORD     T16LEN
                                .WORD     EXPREC
1744 035306          120# :  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C1CLP1
1745 035306          104406          ;
1746 035310          004737 015774      JSR      PC,SOFINIT  ;WRITE TO TSSR TO SOFT INITIALIZE
1747 035314          103405          BCS      125#          ;BR IF SOFT INIT OKAY
1748 035316          010001          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
1749 035320          104455          ERRDF  ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
                                TRAP      C1ERRDF
                                .WORD     506
                                .WORD     SFIERR
                                .WORD     SFIMSG
1750 035330          125# :  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C1CLP1
1751 035332          000137 035516      JMP      300#          ;
1752 035336          200# :  ; (* Verify Extended Features switch can be Inverted to CLEAR *)
1753 035336          200# :  ; Do Write Subsystem Write Miscellaneous to CLEAR Extended Features.
1754 035336          004737 040220      JSR      PC,T16SEXT  ;SETUP PACKET FOR WRITE MISC INVERT
1755 035342          012704 040310      MOV      #T16PK2,R4  ;GET WRITE SUBSYSTEM COMMAND PACKET
1756 035346          010465 000000      MOV      R4,TSD8(R5)  ;SET THE PACKET ADDRESS TO EXECUTE
1757 035352          004737 016336      JSR      PC,CHKTSSR  ;WAIT FOR SSR TO SET
1758 035356          103407          FORCEERROR 232#          ;DO FORCE ERROR IF FORCER=1
1759 035372          010001          BCS      240#          ;BR IF CARRY SET (GOOD RETURN)
1760 035374          010001          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
1761 035376          232# :  NEXT.ERRNO
1762 035376          232# :  ERRDF  ERRNO,T162SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C1ERRDF
                                .WORD     507
                                .WORD     T162SSR
                                .WORD     PKTSSR
1763 035406          005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
1764 035412          240# :  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C1CLP1
1765 035412          104406          ;
1766 035412          104406          ;
1767 035412          104406          ;

```

C11

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 27-4

SEQ 0132

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```

1768      DO a WRITE CHARACTERISTICS without an extended characteristic word
1769 035414 012704 040240      MOV      #T16PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
1770 035420 012764 000016 000006  MOV      #14.,PKBCNT(R4)      ;STORE MESSAGE PACKET SIZE
1771 035426 004737 010662      JSR      PC,WRCHR      ;DO WRITE CHARACTERISTICS COMMAND
1772 035432      FORCERROR      242#      ;DO FORCE ERROR IF FORCER=1
1773 035446 103407      BCS      250#      ;BR IF CARRY SET (GOOD RETURN)
1774 035450 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
1775 035452      NEXT,ERRNO
1776 035452 242#      ERDF      ERRNO,T16SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C#ERDF
                                .WORD      508
                                .WORD      T16SSR
                                .WORD      PKTSSR
                                035452 104455
                                035454 000774
                                035456 036762
                                035460 012046
1777 035462 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
1778 035466 250#      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
                                035466 104406
1779      IF Message Buffer Data Length NOT= 10. Then Print Error
1780 035470 013701 040264      MOV      T16BFR+2,R1      ;GET RECV DATA FIELD LENGTH
1781 035474 012702 000012      MOV      #10.,R2      ;GET EXPD DATA FIELD LENGTH
1782 035500 020102      CMP      R1,R2      ;COMPARE EXPECTED TO RECEIVED
1783 035502 001404      BEQ      270#      ;ERROR IF NOT EQUAL
1784 035504      NEXT,ERRNO
1785 035504 262#      ERHRD      ERRNO,T16LEN,EXPREC      ;PRINT THE ERROR & EXPD/RECV
                                TRAP      C#ERHRD
                                .WORD      509
                                .WORD      T16LEN
                                .WORD      EXPREC
                                035504 104456
                                035506 000775
                                035510 037232
                                035512 015474
1786 035514 270#      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
                                035514 104406
1787
1788
1789      (* Verify Function Reject when Message Buffer 21-19 are non-zero *)
1790      Write to TSSR register to soft initialize the controller
1791 035516 300#
1792      REPEAT FOR MESSAGE BUFFER ADDRESS bits <21:19> FROM 0 TO 7
1793 035516 012737 000001 002312 320#      MOV      #1,DATA      ;START AT BITS<21:19>=001
1794      DO a WRITE CHARACTERISTICS with a message address bit<21:19> non-zero
1795 035524 325#
1796 035524 012704 040240      MOV      #T16PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
1797 035530 012764 000016 000006  MOV      #14.,PKBCNT(R4)      ;STORE MESSAGE PACKET SIZE
1798 035536 013700 002312      MOV      DATA,R0      ;GET TEST DATA
1799      .REPT      3
1800      ASL      R0      ;SHIFT INTO BITS 21:19
1801      .ENDR
1802 035550 010037 040252      MOV      R0,T16DATA+2      ;STORE BUFFER ADDRESS BITS 21:19
1803 035554 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
1804 035560 004737 016250      JSR      PC,WAITF      ;WAIT FOR SSR
1805 035564      FORCERROR      342#      ;DO FORCE ERROR IF FORCER=1
1806 035600 103407      BCS      350#      ;BR IF CARRY SET (GOOD RETURN)
1807 035602 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
1808 035604      NEXT,ERRNO
1809 035604 342#      ERDF      ERRNO,T16SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C#ERDF
                                .WORD      510
                                .WORD      T16SSR
                                .WORD      PKTSSR
                                035604 104455
                                035606 000776
                                035610 036762
                                035612 012046
1810 035614 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG

```

D11

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 27-5

SEQ 0133

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```

1811 035620      350$: CKLOOP                                ;LOOP ON ERROR, IF FLAG SET
      035620 104406                                         TRAP      C$CLP1
1812
1813      ; If TSSR termination code NOT= Function Reject Then Print Error
1814 035622 016501 000002      MOV      TSSR(R5),R1          ;GET RECV TSSR
1815 035626 010102      MOV      R1,R2                      ;COPY RECV TSSR
1816 035630 042702 000016      BIC      @TERCLS,R2          ;CLEAR TC<2:0> EXPD
1817 035634 052702 000006      BIS      @TSREJ,R2          ;SET EXPD TC<2:0>= FUNCTION REJECT
1818 035640      FORCERROR 352$,NOTSSR          ;000FORCE ERROR IF FORCER=1
1819 035650 020102      CMP      R1,R2                      ;EXPD EQUAL RECV?
1820 035652 001404      BEQ      360$            ;BR IF YES
1821 035654      NEXT.ERRNO
1822 035654      352$: ERRHRD  ERRNO,T16REJ,PKTSSR          ;DEVICE FATAL SSR FAILED TO SET
      035654 104456                                         TRAP      C$ERHRD
      035656 000777                                         .WORD     511
      035660 037344                                         .WORD     T16REJ
      035662 012046                                         .WORD     PKTSSR
1823 035664      360$: CKLOOP                                ;LOOP ON ERROR, IF FLAG SET
      035664 104406                                         TRAP      C$CLP1
1824 035666      FORCEEXIT 370$
1825 035676 005237 002312      INC      DATA
1826 035702 023727 002312 000007      CMP      DATA,@7      ;GET NEXT TST PATTERN
1827 035710 101002      BHI      370$            ;DONE ALL DATA?
1828 035712 000137 035524      JMP      325$            ;BR IF YES
1829      ; DO ANOTHER TEST PATTERN
1830 035716      370$: END-REPEAT
1831 035716      370$: ENDSUB                                ;////////// END SUBTEST //////////
      035716 104403                                         L10054:
      035716                                         TRAP      C$ESUB
1832
1833 035720 005737 002222      TST      FATALG            ;ANY FATAL ERRORS ?
1834 035724 001402      BEQ      460$            ;BRANCH IF NOT
1835 035726 004737 017202      JSR      PC,CKDROP          ;TRY TO DROP THE UNIT
1836 035732      460$:
1837
1838
1839
1840
1841
1842      .SBTTL TEST 5: SUBTEST 2: VERIFY TIMERS A,B
1843
1844
1845      ;**
1846      ; TEST 5: SUBTEST 2:
1847      ;
1848      ; SUBTEST DESCRIPTION:
1849      ;
1850      ; This subtest verifies that timers A,B can be reset
1851      ; and that Timer A is twice the frequency of Timer B.
1852      ; Timer A has a period of 25 microseconds and Timer B
1853      ; as a period of 50 microseconds. The timers are
1854      ; checked at 1, 28, 53, and 78 microseconds.
1855      ;
1856      ; TEST STEPS:
1857      ;
1858      ; Write to TSSR register to soft initialize the controller
1859      ; Do WRITE CHARACTERISTICS to setup a Message Buffer

```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

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

```

F11

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 27-7

SEQ 0135

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

1906 036104 103407      BCS      40$          ;BR IF CARRY SET (GOOD RETURN)
1907 036106 010001      MOV      R0,R1        ;SAVE CONTENTS OF TSSR
1908 036110              NEXT.ERRNO
1909 036110 32$:      ERRDF  ERRNO,T162SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                        TRAP      C$ERDF
                        .WORD      513
                        .WORD      T162SSR
                        .WORD      PKTSSR
                        036110 104455
                        036112 001001
                        036114 037017
                        036116 012046
1910 036120 005237 002222      INC      FATFLG          ;SET FATAL ERROR FLAG
1911 036124 104406      40$:      CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
1912      ;      If Timer A NOT= 0 Then Print Error
1913      ;      If Timer B NOT= 0 Then Print Error
1914 036126 005002      CLR      R2          ;INIT EXPD
1915 036130 042702 000010      BIC      $S2.ATIM,R2        ;TIMER A EXPD=0
1916 036134 042702 000004      BIC      $S2.BTIM,R2        ;TIMER B EXPD=0
1917 036140 012700 040302      MOV      $T16BFSTA,R0        ;GET RECV READ STATUS
1918 036144 016001 000002      MOV      2(R0),R1          ;GET RECV BYTE 2
1919 036150 042701 177763      BIC      $+C<$S2.ATIM!$S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
1920 036154      FORCERROR 72$,NOTSSR ;SSO
1921 036164 020201      CMP      R2,R1          ;EXPD EQUAL RECV?
1922 036166 001404      BEQ      80$          ;BR IF YES
1923 036170              NEXT.ERRNO
1924 036170 72$:      ERRHRD  ERRNO,T16T01,TIMEXP ;REPORT ERROR
                        TRAP      C$ERHRD
                        .WORD      514
                        .WORD      T16T01
                        .WORD      TIMEXP
1925 036200 104406      80$:      CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
1926      ;
1927      ;      Do a Write Control RESET TIMER with 28 microsecond delay
1928 036202 012700 000001      MOV      $MS.RSD,R0        ;RESET TIMER COMMAND
1929 036206 013701 036704      MOV      T16D28,R1        ;28 MICROSECOND DELAY
1930 036212 004737 040172      JSR      PC,T16WMISC        ;SETUP T16PK2 COMMAND PACKET
1931 036216 012704 040310      MOV      $T16PK2,R4        ;GET WRITE SUBSYSTEM COMMAND PACKET
1932 036222 010465 000000      MOV      R4,TSD8(R5)        ;SET THE PACKET ADDRESS TO EXECUTE
1933 036226 004737 016336      JSR      PC,CHKTSSR        ;WAIT FOR SSR TO SET
1934 036232      FORCERROR 112$          ;SSO FORCE ERROR IF FORCER=1
1935 036246 103407      BCS      120$          ;BR IF CARRY SET (GOOD RETURN)
1936 036250 010001      MOV      R0,R1        ;SAVE CONTENTS OF TSSR
1937 036252      NEXT.ERRNO
1938 036252 112$:      ERRDF  ERRNO,T162SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                        TRAP      C$ERDF
                        .WORD      515
                        .WORD      T162SSR
                        .WORD      PKTSSR
                        036252 104455
                        036254 001003
                        036256 037017
                        036260 012046
1939 036262 005237 002222      INC      FATFLG          ;SET FATAL ERROR FLAG
1940 036266 104406      120$:      CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
1941      ;      If Timer A NOT= 1 Then Print Error
1942      ;      If Timer B NOT= 1 Then Print Error
1943 036270 005002      CLR      R2          ;INIT EXPD
1944 036272 052702 000010      BIS      $S2.ATIM,R2        ;TIMER A EXPD=1
1945 036276 052702 000004      BIS      $S2.BTIM,R2        ;TIMER B EXPD=1
1946 036302 012700 040302      MOV      $T16BFSTA,R0        ;GET RECV READ STATUS
1947 036306 016001 000002      MOV      2(R0),R1          ;GET RECV BYTE 2

```

G11

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 27-8

SEQ 0136

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

1948 036312 042701 177763      BIC      *C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
1949 036316                      FORCERROR 172$,NOTSSR ;000
1950 036326 020201              CMP      R2,R1 ;EXPD EQUAL RECV?
1951 036330 001404              BEQ      180$ ;BR IF YES
1952 036332                      NEXT,ERRNO
1953 036332 172$: ERRHRD ERRNO,T16T28,TIMEXP ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD     516
                                .WORD     T16T28
                                .WORD     TIMEXP
                                TRAP      C$CLP1
                                036332 104456
                                036334 001004
                                036336 037560
                                036340 015552
1954 036342 180$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                036342 104406
1955
1956 ; Do a Write Control RESET TIMER with 53 microsecond delay
1957 036344 012700 000001      MOV      *MS.RSD,R0 ;RESET TIMER COMMAND
1958 036350 013701 036706      MOV      T16D53,R1 ;53 MICROSECOND DELAY
1959 036354 004737 040172      JSR      PC,T16WMISC ;SETUP T16PK2 COMMAND PACKET
1960 036360 012704 040310      MOV      *T16PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
1961 036364 010465 000000      MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
1962 036370 004737 016336      JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
1963 036374                      FORCERROR 212$ ;000FORCE ERROR IF FORCER=1
1964 036410 103407              BCS      220$ ;BR IF CARRY SET (GOOD RETURN)
1965 036412 010001              MOV      R0,R1 ;SAVE CONTENTS OF TSSR
1966 036414                      NEXT,ERRNO
1967 036414 212$: ERDRF ERRNO,T162SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     517
                                .WORD     T162SSR
                                .WORD     PKTSSR
                                036414 104455
                                036416 001005
                                036420 037017
                                036422 012046
1968 036424 005237 002222      INC      FATFLG ;SET FATAL ERROR FLAG
1969 036430 220$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                036430 104406
1970 ; If Timer A NOT= 0 Then Print Error
1971 ; If Timer B NOT= 1 Then Print Error
1972 036432 005002              CLR      R2 ;INIT EXPD
1973 036434 042702 000010      BIC      *S2.ATIM,R2 ;TIMER A EXPD=0
1974 036440 052702 000004      BIS      *S2.BTIM,R2 ;TIMER B EXPD=1
1975 036444 012700 040302      MOV      *T16BFSTA,R0 ;GET RECV READ STATUS
1976 036450 016001 000002      MOV      2(R0),R1 ;GET RECV BYTE 2
1977 036454 042701 177763      BIC      *C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
1978 036460                      FORCERROR 272$,NOTSSR ;000
1979 036470 020201              CMP      R2,R1 ;EXPD EQUAL RECV?
1980 036472 001404              BEQ      280$ ;BR IF YES
1981 036474                      NEXT,ERRNO
1982 036474 272$: ERRHRD ERRNO,T16T53,TIMEXP ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD     518
                                .WORD     T16T53
                                .WORD     TIMEXP
                                TRAP      C$CLP1
                                036474 104456
                                036476 001006
                                036500 037660
                                036502 015552
1983 036504 280$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                036504 104406
1984 ; Do a Write Control RESET TIMER with 78 microsecond delay
1985 036506 012700 000001      MOV      *MS.RSD,R0 ;RESET TIMER COMMAND
1986 036512 013701 036710      MOV      T16D78,R1 ;78 MICROSECOND DELAY
1987 036516 004737 040172      JSR      PC,T16WMISC ;SETUP T16PK2 COMMAND PACKET
1988 036522 012704 040310      MOV      *T16PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
1989 036526 010465 000000      MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE

```

H11

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 27-9

SEQ 0137

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

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

```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

2033 036710 000142      T16D78:      .WORD 142      ,78 MICROSECOND
2034                      ;+
2035                      ;LOCAL TEXT MESSAGES FOR TEST
2036                      ;-
2037
2038 036712      105      170      164 TST16ID:      .ASCIIZ 'Extended Features Switch and Timers A,B'
2039 036762      127      122      111 T16SSR: .ASCIIZ 'WRITE CHARACTERISTICS Failed'
2040 037017      127      122      111 T162SSR: .ASCIIZ 'WRITE SUBSYSTEM (Write Misc) Failed'
2041 037063      127      122      111 T163SSR: .ASCIIZ 'WRITE SUBSYSTEM (Read Status) Failed'
2042 037130      102      165      163 T16TSBA: .ASCIIZ 'Bus Address Register (TSBA) Incorrect after Write Characteristics'
2043 037232      104      141      164 T16LEN: .ASCIIZ 'Data Field Length in Message Buffer Incorrect after Write Characteristics'
2044 037344      124      123      123 T16REJ: .ASCIIZ 'TSSR Function Reject Not Returned When Non-Existent Buffer Address Specifie
d'
2045 037461      124      151      155 T16T01: .ASCIIZ 'Timer A,B Incorrect after Reset Timer with 1 microsecond Delay'
2046 037560      124      151      155 T16T28: .ASCIIZ 'Timer A,B Incorrect after Reset Timer with 28 microsecond Delay'
2047 037660      124      151      155 T16T53: .ASCIIZ 'Timer A,B Incorrect after Reset Timer with 53 microsecond Delay'
2048 037760      124      151      155 T16T78: .ASCIIZ 'Timer A,B Incorrect after Reset Timer with 78 microsecond Delay'
2049                      .EVEN
2050
2051                      ;+
2052                      ; SET DEFAULT PACKET
2053                      ;-
2054 040060      T16REST:
2055 040060      012700 040240      MOV      #T16PACKET,R0      ;PACKET ADDRESS
2056 040064      012720 100004      MOV      #100004,(R0)+      ;WRITE CHARACTERISTICS WITH ACK
2057 040070      012720 040250      MOV      #T16DATA,(R0)+      ;ADDRESS OF CHAR DATA BLOCK
2058 040074      005020      CLR      (R0)+      ;EXTENDED ADDRESS
2059 040076      012720 000012      MOV      #10.,(R0)+      ;SIZE OF MESSAGE PACKET
2060 040102      012720 040262      MOV      #T16BFR,(R0)+      ;MESSAGE BUFFER ADDRESS
2061 040106      005020      CLR      (R0)+      ;CLEAR EXTENDED BUFFER ADDRESS
2062 040110      012720 000024      MOV      #20.,(R0)+      ;LENGTH OF MESSAGE BUFFER
2063 040114      005020      CLR      (R0)+      ;CLEAR ESS,ENB,EAI,ERI
2064 040116      005010      CLR      (R0)      ;CLEAR EXTENDED FEATURES WORD
2065 040120      005037 040262      CLR      T16BFR      ;CLEAR 1ST LOCATION IN MESSAGE BUFFER
2066 040124      000207      RTS      PC
2067
2068                      ;+
2069                      ; CLEAR MESSAGE BUFFER
2070                      ;-
2071 040126      T16CLRBUF:
2072 040126      SAVREG      ;SAVE R1-R5 UNTIL NEXT RETURN
2073 040132      012701 040262      MOV      #T16BFR,R1      ;GET MESSAGE BUFFER ADDRESS
2074 040136      012702 000026      MOV      #T16BEND-T16BFR,R2      ;SIZE OF MESSAGE BUFFER IN BYTES
2075 040142      105021      10$: CLR      (R1)+      ;CLEAR A BYTE
2076 040144      005302      DEC      R2      ;DONE?
2077 040146      003375      BGT      10$      ;BR IF NO
2078 040150      000207      RTS      PC      ;RETURN
2079
2080                      ;+
2081                      ; SETUP T16PK2 PACKET FOR READ STATUS
2082                      ;-
2083 040152      T16SRD:
2084 040152      004737 040126      JSR      PC,T16CLRBUF      ;CLEAR MESSAGE BUFFER
2085 040156      012700 040320      MOV      #T160T2,R0      ;WRITE SUBSYSTEM DATA BUFFER
2086 040162      112720 000005      MOV      #PW,RDSTATUS,(R0)+      ;STORE READ STATUS COMMAND IN BSEL0
2087 040166      105010      CLR      (R0)      ;CLEAR BSEL1
2088 040170      000207      RTS      PC      ;RETURN
2089

```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

2090
2091
2092      ;+
2093      ; SETUP T16PK2 PACKET FOR WRITE MISC.
2094      ;
2095      ; INPUT:
2096      ;      R0      CONTAINS WRITE MISC FUNCTION CODE (BSEL1)
2097      ;      R1      CONTAINS DELAY (TIMES 800 NS) FOR BSEL2
2098      ;-
2098 040172 T16WMISC:
2099 040172      SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
2100 040176 004737 040126      JSR      PC,T16CLRBUF      ;CLEAR MESSAGE BUFFER
2101 040202 012702 040320      MOV      #T16DT2,R2      ;WRITE SUBSYSTEM DATA BUFFER
2102 040206 112722 000010      MOVB     #PW.WMISC,(R2)+   ;STORE WRITE MISCELLANEOUS IN BSEL0
2103 040212 110022              MOVB     R0,(R2)+         ;STORE WRITE MISC CODE IN BSEL1
2104 040214 110112              MOVB     R1,(R2)          ;STORE DELAY (RESET TIMER) IN BSEL2
2105 040216 000207              RTS      PC               ;RETURN
2106
2107      ;+
2108      ; SETUP T16PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
2109      ;-
2109 040220 T16SEXT:
2110 040220 012700 040320      MOV      #T16DT2,R0      ;WRITE SUBSYSTEM DATA BUFFER
2111 040224 112720 000010      MOVB     #PW.WMISC,(R0)+   ;STORE WRITE MISCELLANEOUS IN BSEL0
2112 040230 112710 000200      MOVB     #MS.EXT,(R0)      ;STORE INVERT EXTENDED FEATURES IN BSEL1
2113 040234 000207              RTS      PC               ;RETURN
2114
2115
2116
2117
2119      040240      .=<.+10>E177770
2121
2122      ;
2123      ;WRITE CHARACTERISTICS COMMAND PACKET
2124      ;
2124 040240 T16PACKET:
2125 040240 100004      .WORD      100004      ;COMMAND PACKET FOR TEST
2126 040242 040250      .WORD      T16DATA     ;WRITE CHARACTERISTICS COMMAND, WITH ACK
2127 040244 000000      .WORD      0           ;ADDRESS OF CHARACTERISTICS BLOCK
2128 040246 000012      .WORD      10.         ;MESSAGE PACKET SIZE
2129
2130 040250 T16DATA:
2131 040250 040262      .WORD      T16BFR       ;CHARACTERISTICS DATA BLOCK
2132 040252 000000      .WORD      0           ;ADDRESS OF MESSAGE BUFFER
2133 040254 000024      .WORD      20.         ;LENGTH OF MESSAGE BUFFER
2134 040256 000000      .WORD      0           ;ESS,ENB,EAI,ERI
2135 040260 000000      .WORD      0           ;EXTENDED FEATURES WORD
2136
2137
2138      ;MESSAGE BUFFER
2139
2140 040262 T16BFR:
2141 040262 000000      .WORD      0           ;BEGIN MESSAGE BUFFER
2142 040264 000000      .WORD      0           ;MESSAGE TYPE
2143 040266 000000      .WORD      0           ;DATA FIELD LENGTH
2144 040270 000000      .WORD      0           ;RBPGR
2145 040272 000000      .WORD      0           ;XST0
2146 040274 000000      .WORD      0           ;XST1
2147 040276 000000      .WORD      0           ;XST2
2148 040300 000000      .WORD      0           ;XST3
2148 040300 000000      .WORD      0           ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM

```

K11

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 27-12

SEQ 0140

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

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

TEST 6: FIFO EXERCISER

```

2172 .SBTTL TEST 6: FIFO EXERCISER
2173 ;++
2174 ; TEST DESCRIPTION:
2175 ;
2176 ; This test uses the Write Subsystem Memory command to
2177 ; verify the controller's FIFO and associated status and
2178 ; control logic.
2179 ;
2180 ; TEST STEPS:
2181 ;
2182 ; REPEAT FOR LOOPCNT
2183 ; BEGIN
2184 ; Do Subtest 1 - FIFO Initialize status test
2185 ; Do Subtest 2 - FIFO Write Single Byte test
2186 ; Do Subtest 3 - FIFO Write Multiple Bytes test
2187 ; Do Subtest 4 - FIFO Verify ILW Status test
2188 ; Do Subtest 5 - FIFO Input Ready test
2189 ; Do Subtest 6 - FIFO Verify Reset FIFO test
2190 ; END
2191 ;--
2192
2193
2194 040426 BGNTST
2195 040426
2199 040426 012700 046656 MOV #TST17ID,R0 ;ASCII MESSAGE TO IDENTIFY TEST
2200 040432 004737 016510 JSR PC,TSTSETUP ;DO INITIAL TEST SETUP
2201 040436 012737 000012 002216 MOV #10.,LOOPCNT ;PERFORM 10 ITERATIONS
2202 040444 004737 017274 JSR PC,KTOFF ;SHUT OFF MEMORY MANAGEMENT
2203 040450 005037 003134 CLR KTENABLE ;REALLY SHUT DOWN KT-11
2204 040454
2205 T17LOOP:
2206
2207
2208 .SBTTL TEST 6: SUBTEST 1: FIFO INITIALIZE STATUS TEST
2209 ;++
2210 ; TEST 6: SUBTEST 1:
2211 ;
2212 ; SUBTEST DESCRIPTION:
2213 ;
2214 ; This test verifies, by using the Read Status select code,
2215 ; that the FIFO status is in the correct initial state after
2216 ; the controller is initialized (Input Ready TRUE,
2217 ; Output Ready and Data In Miss FALSE). These status
2218 ; signals are checked by the controller's self-test
2219 ; sequence, so this subtest is actually more of a partial
2220 ; check of the Read Status function than the FIFO status.
2221 ;
2222 ; TEST STEPS:
2223 ;
2224 ; BEGIN
2225 ; Write to ISSR to soft initialize
2226 ; Do a WRITE CHARACTERISTICS to setup a message buffer
2227 ; Do a WRITE SUBSYSTEM Read Status
2228 ; If Input Ready NOT=1 Then Print Error
2229 ; If Output Ready NOT=0 Then Print Error
2230 ; If Data In Miss NOT=0 Then Print Error
2231 ; END

```

TEST 6: SUBTEST 1: FIFO INITIALIZE STATUS TEST

```

2232      ;--
2233 040454      BGNSUB                      ;////////// BEGIN SUBTEST //////////
      040454      T6.1:
      040454 104402      TRAP      C$BSUB
2234
2235      ;      Write to TSSR register to soft initialize the controller
2236 040456      5$:
2237 040456 004737 015774      JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
2238 040462 103405      BCS      10$      ;BR IF SOFT INIT OKAY
2239 040464 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2240 040466      ERRDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
      040466 104455      TRAP      C$ERDF
      040470 001130      .WORD      600
      040472 003652      .WORD      SFIERR
      040474 012034      .WORD      SFIMSG
2241      ;      Do a WRITE CHARACTERISTICS to setup a message buffer
2242 040476 005037 002222      10$:      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
2243 040502 012704 050250      MOV      @T17PK2,R4      ;GET THE ADDRESS OF COMMAND PACKET
2244 040506 004737 010662      JSR      PC,WRTCHR      ;DO WRITE CHARACTERISTICS COMMAND
2245 040512      FORCERROR      42$      ;DO FORCE ERROR IF FORCER=1
2246 040526 103407      BCS      50$      ;BR IF CARRY SET (GOOD RETURN)
2247 040530 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2248 040532      NEXT,ERRNO
2249 040532      42$:      ERRDF      ERRNO,T17SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      040532 104455      TRAP      C$ERDF
      040534 001131      .WORD      601
      040536 046675      .WORD      T17SSR
      040540 012046      .WORD      PKTSSR
2250 040542 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
2251 040546      50$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      040546 104406      TRAP      C$CLP1
2252
2253      ;      Do a Write Subsystem READ STATUS
2254 040550 004737 050034      JSR      PC,T17SRD      ;SETUP PACKET FOR READ STATUS
2255 040554 012704 050420      MOV      @T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
2256 040560 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
2257 040564 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
2258 040570      FORCERROR      62$      ;DO FORCE ERROR IF FORCER=1
2259 040604 103407      BCS      70$      ;BR IF CARRY SET (GOOD RETURN)
2260 040606 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2261 040610      NEXT,ERRNO
2262 040610      62$:      ERRDF      ERRNO,T173SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      040610 104455      TRAP      C$ERDF
      040612 001132      .WORD      602
      040614 046776      .WORD      T173SSR
      040616 012046      .WORD      PKTSSR
2263 040620 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
2264 040624      70$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      040624 104406      TRAP      C$CLP1
2265      ;      Set WORDS 0-7 of expd message buffer = to recv since not testing
2266 040626 004737 050216      JSR      PC,T17SETEXP      ;SET WORDS 0-7 EXPD-RECV
2267 040632 012701 046452      MOV      @T17EXSTA,R1      ;GET EXPECTED READ STATUS
2268 040636 012702 050312      MOV      @T17BFSTA,R2      ;GET RECV READ STATUS
2269 040642 012221      MOV      (R2)+,(R1)+      ;SET EXPD WORD #8 = RECV TEMP
2270 040644 011211      MOV      (R2),(R1)      ;SET EXPD WORD #9 = RECV TEMP
2271 040646 052711 000020      BIS      #S2.INRDY,(R1)      ;SET EXP INPUT READY= TRUE
2272 040652 042711 000040      BIC      #S2.OUTRDY,(R1)      ;SET EXP OUTPUT READY= FALSE

```

TEST 6: SUBTEST 1: FIFO INITIALIZE STATUS TEST

```

2273 040656 042711 000200      BIC      #S2.DIM,(R1)      ;SET EXP DATA IN MISS = FALSE
2274                          ; If Input Ready NOT=1 then Print Error
2275                          ; If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2276 040662 005000      CLR      R0      ;HIGH RECV ADDRESS FOR CKMSG2
2277 040664 012701 050272      MOV      #T17BFR,R1      ;LOW RECV ADDRESS FOR CKMSG2
2278 040670 012702 046432      MOV      #T17EXP,R2      ;EXPD ADDRESS
2279 040674 012703 000024      MOV      #20.,R3      ;NUMBER OF BYTES TO COMPARE
2280 040700 004737 011500      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
2281 040704      FORCERROR      82$,NOTSSR      ;88D
2282 040714 103404      BCS      90$      ;BR IF YES
2283 040716      NEXT,ERRNO
2284 040716 82$:      ERRHRD  ERRNO,T171CMP,MSGSTAT      ;REPORT ERROR
                        040716 104456      TRAP      C$ERHRD
                        040720 001133      .WORD      603
                        040722 047215      .WORD      T171CMP
                        040724 012350      .WORD      MSGSTAT
2285 040726 90$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                        040726 104406      TRAP      C$CLP1
2286
2287 040730      ENDSUR      ;////////// END SUBTEST //////////
                        040730      L10057:
                        040730 104403      TRAP      C$ESUB
2288
2289 040732 005737 002222      TST      FATFLG      ;ANY FATAL ERRORS ?
2290 040736 001402      BEQ      160$      ;BRANCH IF NOT
2291 040740 004737 017202      JSR      PC,CKDROP      ;TRY TO DROP THE UNIT
2292 040744 160$:

```

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

```

2293
2294
2295
2296
2297 ;**
2298 ; TEST 6: SUBTEST 2:
2299 ;
2300 ; SUBTEST DESCRIPTION:
2301 ;
2302 ; This subtest verifies the ability of the FIFO to correctly
2303 ; pass a single data byte from input to output. For each
2304 ; of 256 data values (0-377 octal) the following is done:
2305 ; 1. Initial FIFO status is checked
2306 ; 2. The Write FIFO function, specifying a count of
2307 ; one byte to be written is executed.
2308 ; 3. Read Status is executed and FIFO status is checked.
2309 ; 4. Read FIFO is executed and the data and final status
2310 ; is checked.
2311 ;
2312 ; TEST STEPS:
2313 ;
2314 ; BEGIN
2315 ; Write to TSSR to soft initialize
2316 ; Do a WRITE CHARACTERISTICS to setup a message buffer
2317 ; Do a Write Subsystem READ STATUS
2318 ; If Input Ready NOT=1 Then Print Error
2319 ; If Output Ready NOT=0 Then Print Error
2320 ; If Data In Miss NOT=0 Then Print Error
2321 ;
2322 ; REPEAT FOR DATA FROM 0 TO 377 OCTAL
2323 ; BEGIN

```


TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

PC	PC+1	PC+2	PC+3	PC+4	PC+5	PC+6	PC+7	PC+8	PC+9	PC+10	PC+11	PC+12	PC+13	PC+14	PC+15	PC+16	PC+17	PC+18	PC+19	PC+20	PC+21	PC+22	PC+23	PC+24	PC+25	PC+26	PC+27	PC+28	PC+29	PC+30	PC+31	PC+32	PC+33	PC+34	PC+35	PC+36	PC+37	PC+38	PC+39	PC+40	PC+41	PC+42	PC+43	PC+44	PC+45	PC+46	PC+47	PC+48	PC+49	PC+50	PC+51	PC+52	PC+53	PC+54	PC+55	PC+56	PC+57	PC+58	PC+59	PC+60	PC+61	PC+62	PC+63	PC+64	PC+65	PC+66	PC+67	PC+68	PC+69	PC+70	PC+71	PC+72	PC+73	PC+74	PC+75	PC+76	PC+77	PC+78	PC+79	PC+80	PC+81	PC+82	PC+83	PC+84	PC+85	PC+86	PC+87	PC+88	PC+89	PC+90	PC+91	PC+92	PC+93	PC+94	PC+95	PC+96	PC+97	PC+98	PC+99	PC+100	PC+101	PC+102	PC+103	PC+104	PC+105	PC+106	PC+107	PC+108	PC+109	PC+110	PC+111	PC+112	PC+113	PC+114	PC+115	PC+116	PC+117	PC+118	PC+119	PC+120	PC+121	PC+122	PC+123	PC+124	PC+125	PC+126	PC+127	PC+128	PC+129	PC+130	PC+131	PC+132	PC+133	PC+134	PC+135	PC+136	PC+137	PC+138	PC+139	PC+140	PC+141	PC+142	PC+143	PC+144	PC+145	PC+146	PC+147	PC+148	PC+149	PC+150	PC+151	PC+152	PC+153	PC+154	PC+155	PC+156	PC+157	PC+158	PC+159	PC+160	PC+161	PC+162	PC+163	PC+164	PC+165	PC+166	PC+167	PC+168	PC+169	PC+170	PC+171	PC+172	PC+173	PC+174	PC+175	PC+176	PC+177	PC+178	PC+179	PC+180	PC+181	PC+182	PC+183	PC+184	PC+185	PC+186	PC+187	PC+188	PC+189	PC+190	PC+191	PC+192	PC+193	PC+194	PC+195	PC+196	PC+197	PC+198	PC+199	PC+200	PC+201	PC+202	PC+203	PC+204	PC+205	PC+206	PC+207	PC+208	PC+209	PC+210	PC+211	PC+212	PC+213	PC+214	PC+215	PC+216	PC+217	PC+218	PC+219	PC+220	PC+221	PC+222	PC+223	PC+224	PC+225	PC+226	PC+227	PC+228	PC+229	PC+230	PC+231	PC+232	PC+233	PC+234	PC+235	PC+236	PC+237	PC+238	PC+239	PC+240	PC+241	PC+242	PC+243	PC+244	PC+245	PC+246	PC+247	PC+248	PC+249	PC+250	PC+251	PC+252	PC+253	PC+254	PC+255	PC+256	PC+257	PC+258	PC+259	PC+260	PC+261	PC+262	PC+263	PC+264	PC+265	PC+266	PC+267	PC+268	PC+269	PC+270	PC+271	PC+272	PC+273	PC+274	PC+275	PC+276	PC+277	PC+278	PC+279	PC+280	PC+281	PC+282	PC+283	PC+284	PC+285	PC+286	PC+287	PC+288	PC+289	PC+290	PC+291	PC+292	PC+293	PC+294	PC+295	PC+296	PC+297	PC+298	PC+299	PC+300	PC+301	PC+302	PC+303	PC+304	PC+305	PC+306	PC+307	PC+308	PC+309	PC+310	PC+311	PC+312	PC+313	PC+314	PC+315	PC+316	PC+317	PC+318	PC+319	PC+320	PC+321	PC+322	PC+323	PC+324	PC+325	PC+326	PC+327	PC+328	PC+329	PC+330	PC+331	PC+332	PC+333	PC+334	PC+335	PC+336	PC+337	PC+338	PC+339	PC+340	PC+341	PC+342	PC+343	PC+344	PC+345	PC+346	PC+347	PC+348	PC+349	PC+350	PC+351	PC+352	PC+353	PC+354	PC+355	PC+356	PC+357	PC+358	PC+359	PC+360	PC+361	PC+362	PC+363	PC+364	PC+365	PC+366	PC+367	PC+368	PC+369	PC+370	PC+371	PC+372	PC+373	PC+374	PC+375	PC+376	PC+377	PC+378	PC+379	PC+380	PC+3
----	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	------

D12

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 28-5

SEQ 0146

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

2411	041324	012704	050420	MOV	#T17PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
2412	041330	010465	000000	MOV	R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
2413	041334	004737	016336	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET	
2414	041340			FORCERROR	107#	;SSDFORCE ERROR IF FORCER=1	
2415	041354	103407		BCS	110#	;BR IF CARRY SET (GOOD RETURN)	
2416	041356	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR	
2417	041360			NEXT,ERRNO			
2418	041360			107#:	ERRDF ERRNO,T175SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	
	041360	104455				TRAP	C#ERDF
	041362	001140				.WORD	608
	041364	047.06				.WORD	T175SSR
	041366	012046				.WORD	PKTSSR
2419	041370	005237	002222	INC	FATFLG	;SET FATAL ERROR FLAG	
2420	041374			110#:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	041374	104406				TRAP	C#CLP1
2421							
2422					Do a Write Subsystem READ STATUS		
2423	041376	004737	050034	JSR	PC,T17SRD	;SETUP PACKET FOR READ STATUS	
2424	041402	012704	050420	MOV	#T17PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
2425	041406	010465	000000	MOV	R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
2426	041412	004737	016336	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET	
2427	041416			FORCERROR	112#	;SSDFORCE ERROR IF FORCER=1	
2428	041432	103407		BCS	120#	;BR IF CARRY SET (GOOD RETURN)	
2429	041434	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR	
2430	041436			NEXT,ERRNO			
2431	041436			112#:	ERRDF ERRNO,T173SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	
	041436	104455				TRAP	C#ERDF
	041440	001141				.WORD	609
	041442	046776				.WORD	T173SSR
	041444	012046				.WORD	PKTSSR
2432	041446	005237	002222	INC	FATFLG	;SET FATAL ERROR FLAG	
2433	041452			120#:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	041452	104406				TRAP	C#CLP1
2434					Set WORDS 0-7 of expd message buffer = to recv since not testing		
2435	041454	004737	050216	JSR	PC,T17SETEXP	;SET WORDS 0-7 EXPD=RECV	
2436	041460	012701	046452	MOV	#T17EXSTA,R1	;GET EXPECTED READ STATUS	
2437	041464	012702	050312	MOV	#T17BFSTA,R2	;GET RECV READ STATUS	
2438	041470	012221		MOV	(R2), (R1)	;SET EXPD WORD #8 = RECV TEMP	
2439	041472	011211		MOV	(R2), (R1)	;SET EXPD WORD #9 = RECV TEMP	
2440	041474	052711	000020	BIS	#S2.INRDY,(R1)	;SET EXP INPUT READY= 1	
2441	041500	052711	000040	BIS	#S2.OTRDY,(R1)	;SET EXP OUTPUT READY= 1	
2442	041504	042711	000200	BIC	#S2.DIM,(R1)	;SET EXP DATA IN MISS = 0	
2443					If Input Ready NOT=1 then Print Error		
2444					If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error		
2445	041510	005000		CLR	R0	;HIGH RECV ADDRESS FOR CKMSG2	
2446	041512	012701	050272	MOV	#T17BFR,R1	;LOW RECV ADDRESS FOR CKMSG2	
2447	041516	012702	046432	MOV	#T17EXP,R2	;EXPD ADDRESS	
2448	041522	012703	000024	MOV	#20,R3	;NUMBER OF BYTES TO COMPARE	
2449	041526	004737	011500	JSR	PC,CKMSG2	;EXPD EQUAL RECV?	
2450	041532			FORCERROR	132#,NOTSSR	;SSD	
2451	041542	103404		BCS	140#	;BR IF YES	
2452	041544			NEXT,ERRNO			
2453	041544			132#:	ERRHRD ERRNO,T173CMP,MSGSTAT	;REPORT ERROR	
	041544	104456				TRAP	C#ERHRD
	041546	001142				.WORD	610
	041550	047373				.WORD	T173CMP
	041552	012350				.WORD	MSGSTAT

E12

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 28-6

SEQ 0147

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```

2454 041554      140$: CKLOOP                                ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
2455 041554 104406
2456 ; Do Write Subsystem READ FIFO with byte count equal to 1
2457 041556 012700 000001      MOV      #1,R0                ;SET READ BYTE COUNT
2458 041562 004737 050156      JSR      PC,T17RFIF           ;SETUP T17PK2 FOR READ FIFO
2459 041566 012704 050420      MOV      #T17PK2,R4          ;GET WRITE SUBSYSTEM COMMAND PACKET
2460 041572 010465 000000      MOV      R4,TSD8(R5)         ;SET THE PACKET ADDRESS TO EXECUTE
2461 041576 004737 016336      JSR      PC,CHKTSSR          ;WAIT FOR SSR TO SET
2462 041602      FORCERROR      142$                        ;GOOD FORCE ERROR IF FORCER=1
2463 041616 103407      BCS      150$                      ;BR IF CARRY SET (GOOD RETURN)
2464 041620 010001      MOV      R0,R1                      ;SAVE CONTENTS OF TSSR
2465 041622      NEXT,ERRNO
2466 041622      142$: ERDF      ERRNO,T176SSR,PKTSSR        ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      611
                                .WORD      T176SSR
                                .WORD      PKTSSR
2467 041632 005237 002222      INC      FATFLG              ;SET FATAL ERROR FLAG
2468 041636      150$: CKLOOP                                ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
2469 041636 104406
2470 ; Set WORDS 0-7 of expd message buffer * to rcv since not testing
2471 041640 004737 050216      JSR      PC,T17SETEXP         ;SET WORDS 0-7 EXPD=RCV
2472 041644 012701 046452      MOV      #T17EXSTA,R1        ;GET EXPECTED READ STATUS
2473 041650 012702 050312      MOV      #T17BFSTA,R2        ;GET RCV READ STATUS
2474 041654 013721 002312      MOV      DATA,(R1)+         ;SET EXPD WORD #8 = COUNT DATA
2475 041660 011211      MOV      (R2),(R1)                  ;SET EXPD WORD #9 = RCV (NOT TESTING)
2476 ; If Data read from FIFO NOT= to Data sent Then Print Error
2477 ; The data is in WORD #8 of the message buffer
2478 041662 005000      CLR      R0                          ;HIGH RCV ADDRESS FOR CKMSG2
2479 041664 012701 050272      MOV      #T17BFR,R1          ;LOW RCV ADDRESS FOR CKMSG2
2480 041670 012702 046432      MOV      #T17EXP,R2          ;EXPD ADDRESS
2481 041674 012703 000022      MOV      #18.,R3             ;NUMBER OF BYTES TO COMPARE
2482 041700 004737 011500      JSR      PC,CKMSG2           ;EXPD EQUAL RCV?
2483 041704      FORCERROR      152$,NOTSSR                ;GOOD
2484 041714 103404      BCS      160$                      ;BR IF YES
2485 041716      152$: ERHRD      ERRNO,T172CMP,MSGSUB      ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD      612
                                .WORD      T172CMP
                                .WORD      MSGSUB
2486 041726      160$: CKLOOP                                ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
2487 041726 104406
2488 ; Do a Write Subsystem READ STATUS
2489 041730 004737 050034      JSR      PC,T17SRD           ;SETUP PACKET FOR READ STATUS
2490 041734 012704 050420      MOV      #T17PK2,R4          ;GET WRITE SUBSYSTEM COMMAND PACKET
2491 041740 010465 000000      MOV      R4,TSD8(R5)         ;SET THE PACKET ADDRESS TO EXECUTE
2492 041744 004737 016336      JSR      PC,CHKTSSR          ;WAIT FOR SSR TO SET
2493 041750      FORCERROR      162$                        ;GOOD FORCE ERROR IF FORCER=1
2494 041764 103407      BCS      170$                      ;BR IF CARRY SET (GOOD RETURN)
2495 041766 010001      MOV      R0,R1                      ;SAVE CONTENTS OF TSSR
2496 041770      NEXT,ERRNO
2497 041770      162$: ERDF      ERRNO,T173SSR,PKTSSR        ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      613
                                .WORD      T173SSR
                                .WORD      PKTSSR
2498 041772 104455
2499 041772 001145

```

F12

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 28-7

SEQ 0148

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```

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

```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```

2545      and byte counts from 2 to 64.
2546      1. Initial FIFO status is checked
2547      2. The Write FIFO function.
2548      3. Read Status is executed and FIFO status is checked.
2549      4. Read FIFO is executed and the data and final status
2550         is checked.
2551
2552      TEST STEPS:
2553
2554      BEGIN
2555      Write to TSSR to soft initialize
2556      Do a WRITE CHARACTERISTICS to setup a message buffer
2557      Do a Write Subsystem READ STATUS
2558      If Input Ready NOT=1 Then Print Error
2559      If Output Ready NOT=0 Then Print Error
2560      If Data In Miss NOT=0 Then Print Error
2561      If Last Word NOT=0 Then Print Error
2562      REPEAT FOR DATA 0 TO 377, 377 TO 0, FLOATING 1'S,0'S AND ALL 1'S/0'S
2563      REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
2564      BEGIN
2565      Do a Write Subsystem WRITE NPR to set tape direction out
2566      Do a Write Subsystem WRITE FIFO
2567      Do a Write Subsystem READ STATUS
2568      If Input Ready NOT=1 Then Print Error
2569      If Output Ready NOT=1 Then Print Error
2570      If Data In Miss NOT=0 Then Print Error
2571      If Last Word NOT=0 Then Print Error
2572      Do Write Subsystem READ FIFO
2573      If Data read from FIFO NOT= to Data sent Then Print Error
2574      Do a Write Subsystem READ STATUS
2575      If Input Ready NOT=1 Then Print Error
2576      If Output Ready NOT=0 Then Print Error
2577      If Data In Miss NOT=0 Then Print Error
2578      If Last Word NOT=0 Then Print Error
2579      END
2580      END
2581      --
2582      BGNSUB                      ;////////// BEGIN SUBTEST //////////
2583      042154                      T6.3: TRAP C$BSUB
2584      042154 104402
2585      Write to TSSR register to soft initialize the controller
2586      5*: JSR PC,SOFINIT ;WRITE TO TSSR TO SOFT INITIALIZE
2587      BCS 10$ ;BR IF SOFT INIT OKAY
2588      MOV R0,R1 ;SAVE CONTENTS OF TSSR
2589      ERDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
2590      042166 104455 TRAP C$ERDF
2591      042170 001146 .WORD 614
2592      042172 003652 .WORD SFIERR
2593      042174 012034 .WORD SFIMSG
2594      Do a WRITE CHARACTERISTICS to setup a message buffer
2595      10$: CLR FATFLG ;CLEAR FATAL ERROR FLAG
2596      MOV @T17PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
2597      JSR PC,WRCHR ;DO WRITE CHARACTERISTICS COMMAND
2598      FORCERROR 42$ ;FORCE ERROR IF FORCER=1
2599      BCS 50$ ;BR IF CARRY SET (GOOD RETURN)

```

H12

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 28-9

SEQ 0150

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```

2596 042230 010001      MOV      R0,R1          ;SAVE CONTENTS OF TSSR
2597 042232      NEXT.ERRNO
2598 042232      42$:  ERRDF  ERRNO,T17SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                        TRAP      C$ERDF
                        .WORD      615
                        .WORD      T17SSR
                        .WORD      PKTSSR
2599 042242 005237 002222      INC      FATFLG          ;SET FATAL ERROR FLAG
2600 042246      50$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
2601      ; Do a Write Subsystem READ STATUS
2602 042250 004737 050034      JSR      PC,T17SRD          ;SETUP PACKET FOR READ STATUS
2603 042254 012704 050420      MOV      @T17PK2,R4          ;GET WRITE SUBSYSTEM COMMAND PACKET
2604 042260 010465 000000      MOV      R4,TSDB(R5)          ;SET THE PACKET ADDRESS TO EXECUTE
2605 042264 004737 016336      JSR      PC,CHKTSSR          ;WAIT FOR SSR TO SET
2606 042270      FORCERROR 62$          ;FORCE ERROR IF FORCER=1
2607 042304 103407      BCS      70$          ;BR IF CARRY SET (GOOD RETURN)
2608 042306 010001      MOV      R0,R1          ;SAVE CONTENTS OF TSSR
2609 042310      NEXT.ERRNO
2610 042310      62$:  ERRDF  ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                        TRAP      C$ERDF
                        .WORD      616
                        .WORD      T173SSR
                        .WORD      PKTSSR
2611 042320 005237 002222      INC      FATFLG          ;SET FATAL ERROR FLAG
2612 042324      70$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
2613      ; Set WORDS 0-7 of expd message buffer = to recv since not testing
2614 042326 004737 050216      JSR      PC,T17SETEXP          ;SET WORDS 0-7 EXPD=RCV
2615 042332 012701 046452      MOV      @T17EXSTA,R1          ;GET EXPECTED READ STATUS
2616 042336 012702 050312      MOV      @T17BFSTA,R2          ;GET RCV READ STATUS
2617 042342 012221      MOV      (R2)+,(R1)+          ;SET EXPD WORD #8 = RCV TEMP
2618 042344 011211      MOV      (R2),(R1)          ;SET EXPD WORD #9 = RCV TEMP
2619 042346 052711 000020      BIS      @S2.INRDY,(R1)          ;SET EXP INPUT READY= 1
2620 042352 042711 000040      BIC      @S2.OUTRDY,(R1)          ;SET EXP OUTPUT READY= 0
2621 042356 042711 000200      BIC      @S2.DIM,(R1)          ;SET EXP DATA IN MISS = 0
2622 042362 042711 000100      BIC      @S2.ILW,(R1)          ;SET EXP LAST WORD (ILW)=0
2623      ; If Input Ready NOT=1 then Print Error
2624      ; If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2625      ; If Last Word NOT=0 Then Print Error
2626 042366 005000      CLR      R0          ;HIGH RCV ADDRESS FOR CKMSG2
2627 042370 012701 050272      MOV      @T17BFR,R1          ;LOW RCV ADDRESS FOR CKMSG2
2628 042374 012702 046432      MOV      @T17EXP,R2          ;EXPD ADDRESS
2629 042400 012703 000024      MOV      @20.,R3          ;NUMBER OF BYTES TO COMPARE
2630 042404 004737 011500      JSR      PC,CKMSG2          ;EXPD EQUAL RCV?
2631 042410      FORCERROR 82$,NOTSSR          ;880
2632 042420 103404      BCS      90$          ;BR IF YES
2633 042422      NEXT.ERRNO
2634 042422      82$:  ERRHRD  ERRNO,T171CMP,MSGSTAT ;REPORT ERROR
                        TRAP      C$ERHRD
                        .WORD      617
                        .WORD      T171CMP
                        .WORD      MSGSTAT
2635 042432      90$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
2636
2637

```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```

2638
2639      ; REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
2640      ; TSTFLAG =1 FOR INCREMENT TEST PATTERN
2641      ;           =2 FOR DECREMENT TEST PATTERN
2642      ;           =3 FOR TSTBLK TABLE PATTERN
2643 042434 012737 000001 002314      MOV     #1,TSTFLAG      ;TEST PATTERN FLAG
2644 042442      95$:      MOV     #2,COUNT      ;GET FIRST BYTE COUNT
2645 042442 012737 000002 002310
2646 042450      100$:
2647      ; Do a Write Subsystem WRITE NPR to set tape direction out
2648 042450 012700 000100      MOV     #NPR,OUT,R0      ;SET TAPE DIRECTION OUT
2649 042454 004737 050076      JSR     PC,T17SNPR      ;SETUP T17PK2 FOR WRITE NPR
2650 042460 012704 050420      MOV     #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
2651 042464 010465 000000      MOV     R4,TSD0B(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
2652 042470 004737 016336      JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
2653 042474      FORCERROR 102$      ;DO FORCE ERROR IF FORCER=1
2654 042510 103407      BCS     105$      ;BR IF CARRY SET (GOOD RETURN)
2655 042512 010001      MOV     R0,R1      ;SAVE CONTENTS OF TSSR
2656 042514      NEXT.ERRNO
2657 042514      102$:      ERRDF  ERRNO,T174SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
2658      042514 104455      TRAP     C$ERDF
2659      042516 001152      .WORD    618
2660      042520 047043      .WORD    T174SSR
2661      042522 012046      .WORD    PKTSSR
2662 042524 005237 002222      INC     FATFLG      ;SET FATAL ERROR FLAG
2663 042530      105$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
2664      042530 104406      TRAP     C$CLP1
2665      ; Do a Write Subsystem WRITE FIFO
2666 042532 004737 050176      JSR     PC,T17CLEXP      ;CLEAR EXPD BUFFER
2667 042536 012701 046554      MOV     #T17WFDATA,R1      ;EXPD WRITE FIFO DATA BUFFER
2668 042542 013702 002310      MOV     COUNT,R2      ;TEST PATTERN SIZE
2669 042546 022737 000001 002314      CMP     #1,TSTFLAG      ;INCREMENT PATTERN THIS TIME THRU?
2670 042554 001005      115$:      BNE     115$      ;BR IF NO
2671 042556 005000      CLR     R0      ;INCREMENT TEST PATTERN
2672 042560 110021      110$:      MOV     R0,(R1)+      ;STORE INCREMENT TEST BYTE
2673 042562 005200      INC     R0      ;SET NEXT PATTERN
2674 042564 005302      DEC     R2      ;DONE?
2675 042566 003374      BGT     110$      ;BR IF NO
2676 042570 022737 000002 002314      115$:      CMP     #2,TSTFLAG      ;DECREMENT PATTERN THIS TIME THRU?
2677 042576 001006      BNE     125$      ;BR IF NO
2678 042600 012700 000377      MOV     #377,R0      ;DECREMENT TEST PATTERN
2679 042604 110021      120$:      MOV     R0,(R1)+      ;STORE DECREMENT TEST BYTE
2680 042606 005300      DEC     R0      ;SET NEXT PATTERN
2681 042610 005302      DEC     R2      ;DONE?
2682 042612 003374      BGT     120$      ;BR IF NO
2683 042614 022737 000003 002314      125$:      CMP     #3,TSTFLAG      ;TSTBLK PATTERNS THIS TIME THRU?
2684 042622 001005      BNE     135$      ;BR IF NO
2685 042624 012700 002752      MOV     #TSTBLK,R0      ;FLOAT 1'S/0'S ETC. TEST TABLE
2686 042630 112021      130$:      MOV     (R0)+,(R1)+      ;STORE A TSTBLK BYTE
2687 042632 005302      DEC     R2      ;DONE?
2688 042634 003375      BGT     130$      ;BR IF NO
2689 042636      135$:      MOV     COUNT,R0      ;FIFO BYTE COUNT
2690 042642 012701 046554      MOV     #T17WFDATA,R1      ;FIFO WRITE DATA ADDRESS
2691 042646 004737 050122      JSR     PC,T17WFIF      ;SETUP T17PK2 FOR WRITE FIFO
2692 042652 012704 050420      MOV     #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
2693 042656 010465 000000      MOV     R4,TSD0B(R5)      ;SET THE PACKET ADDRESS TO EXECUTE

```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```

2690 042662 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
2691 042666      FORCERROR      142$      ;000FORCE ERROR IF FORCER=1
2692 042702 103407      BCS      150$      ;BR IF CARRY SET (GOOD RETURN)
2693 042704 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2694 042706      NEXT.ERRNO
2695 042706      142$: ERRDF      ERRNO,T175SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      619
                                .WORD      T175SSR
                                .WORD      PKTSSR
                                2696 042716 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
2697 042722      150$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
2698
2699      ; Do a Write Subsystem READ STATUS
2700 042724 004737 050034      JSR      PC,T17SRD      ;SETUP PACKET FOR READ STATUS
2701 042730 012704 050420      MOV      @T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
2702 042734 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
2703 042740 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
2704 042744      FORCERROR      157$      ;000FORCE ERROR IF FORCER=1
2705 042760 103407      BCS      160$      ;BR IF CARRY SET (GOOD RETURN)
2706 042762 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2707 042764      NEXT.ERRNO
2708 042764      157$: ERRDF      ERRNO,T173SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      620
                                .WORD      T173SSR
                                .WORD      PKTSSR
                                2709 042774 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
2710 043000      160$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
2711
2712      ; Set WORDS 0-7 of expd message buffer = to recv since not testing
2713 043002 004737 050216      JSR      PC,T17SETEXP      ;SET WORDS 0-7 EXPD=RECV
2714 043006 012701 046452      MOV      @T17EXSTA,R1      ;GET EXPECTED READ STATUS
2715 043012 012702 050312      MOV      @T17BFSTA,R2      ;GET RECV READ STATUS
2716 043016 012221      MOV      (R2)+,(R1)+      ;SET EXPD WORD #8 = RECV TEMP
2717 043020 011211      MOV      (R2),(R1)      ;SET EXPD WORD #9 = RECV TEMP
2718 043022 052711 000020      BIS      @S2.INRDY,(R1)      ;SET EXP INPUT READY= 1
2719 043026 052711 000040      BIS      @S2.OTRDY,(R1)      ;SET EXP OUTPUT READY= 1
2720 043032 042711 000200      BIC      @S2.DIM,(R1)      ;SET EXP DATA IN MISS = 0
2721 043036 042711 000100      BIC      @S2.ILW,(R1)      ;SET EXP LAST WORD (ILW)=0
2722      ; If Input Ready NOT=1 then Print Error
2723      ; If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
2724 043042 005000      CLR      R0      ;HIGH RECV ADDRESS FOR CKMSG2
2725 043044 012701 050272      MOV      @T17BFR,R1      ;LOW RECV ADDRESS FOR CKMSG2
2726 043050 012702 046432      MOV      @T17EXP,R2      ;EXPD ADDRESS
2727 043054 012703 000024      MOV      @20.,R3      ;NUMBER OF BYTES TO COMPARE
2728 043060 004737 011500      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
2729 043064      FORCERROR      162$,NOTSSR      ;000
2730 043074 103404      BCS      170$      ;BR IF YES
2731 043076      NEXT.ERRNO
2732 043076      162$: ERRHRD      ERRNO,T173CMP,MSGSTAT      ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD      621
                                .WORD      T173CMP
                                .WORD      MSGSTAT
2733 043076 104456
2734 043100 001155
2735 043102 047373
2736 043104 012350

```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

Line	Address	Hex	Label	Code	Comment
2733	043106		170\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
	043106	104406			TRAP C\$CLP1
2734					
2735					
2736	043110	013700		002310	
2737	043114	004737		050156	
2738	043120	012704		050420	
2739	043124	010465		000000	
2740	043130	004737		016336	
2741	043134				
2742	043150	103407			
2743	043152	010001			
2744	043154				
2745	043154		172\$:		
	043154	104455			
	043156	001156			
	043160	047152			
	043162	012046			
2746	043164	005237		002222	
2747	043170		180\$:		
	043170	104406			
2748					
2749					
2750	043172	005000			
2751	043174	012702		046554	
2752	043200	012701		050312	
2753	043204	013703		002310	
2754	043210	004737		011500	
2755	043214				
2756	043224	103406			
2757	043226				
2758	043226	013701		002310	
2759	043232		192\$:		
	043232	104456			
	043234	001157			
	043236	047542			
	043240	012170			
2760	043242		200\$:		
	043242	104406			
2761					
2762					
2763	043244	004737		050034	
2764	043250	012704		050420	
2765	043254	010465		000000	
2766	043260	004737		016336	
2767	043264				
2768	043300	103407			
2769	043302	010001			
2770	043304				
2771	043304		212\$:		
	043304	104455			
	043306	001160			
	043310	046776			
	043312	012046			
2772	043314	005237		002222	
2773	043320		220\$:		
	043320	104406			

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```

2774      ; Set WORDS 0-7 of expd message buffer = to recv since not testing
2775 043322 004737 050216      JSR PC,T17SETEXP      ;SET WORDS 0-7 EXPD=RCV
2776 043326 012701 046452      MOV #T17EXSTA,R1      ;GET EXPECTED READ STATUS
2777 043332 012702 050312      MOV #T17BFSTA,R2      ;GET RCV READ STATUS
2778 043336 012221              MOV (R2)+,(R1)+      ;SET EXPD WORD #8 = RCV TEMP
2779 043340 011211              MOV (R2),(R1)         ;SET EXPD WORD #9 = RCV TEMP
2780 043342 052711 000020      BIS #S2.INRDY,(R1)     ;SET EXP INPUT READY= 1
2781 043346 042711 000040      BIC #S2.OTRDY,(R1)     ;SET EXP OUTPUT READY= 0
2782 043352 042711 000200      BIC #S2.DIM,(R1)       ;SET EXP DATA IN MISS = 0
2783 043356 042711 000100      BIC #S2.ILW,(R1)       ;SET EXP LAST WORD (ILW)=0
2784      ; If Input Ready NOT=1 then Print Error
2785      ; If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2786 043362 005000              CLR R0                ;HIGH RCV ADDRESS FOR CKMSG2
2787 043364 012701 050272      MOV #T17BFR,R1         ;LOW RCV ADDRESS FOR CKMSG2
2788 043370 012702 046432      MOV #T17EXP,R2         ;EXPD ADDRESS
2789 043374 012703 000024      MOV #20.,R3            ;NUMBER OF BYTES TO COMPARE
2790 043400 004737 011500      JSR PC,CKMSG2           ;EXPD EQUAL RCV?
2791 043404              FORCERROR 232$,NOTSSR        ;000
2792 043414 103404              BCS 240$              ;BR IF YES
2793 043416              NEXT.ERRNO
2794 043416 104456              232$: ERRHRD ERRNO,T174CMP,MSGSTAT ;REPORT ERROR
                                TRAP C$ERHRD
                                .WORD 625
                                .WORD T174CMP
                                .WORD MSGSTAT
2795 043426 104406              240$: CKLOOP           ;LOOP ON ERROR, IF FLAG SET
                                TRAP C$CLP1
2796 043430              FORCEXIT 250$              ;000
2797 043440 005237 002310      INC COUNT             ;GET NEXT BYTE COUNT
2798 043444 023727 002310 000077      CMP COUNT,#77   ;DONE 0 TO 77
2799 043452 101002              BHI 250$             ;BR IF YES
2800 043454 000137 042450      JMP 100$             ;DO ANOTHER BYTE COUNT
2801 043460 005237 002314 000003      250$: INC TSTFLAG ;GET NEXT TEST PATTERN CODE
2802 043464 023727 002314              CMP TSTFLAG,#3 ;DONE INC,DEC,TSTBLK PATTERNS?
2803 043472 101002              BHI 255$             ;BR IF YES
2804 043474 000137 042442      JMP 95$              ;DO ANOTHER TEST PATTERN
2805 043500              255$: ENDSUB
                                ;////////// END SUBTEST ///////////
                                L10061:
                                TRAP C$ESUB
2806 043500 104403              260$: TST FATFLG      ;ANY FATAL ERRORS ?
2807              BEQ 260$      ;BRANCH IF NOT
2808 043502 005737 002222      JSR PC,CKDROP         ;TRY TO DROP THE UNIT
2809 043506 001402
2810 043510 004737 017202
2811 043514
2812
2813
2814
2815      .SBTTL TEST 6: SUBTEST 4: FIFO Verify ILW Status
2816
2817      ;**
2818      ; TEST 6: SUBTEST 4:
2819      ;
2820      ; SUBTEST DESCRIPTION:
2821      ;
2822      ; This subtest verifies that reading the FIFO when it is
2823      ; empty causes the Last Word (ILW) status to assert.

```

TEST 6: SUBTEST 4: FIFO VERIFY ILW STATUS

```

2824      ;
2825      ; TEST STEPS:
2826      ;
2827      ; BEGIN
2828      ;       Write to TSSR to soft initialize
2829      ;       Do Write Subsystem READ FIFO with byte count equal to 1
2830      ;       Do a Write Subsystem READ STATUS
2831      ;       If Input Ready NOT=1 Then Print Error
2832      ;       If Output Ready NOT=0 Then Print Error
2833      ;       If Data In Miss NOT=0 Then Print Error
2834      ;       If Last Word (ILW) NOT=1 Then Print Error
2835      ;       END
2836      ; --
2837      043514      BGNSUB                      ;////////// BEGIN SUBTEST //////////
2838      043514      104402                      T6.4: TRAP C$BSUB
2839      ;
2840      043516      ; Write to TSSR register to soft initialize the controller
2841      043516      004737      015774      5$: JSR PC,SOFINIT ;WRITE TO TSSR TO SOFT INITIALIZE
2842      043522      103405      BCS 10$ ;BR IF SOFT INIT OKAY
2843      043524      010001      MOV RO,R1 ;SAVE CONTENTS OF TSSR
2844      043526      104455      ERRDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
2845      043526      001161      TRAP C$ERDF
2846      043530      003652      .WORD 625
2847      043532      012034      .WORD SFIERR
2848      043534      012034      .WORD SFIMSG
2849      ;
2850      043536      005037      002222      10$: Do a WRITE CHARACTERISTICS to setup a message buffer
2851      043542      012704      050250      CLR FATFLG ;CLEAR FATAL ERROR FLAG
2852      043546      004737      010662      MOV #T17PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
2853      043552      103407      JSR PC,WRTCHR ;DO WRITE CHARACTERISTICS COMMAND
2854      043566      010001      FORCERROR 42$ ;GOODFORCE ERROR IF FORCER=1
2855      043570      010001      BCS 50$ ;BR IF CARRY SET (GOOD RETURN)
2856      043572      010001      MOV RO,R1 ;SAVE CONTENTS OF TSSR
2857      043572      010001      NEXT,ERRNO
2858      043572      104455      42$: ERRDF ERRNO,T17SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
2859      043574      001162      TRAP C$ERDF
2860      043576      046675      .WORD 626
2861      043600      012046      .WORD T17SSR
2862      043602      005237      002222      INC FATFLG ;SET FATAL ERROR FLAG
2863      043606      104406      50$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
2864      043606      104406      TRAP C$CLP1
2865      ;
2866      043610      012700      000001      Do Write Subsystem READ FIFO with byte count equal to 1
2867      043614      004737      050156      MOV #1,R0 ;SET READ BYTE COUNT
2868      043620      012704      050420      JSR PC,T17RFIF ;SETUP T17PK2 FOR READ FIFO
2869      043624      010465      000000      MOV #T17PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
2870      043630      004737      016336      MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
2871      043634      103407      JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
2872      043650      010001      FORCERROR 142$ ;GOODFORCE ERROR IF FORCER=1
2873      043652      010001      BCS 150$ ;BR IF CARRY SET (GOOD RETURN)
2874      043654      010001      MOV RO,R1 ;SAVE CONTENTS OF TSSR
2875      043654      104455      142$: ERRDF ERRNO,T176SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
2876      043656      001163      TRAP C$ERDF
2877      043656      001163      .WORD 627

```

TEST 6: SUBTEST 4: FIFO VERIFY ILW STATUS

```

      043660 047152
      043662 012046
2868 043664 005237 002222      INC    FATFLG      ;SET FATAL ERROR FLAG
2869 043670      104406      150$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      043670      104406      TRAP    C$CLP1
2870
2871      ;      Do a Write Subsystem READ STATUS
2872 043672 004737 050034      JSR     PC,T17SRD      ;SETUP PACKET FOR READ STATUS
2873 043676 012704 050420      MOV     @T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
2874 043702 010465 000000      MOV     R4,T5DB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
2875 043706 004737 016336      JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
2876 043712      FORCERROR 162$      ;GOODFORCE ERROR IF FORCER=1
2877 043726 103407      BCS     170$      ;BR IF CARRY SET (GOOD RETURN)
2878 043730 010001      MOV     R0,R1      ;SAVE CONTENTS OF TSSR
2879 043732      NEXT.ERRNO
2880 043732      162$: ERRDF  ERRNO,T173SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      043732 104455      TRAP    C$ERDF
      043734 001164      .WORD   628
      043736 046776      .WORD   T173SSR
      043740 012046      .WORD   PKTSSR
2881 043742 005237 002222      INC    FATFLG      ;SET FATAL ERROR FLAG
2882 043746      104406      170$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      043746      104406      TRAP    C$CLP1
2883      ;      Set WORDS 0-7 of expd message buffer = to recv since not testing
2884 043750 004737 050216      JSR     PC,T17SETEXP      ;SET WORDS 0-7 EXPD=RECV
2885 043754 012701 046452      MOV     @T17EXSTA,R1      ;GET EXPECTED READ STATUS
2886 043760 012702 050312      MOV     @T17BFSTA,R2      ;GET RECV READ STATUS
2887 043764 012221      MOV     (R2)+,(R1)+      ;SET EXPD WORD #8 = RECV TEMP
2888 043766 011211      MOV     (R2),(R1)      ;SET EXPD WORD #9 = RECV TEMP
2889 043770 052711 000020      BIS     @S2.INRDY,(R1)      ;SET EXP INPUT READY= 1
2890 043774 042711 000040      BIC     @S2.OTRDY,(R1)      ;SET EXP OUTPUT READY= 0
2891 044000 042711 000200      BIC     @S2.DIM,(R1)      ;SET EXP DATA IN MISS = 0
2892 044004 052711 000100      BIS     @S2.ILW,(R1)      ;SET EXP LAST WORD (ILW)=1
2893      ;      If Input Ready NOT=1 then Print Error
2894      ;      If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2895      ;      If Last Word (ILW) NOT=1 Then Print Error
2896 044010 005000      CLR     R0      ;HIGH RECV ADDRESS FOR CKMSG2
2897 044012 012701 050272      MOV     @T17BFR,R1      ;LOW RECV ADDRESS FOR CKMSG2
2898 044016 012702 046432      MOV     @T17EXP,R2      ;EXPD ADDRESS
2899 044022 012703 000024      MOV     @20.,R3      ;NUMBER OF BITES TO COMPARE
2900 044026 004737 011500      JSR     PC,CKMSG2      ;EXPD EQUAL RECV?
2901 044032      FORCERROR 172$,NOTSSR      ;GOOD
2902 044042 103404      BCS     180$      ;BR IF YES
2903 044044      NEXT.ERRNO
2904 044044      172$: ERRHRD  ERRNO,T176CMP,MSGSTAT      ;REPORT ERROR
      044044 104456      TRAP    C$ERHRD
      044046 001165      .WORD   629
      044050 047616      .WORD   T176CMP
      044052 012350      .WORD   MSGSTAT
2905 044054      180$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      044054      104406      TRAP    C$CLP1
2906
2907 044056      ENDSUB      ;////////// END SUBTEST //////////
      044056      110062:      .WORD   110062
      044056 104403      TRAP    C$ESUB
2908
2909 044060 005737 002222      TST     FATFLG      ;ANY FATAL ERRORS ?

```

B13

TSV5 - HARDWARE TESTS MACRO M1200 20 MAR-84 08:26 PAGE 28-16

SEQ 0157

TEST 6: SUBTEST 4: (FIFO VERIFY ILW STATUS)

```

2910 044064 001401 BEQ 2601 ;BRANCH IF NOT
2911 044066 001737 017202 JSR PC,CKDROP ;TRY TO DROP THE UNIT
2912 044072 2601:
2913
2914
2915 .SBTTL TEST 6: SUBTEST 5: FIFO Verify Input Ready
2916
2917
2918 ; TEST 6: SUBTEST 5:
2919 ; SUBTEST DESCRIPTION:
2920 ;
2921 ; This subtest verifies that writing 64. bytes into the FIFO
2922 ; without reading any out causes the Input Ready status to
2923 ; negate. The Subtest then verifies that writing a 65th byte
2924 ; into the FIFO causes the Data In Miss status to assert.
2925 ; Next it is verified that the original 64 bytes can be read
2926 ; out correctly and that the data has not been corrupted.
2927
2928 ; TEST STEPS:
2929 ;
2930 ; BEGIN
2931 ; Write to TSSR to soft initialize
2932 ; Do a WRITE CHARACTERISTICS to setup a message buffer
2933 ; Do a Write Subsystem WRITE NPR to set tape direction out
2934 ; Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
2935 ; Do a Write Subsystem READ STATUS
2936 ; If Input Ready NOT=0 Then Print Error
2937 ; If Output Ready NOT=1 Then Print Error
2938 ; If Data In Miss NOT=0 Then Print Error
2939 ; Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
2940 ; Do a Write Subsystem READ STATUS
2941 ; If Input Ready NOT=0 Then Print Error
2942 ; If Output Ready NOT=1 Then Print Error
2943 ; If Data In Miss NOT=1 Then Print Error
2944 ; Do Write Subsystem READ FIFO
2945 ; If Data read from FIFO NOT= to Data sent Then Print Error
2946 ; Do a Write Subsystem READ STATUS
2947 ; If Input Ready NOT=1 Then Print Error
2948 ; If Output Ready NOT=0 Then Print Error
2949 ; If Data In Miss NOT=1 Then Print Error
2950 ; END
2951
2952
2953 044072 BGNSUB ;////////// BEGIN SUBTEST //////////
2954 044072 104402 ;
2955 044072 104402 ;
2956 044074 004737 015774 ;
2957 044074 103405 ;
2958 044100 01:001 ;
2959 044102 01:001 ;
2960 044104 104455 ;
2961 044106 001165 ;
2962 044110 003652 ;
2963 044112 012034 ;

```

C13

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 28-17

SEQ 0158

TEST 6: SUBTEST 5: FIFO VERIFY INPUT READY

```

2961      Do a WRITE CHARACTERISTICS to setup a message buffer
2962 044114 005037 002222      104: CLR    FATFLG      ;CLEAR FATAL ERROR FLAG
2963 044120 012704 050250      MOV    @T17PK2,R4      ;GET THE ADDRESS OF COMMAND PACKET
2964 044124 004737 010662      JSR    PC,WRCHR      ;DO WRITE CHARACTERISTICS COMMAND
2965 044130      FORCERROR      42:      ;DO FORCE ERROR IF FORCER=1
2966 044144 103407      BCS    50:      ;BR IF CARRY SET (GOOD RETURN)
2967 044146 010001      MOV    R0,R1      ;SAVE CONTENTS OF TSSR
2968 044150      NEXT,ERRNO
2969 044150      42: ERRDF  ERRNO,T17SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      044150 104455      TRAP    C1ERDF
      044152 001166      .WORD  630
      044154 046675      .WORD  T17SSR
      044156 012046      .WORD  PKTSSR
2970 044160 005237 002222      50: INC    FATFLG      ;SET FATAL ERROR FLAG
2971 044164      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      044164 104406      TRAP    C1CLP1

2972      Do a Write Subsystem WRITE NPR to set tape direction out
2973      100: MOV    @NP.OUT,R0      ;SET TAPE DIRECTION OUT
2974 044166 012700 000100      JSR    PC,T17SNPR      ;SETUP T17PK2 FOR WRITE NPR
2975 044172 004737 050076      MOV    @T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
2976 044176 012704 050420      MOV    R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
2977 044202 010465 000000      JSR    PC,CHKTSSR      ;WAIT FOR SSR TO SET
2978 044206 004737 016336      FORCERROR      102:      ;DO FORCE ERROR IF FORCER=1
2979 044212      BCS    105:      ;BR IF CARRY SET (GOOD RETURN)
2980 044226 103407      MOV    R0,R1      ;SAVE CONTENTS OF TSSR
2981 044230 010001      NEXT,ERRNO
2982 044232      102: ERRDF  ERRNO,T174SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
2983 044232      TRAP    C1ERDF
      044232 104455      .WORD  631
      044234 001167      .WORD  T174SSR
      044236 047043      .WORD  PKTSSR
      044240 012046
2984 044242 005237 002222      105: INC    FATFLG      ;SET FATAL ERROR FLAG
2985 044246      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      044246 104406      TRAP    C1CLP1

2986      Do a Write Subsystem WRITE FIFO 64 bytes incrementing pattern
2987      002310      MOV    @64.,COUNT      ;WRITE 64 BYTES
2988 044250 012737 000100      MOV    @T17WFDATA,R1      ;EXPD WRITE FIFO DATA BUFFER
2989 044256 012701 046554      MOV    @64.,R2      ;TEST PATTERN SIZE
2990 044262 012702 000100      CLR    R0      ;INCREMENT TEST PATTERN
2991 044266 005000      MOVB    R0,(R1)+      ;STORE INCREMENT TEST BYTE
2992 044270 110021      INC    R0      ;SET NEXT PATTERN
2993 044272 005200      DEC    R2      ;DONE?
2994 044274 005302      BGT    110:      ;BR IF NO
2995 044276 003374      MOV    COUNT,R0      ;FIFO BYTE COUNT
2996 044300 013700 002310      MOV    @T17WFDATA,R1      ;FIFO WRITE DATA ADDRESS
2997 044304 012701 046554      JSR    PC,T17WIF      ;SETUP T17PK2 FOR WRITE FIFO
2998 044310 004737 050122      MOV    @T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
2999 044314 012704 050420      MOV    R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
3000 044320 010465 000000      JSR    PC,CHKTSSR      ;WAIT FOR SSR TO SET
3001 044324 004737 016336      FORCERROR      142:      ;DO FORCE ERROR IF FORCER=1
3002 044330      BCS    150:      ;BR IF CARRY SET (GOOD RETURN)
3003 044344 103407      MOV    R0,R1      ;SAVE CONTENTS OF TSSR
3004 044346 010001      NEXT,ERRNO
3005 044350      142: ERRDF  ERRNO,T175SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
3006 044350      TRAP    C1ERDF
      044350 104455

```


E13

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 28-19

SEQ 0160

TEST 6: SUBTEST 5: FIFO VERIFY INPUT READY

```

3050 044562 012704 050420      MOV      @T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
3051 044566 010465 000000      MOV      R4,TSD8(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
3052 044572 004737 016336      JSR      PC,CHKTSSR    ;WAIT FOR SSR TO SET
3053 044576                FORCERROR 172$      ;SSDFORCE ERROR IF FORCER=1
3054 044612 103407                BCS      180$      ;BR IF CARRY SET (GOOD RETURN)
3055 044614 010001                MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3056 044616                NEXT,ERRNO
3057 044616 172$: ERRDF  ERRNO,T175SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     635
                                .WORD     T175SSR
                                .WORD     PKTSSR
                                044616 104455
                                044620 001173
                                044622 047106
                                044624 012046
3058 044626 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
3059 044632 180$: CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                044632 104406
3060
3061      ; Do a Write Subsystem READ STATUS
3062      ; If Input Ready NOT=0 Then Print Error
3063      ; If Output Ready NOT=1 Then Print Error
3064      ; If Data In Miss NOT=1 Then Print Error
3065 044634 004737 050034      JSR      PC,T17SRD      ;SETUP PACKET FOR READ STATUS
3066 044640 012704 050420      MOV      @T17PK2,R4    ;GET WRITE SUBSYSTEM COMMAND PACKET
3067 044644 010465 000000      MOV      R4,TSD8(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
3068 044650 004737 016336      JSR      PC,CHKTSSR    ;WAIT FOR SSR TO SET
3069 044654                FORCERROR 187$      ;SSDFORCE ERROR IF FORCER=1
3070 044670 103407                BCS      190$      ;BR IF CARRY SET (GOOD RETURN)
3071 044672 010001                MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3072 044674                NEXT,ERRNO
3073 044674 187$: ERRDF  ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     636
                                .WORD     T173SSR
                                .WORD     PKTSSR
                                044674 104455
                                044676 001174
                                044700 046776
                                044702 012046
3074 044704 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
3075 044710 190$: CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                044710 104406
3076      ; Set WORDS 0-7 of expd message buffer = to recv since not testing
3077 044712 004737 050216      JSR      PC,T17SETEXP    ;SET WORDS 0-7 EXPD=RECV
3078 044716 012701 046452      MOV      @T17EXSTA,R1  ;GET EXPECTED READ STATUS
3079 044722 012702 050312      MOV      @T17BFSTA,R2  ;GET RECV READ STATUS
3080 044726 012221                MOV      (R2)+,(R1)+  ;SET EXPD WORD #8 = RECV TEMP
3081 044730 011211                MOV      (R2),(R1)    ;SET EXPD WORD #9 = RECV TEMP
3082 044732 042711 000020      BIC      @S2.INRDY,(R1)  ;SET EXP INPUT READY= 0
3083 044736 052711 000040      BIS      @S2.OTRDY,(R1)  ;SET EXP OUTPUT READY= 1
3084 044742 052711 000200      BIS      @S2.DIM,(R1)  ;SET EXP DATA IN MISS = 1
3085 044746 005000                CLR      R0          ;HIGH RECV ADDRESS FOR CKMSG2
3086 044750 012701 050272      MOV      @T17BFR,R1    ;LOW RECV ADDRESS FOR CKMSG2
3087 044754 012702 046432      MOV      @T17EXP,R2    ;EXPD ADDRESS
3088 044760 012703 000024      MOV      @20.,R3      ;NUMBER OF BYTES TO COMPARE
3089 044764 004737 011500      JSR      PC,CKMSG2    ;EXPD EQUAL RECV?
3090 044770                FORCERROR 192$,NOTSSR ;SSD
3091 045000 103404                BCS      200$      ;BR IF YES
3092 045002                NEXT,ERRNO
3093 045002 192$: ERRHRD  ERRNO,T173CMP,MSGSTAT ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD     637
                                .WORD     T173CMP
                                045002 104456
                                045004 001175
                                045006 047373

```

F13

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 28-20

SEQ 0161

TEST 6: SUBTEST 5: FIFO VERIFY INPUT READY

3094	045010	012350		200\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	.WORD MSGSTAT
	045012	104406				TRAP	C\$CLP1
3095				:	Do Write Subsystem READ FIFO		
3096	045014	013700	002310		MOV COUNT,R0	;SET READ BYTE COUNT	
3097	045020	004737	050156		JSR PC,T17RFIF	;SETUP T17PK2 FOR READ FIFO	
3098	045024	012704	050420		MOV @T17PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
3099	045030	010465	000000		MOV R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
3100	045034	004737	016336		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET	
3101	045040				FORCERROR 212\$	;SSDFORCE ERROR IF FORCER=1	
3102	045054	103407			BCS 220\$	;BR IF CARRY SET (GOOD RETURN)	
3103	045056	010001			MOV R0,R1	;SAVE CONTENTS OF TSSR	
3104	045060				NEXT,ERRNO		
3105	045060			212\$:	ERRDF ERRNO,T176SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	
	045060	104455				TRAP	C\$ERDF
	045062	001176				.WORD	638
	045064	047152				.WORD	T176SSR
	045066	012046				.WORD	PKTSSR
3106	045070	005237	002222		INC FATFLG	;SET FATAL ERROR FLAG	
3107	045074			220\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	045074	104406				TRAP	C\$CLP1
3108				:	If Data read from FIFO NOT= to Data sent Then Print Error		
3109					CLR R0	;HIGH RECV ADDRESS FOR CKMSG2	
3110	045076	005000			MOV @T17WFDATA,R2	;GET EXPECTED ADDRESS FOR CKMSG2	
3111	045100	012702	046554		MOV @T17RFSTA,R1	;GET RECEIVED ADDRESS FOR CKMSG2	
3112	045104	012701	050312		MOV COUNT,R3	;NUMBER OF BYTES TO COMPARE	
3113	045110	013703	002310		JSR PC,CKMSG2	;EXPD EQUAL RECV?	
3114	045114	004737	011500		FORCERROR 232\$,NOTSSR	;SSD	
3115	045120				BCS 240\$	;BR IF YES	
3116	045130	103406			NEXT,ERRNO		
3117	045132			232\$:	MOV COUNT,R1	;GET BYTE COUNT	
3118	045132	013701	002310		ERRHRD ERRNO,T175CMP,FIFEXP	;REPORT ERROR	
3119	045136					TRAP	C\$ERHRD
	045136	104456				.WORD	639
	045140	001177				.WORD	T175CMP
	045142	047542				.WORD	FIFEXP
	045144	012170					
3120	045146			240\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	045146	104406				TRAP	C\$CLP1
3121				:	Do a Write Subsystem READ STATUS		
3122				:	If Input Ready NOT=1 Then Print Error		
3123				:	If Output Ready NOT=0 Then Print Error		
3124				:	If Data In Miss NOT=1 Then Print Error		
3125				:			
3126	045150	004737	050034		JSR PC,T17SRD	;SETUP PACKET FOR READ STATUS	
3127	045154	012704	050420		MOV @T17PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
3128	045160	010465	000000		MOV R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
3129	045164	004737	016336		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET	
3130	045170				FORCERROR 252\$	;SSDFORCE ERROR IF FORCER=1	
3131	045204	103407			BCS 260\$	;BR IF CARRY SET (GOOD RETURN)	
3132	045206	010001			MOV R0,R1	;SAVE CONTENTS OF TSSR	
3133	045210				NEXT,ERRNO		
3134	045210			252\$:	ERRDF ERRNO,T173SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	
	045210	104455				TRAP	C\$ERDF
	045212	001200				.WORD	640
	045214	046776				.WORD	T173SSR
	045216	012046				.WORD	PKTSSR

TEST 6: SUBTEST 5: FIFO VERIFY INPUT READY

```

3135 045220 005237 002222          INC    FATFLG          ;SET FATAL ERROR FLAG
3136 045224          260$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP    C$CLP1
3137          ; Set WORDS 0-7 of expd message buffer = to recv since not testing
3138 045226 004737 050216          JSR    PC,T17SETEXP      ;SET WORDS 0-7 EXPD=RCV
3139 045232 012701 046452          MOV    $T17EXSTA,R1      ;GET EXPECTED READ STATUS
3140 045236 012702 050312          MOV    $T17BFSTA,R2      ;GET RCV READ STATUS
3141 045242 012221          MOV    (R2)+,(R1)+          ;SET EXPD WORD 08 = RCV TEMP
3142 045244 011211          MOV    (R2),(R1)             ;SET EXPD WORD 09 = RCV TEMP
3143 045246 052711 000020          BIS    $S2.INRDY,(R1)    ;SET EXP INPUT READY= 1
3144 045252 042711 000040          BIC    $S2.OU1RDY,(R1)    ;SET EXP OUTPUT READY= 0
3145 045256 052711 000200          BIS    $S2.DIM,(R1)      ;SET EXP DATA IN MISS = 1
3146 045262 005000          CLR    R0                    ;HIGH RCV ADDRESS FOR CKMSG2
3147 045264 012701 050272          MOV    $T17BFR,R1        ;LOW RCV ADDRESS FOR CKMSG2
3148 045270 012702 046432          MOV    $T17EXP,R2        ;EXPD ADDRESS
3149 045274 012703 000024          MOV    $20.,R3           ;NUMBER OF BYTES TO COMPARE
3150 045300 004737 011500          JSR    PC,CKMSG2          ;EXPD EQUAL RCV?
3151 045304          FORCERROR 272$,NOTSSR ;880
3152 045314 103404          BCS    280$                  ;BR IF YES
3153 045316          NEXT.ERRNO
3154 045316          272$: ERHRD  ERRNO,T174CMP,MSGSTAT ;REPORT ERROR
                                TRAP    C$ERHRD
                                .WORD   641
                                .WORD   T174CMP
                                .WORD   MSGSTAT
3155 045326          280$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP    C$CLP1
3156          ;
3157 045330          ENDSUB                    ;////////// END SUBTEST ///////////
                                L10063:
                                TRAP    C$ESUB
3158          ;
3159 045332 005737 002222          TST    FATFLG          ;ANY FATAL ERRORS ?
3160 045336 001402          BEQ    300$              ;BRANCH IF NOT
3161 045340 004737 017202          JSR    PC,CKDROP        ;TRY TO DROP THE UNIT
3162 045344          300$:

```

.SBTTL TSV5 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.
;
;

```

H13

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 28-22

SEQ 0163

TEST 6: SUBTEST 6: FIFO VERIFY RESET FIFO TEST

```

3184      ; TEST STEPS:
3185      ;
3186      ; BEGIN
3187      ;   Write to TSSR to soft initialize
3188      ;   Do a WRITE CHARACTERISTICS to setup a message buffer
3189      ;   Do a Write Subsystem Write Misc to Reset FIFO
3190      ;   Do a Write Subsystem READ STATUS
3191      ;   If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
3192      ;   signals NOT=0 Then Print Error
3193      ;   Do a Write Subsystem WRITE NPR to set tape direction out
3194      ;
3195      ; REPEAT FOR BYTE COUNT 1 TO 65.
3196      ; BEGIN
3197      ;   Do a Write Subsystem WRITE FIFO with the current byte count
3198      ;   Do a Write Subsystem Write Misc to Reset FIFO
3199      ;   Do a Write Subsystem READ STATUS
3200      ;   If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
3201      ;   signals NOT=0 Then Print Error
3202      ; END
3203      ;--
3204      045344      BGNSUB                      ;////////// BEGIN SUBTEST //////////
3205      045344      104402                      T6.6:      TRAP      C$BSUB
3206      ;
3207      045346      004737 015774      5$:      JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
3208      045352      103405      BCS      10$      ;BR IF SOFT INIT OKAY
3209      045354      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3210      045356      104455      ERDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
3211      045360      001201      TRAP      C$ERDF
3212      045362      003652      .WORD      641
3213      045364      012034      .WORD      SFIERR
3214      045366      005037 002222      10$:      Do a WRITE CHARACTERISTICS to setup a message buffer
3215      045372      012704 050250      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
3216      045376      004737 010662      MOV      @T17PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
3217      045416      103407      JSR      PC,WRTPCH      ;DO WRITE CHARACTERISTICS COMMAND
3218      045420      010001      FORCERROR 42$      ;GOODFORCE ERROR IF FORCER=1
3219      045422      045422      BCS      50$      ;BR IF CARRY SET (GOOD RETURN)
3220      045422      045422      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3221      045432      005237 002222      42$:      ERDF      ERRNO,T17SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
3222      045436      104406      TRAP      C$ERDF
3223      045440      004737 050054      INC      FATFLG      ;SET FATAL ERROR FLAG
3224      045444      012704 050420      50$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
3225      045450      010465 000000      TRAP      C$CLP1
3226      045454      004737 016336      ; Do a Write Subsystem Write Misc to Reset FIFO
3227      045460      103407      JSR      PC,T17RSFIF      ;SETUP PKT FOR WRITE MISC RESET FIFO
3228      045474      JSR      PC,T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
3229      045480      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
3230      045484      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
3231      045488      FORCERROR 62$      ;GOODFORCE ERROR IF FORCER=1
3232      045492      BCS      70$      ;BR IF CARRY SET (GOOD RETURN)

```

TEST 6: SUBTEST 6: FIFO VERIFY RESET FIFO TEST

```

3230 045476 010001      MOV     R0,R1      ;SAVE CONTENTS OF TSSR
3231 045500      NEXT,ERRNO
3232 045500      62$:  ERRDF  ERRNO,T172SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     643
                                .WORD     T172SSR
                                .WORD     PKTSSR
3233 045510 005237 002222      INC     FATFLG      ;SET FATAL ERROR FLAG
3234 045514      70$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
3235
3236      ; Do a Write Subsystem READ STATUS
3237      ; If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
3238      ; signals NOT=0 Then Print Error
3239 045516 004737 050034      JSR     PC,T17SRD      ;SETUP PACKET FOR READ STATUS
3240 045522 012704 050420      MOV     @T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
3241 045526 010465 000000      MOV     R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
3242 045532 004737 016336      JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
3243 045536      FORCERROR 77$      ;GOOD FORCE ERROR IF FORCER=1
3244 045552 103407      BCS     80$      ;BR IF CARRY SET (GOOD RETURN)
3245 045554 010001      MOV     R0,R1      ;SAVE CONTENTS OF TSSR
3246 045556      NEXT,ERRNO
3247 045556      77$:  ERRDF  ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     644
                                .WORD     T173SSR
                                .WORD     PKTSSR
3248 045566 005237 002222      INC     FATFLG      ;SET FATAL ERROR FLAG
3249 045572      80$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
3250 045574 004737 050216      JSR     PC,T17SETEXP      ;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
3251 045600 012701 046452      MOV     @T17EXSTA,R1      ;GET EXPECTED READ STATUS
3252 045604 012702 050312      MOV     @T17BFSTA,R2      ;GET RCV READ STATUS
3253 045610 011211      MOV     (R2),(R1)      ;SET EXPD WORD #8 = RCV TEMP
3254 045612 042711 002000      BIC     @S1.ICER,(R1)      ;SET EXPD ICER =0
3255 045616 042711 001000      BIC     @S1.IFMK,(R1)      ;SET EXPD IFMK =0
3256 045622 042711 000400      BIC     @S1.IHER,(R1)      ;SET EXPD IHER =0
3257 045626 016261 000002 000002      MOV     2(R2),2(R1)      ;SET EXPD WORD #9 = RCV (NOT TESTING)
3258 045634 005000      CLR     R0      ;HIGH RCV ADDRESS FOR CKMSG2
3259 045636 012701 050272      MOV     @T17BFR,R1      ;LOW RCV ADDRESS FOR CKMSG2
3260 045642 012702 046432      MOV     @T17EXP,R2      ;EXPD ADDRESS
3261 045646 012703 000024      MOV     @20.,R3      ;NUMBER OF BYTES TO COMPARE
3262 045652 004737 011500      JSR     PC,CKMSG2      ;EXPD EQUAL RCV?
3263 045656      FORCERROR 92$,NOTSSR      ;GOOD
3264 045666 103404      BCS     100$      ;BR IF YES
3265 045670      NEXT,ERRNO
3266 045670      92$:  ERRHRD ERRNO,T177CMP,MSGSTAT ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD     645
                                .WORD     T177CMP
                                .WORD     MSGSTAT
3267 045700      100$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
3268
3269      ; Do a Write Subsystem WRITE NPR to set tape direction out
3270 045702 012700 000100      MOV     @NP.OUT,R0      ;SET TAPE DIRECTION OUT
3271 045706 004737 050076      JSR     PC,T17SNPR      ;SETUP T17PK2 FOR WRITE NPR

```

TEST 6: SUBTEST 6: FIFO VERIFY RESET FIFO TEST

```

3272 045712 012704 050420      MOV      #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
3273 045716 010465 000000      MOV      R4,TSD8(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
3274 045722 004737 016336      JSR      PC,CHKTSSR    ;WAIT FOR SSR TO SET
3275 045726      FORCERROR      112$      ;DO FORCE ERROR IF FORCER=1
3276 045742 103407      BCS      120$      ;BR IF CARRY SET (GOOD RETURN)
3277 045744 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3278 045746      NEXT,ERRNO
3279 045746 112$: ERRDF      ERRNO,T174SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                     TRAP      C$ERDF
                                     .WORD      646
                                     .WORD      T174SSR
                                     .WORD      PKTSSR
                                     045746 104455
                                     045750 001206
                                     045752 047043
                                     045754 012046
3280 045756 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
3281 045762 120$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                     TRAP      C$CLP1
                                     045762 104406
3282
3283      ;      Setup incrementing pattern in FIFO data buffer
3284 045764 012701 046452      MOV      #T17EXSTA,R1    ;EXPD WRITE FIFO DATA BUFFER
3285 045770 012702 000100      MOV      #64.,R2      ;TEST PATTERN SIZE
3286 045774 005000      CLR      R0      ;INCREMENT TEST PATTERN
3287 045776 110021 130$: MOV8      R0,(R1)+    ;STORE INCREMENT TEST BYTE
3288 046000      INC      R0      ;SET NEXT PATTERN
3289 046002 005302      DEC      R2      ;DONE?
3290 046004 003374      BGT      130$      ;BR IF NO
3291
3292      ; REPEAT FOR BYTE COUNT 1 TO 65.
3293 046006 012737 000001 002310      MOV      #1,COUNT    ;GET FIRST BYTE COUNT
3294      ; Do a Write Subsystem WRITE FIFO with the current byte count
3295 046014 150$:      ;REPEAT LOOP LABEL
3296 046014 013700 002310      MOV      COUNT,R0    ;FIFO BYTE COUNT
3297 046020 012701 046452      MOV      #T17EXSTA,R1    ;FIFO WRITE DATA ADDRESS
3298 046024 004737 050122      JSR      PC,T17WFIF    ;SETUP T17PK2 FOR WRITE FIFO
3299 046030 012704 050420      MOV      #T17PK2,R4    ;GET WRITE SUBSYSTEM COMMAND PACKET
3300 046034 010465 000000      MOV      R4,TSD8(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
3301 046040 004737 016336      JSR      PC,CHKTSSR    ;WAIT FOR SSR TO SET
3302 046044      FORCERROR      152$      ;DO FORCE ERROR IF FORCER=1
3303 046060 103407      BCS      160$      ;BR IF CARRY SET (GOOD RETURN)
3304 046062 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3305 046064      NEXT,ERRNO
3306 046064 152$: ERRDF      ERRNO,T175SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                     TRAP      C$ERDF
                                     .WORD      647
                                     .WORD      T175SSR
                                     .WORD      PKTSSR
                                     046064 104455
                                     046066 001207
                                     046070 047106
                                     046072 012046
3307 046074 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
3308 046100 160$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                     TRAP      C$CLP1
                                     046100 104406
3309
3310      ; Do a Write Subsystem Write Misc to Reset FIFO
3311 046102 004737 050054      JSR      PC,T17RSFIF    ;SETUP PKT FOR WRITE MISC RESET FIFO
3312 046106 012704 050420      MOV      #T17PK2,R4    ;GET WRITE SUBSYSTEM COMMAND PACKET
3313 046112 010465 000000      MOV      R4,TSD8(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
3314 046116 004737 016336      JSR      PC,CHKTSSR    ;WAIT FOR SSR TO SET
3315 046122      FORCERROR      162$      ;DO FORCE ERROR IF FORCER=1
3316 046136 103407      BCS      170$      ;BR IF CARRY SET (GOOD RETURN)
3317 046140 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3318 046142      NEXT,ERRNO

```

TEST 6: SUBTEST 6: FIFO VERIFY RESET FIFO TEST

```

3319 046142      162$:  ERRDF  ERRNO,T172SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
      046142 104455                                     TRAP  C$ERDF
      046144 001210                                     .WORD 648
      046146 046732                                     .WORD T172SSR
      046150 012046                                     .WORD PKTSSR
3320 046152 005237 002222      INC  FATFLG  ;SET FATAL ERROR FLAG
3321 046156      170$:  CKLOOP  ;LOOP ON ERROR, IF FLAG SET
      046156 104406                                     TRAP  C$CLP1
3322
3323      ; Do a Write Subsystem READ STATUS
3324      ; If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
3325      ; signals NOT=0 Then Print Error
3326 046160 004737 050034      JSR  PC,T17SRD  ;SETUP PACKET FOR READ STATUS
3327 046164 012704 050420      MOV  #T17PK2,R4  ;GET WRITE SUBSYSTEM COMMAND PACKET
3328 046170 010465 000000      MOV  R4,TSD8(R5)  ;SET THE PACKET ADDRESS TO EXECUTE
3329 046174 004737 016336      JSR  PC,CHKTSSR  ;WAIT FOR SSR TO SET
3330 046200      FORCERROR 177$  ;GOODFORCE ERROR IF FORCER=1
3331 046214 103407      BCS 180$  ;BR IF CARRY SET (GOOD RETURN)
3332 046216 010001      MOV  R0,R1  ;SAVE CONTENTS OF TSSR
3333 046220      NEXT,ERRNO
3334 046220      177$:  ERRDF  ERRNO,T173SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
      046220 104455                                     TRAP  C$ERDF
      046222 001211                                     .WORD 649
      046224 046776                                     .WORD T173SSR
      046226 012046                                     .WORD PKTSSR
3335 046230 005237 002222      INC  FATFLG  ;SET FATAL ERROR FLAG
3336 046234      180$:  CKLOOP  ;LOOP ON ERROR, IF FLAG SET
      046234 104406                                     TRAP  C$CLP1
3337 046236 004737 050216      JSR  PC,T17SETEXP  ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
3338 046242 012701 046452      MOV  #T17EXSTA,R1  ;GET EXPECTED READ STATUS
3339 046246 012702 050312      MOV  #T17BFSTA,R2  ;GET RECV READ STATUS
3340 046252 011211      MOV  (R2),(R1)  ;SET EXPD WORD #8 = RECV TEMP
3341 046254 042711 002000      BIC  #S1.ICER,(R1)  ;SET EXPD ICER =0
3342 046260 042711 001000      BIC  #S1.IFMK,(R1)  ;SET EXPD IFMK =0
3343 046264 042711 000400      BIC  #S1.IHER,(R1)  ;SET EXPD IHER =0
3344 046270 016261 000002 000002      MOV  2(R2),2(R1)  ;SET EXPD WORD #9 = RECV (NOT TESTING)
3345 046276 005000      CLR  R0  ;HIGH RECV ADDRESS FOR CKMSG2
3346 046300 012701 050272      MOV  #T17BFR,R1  ;LOW RECV ADDRESS FOR CKMSG2
3347 046304 012702 046432      MOV  #T17EXP,R2  ;EXPD ADDRESS
3348 046310 012703 000024      MOV  #20.,R3  ;NUMBER OF BYTES TO COMPARE
3349 046314 004737 011500      JSR  PC,CKMSG2  ;EXPD EQUAL RECV?
3350 046320      FORCERROR 192$,NOTSSR  ;GOOD
3351 046330 103404      BCS 200$  ;BR IF YES
3352 046332      NEXT,ERRNO
3353 046332      192$:  ERRHRD  ERRNO,T177CMP,MSGSTAT  ;REPORT ERROR
      046332 104456                                     TRAP  C$ERHRD
      046334 001212                                     .WORD 650
      046336 047724                                     .WORD T177CMP
      046340 012350                                     .WORD MSGSTAT
3354 046342      200$:  CKLOOP  ;LOOP ON ERROR, IF FLAG SET
      046342 104406                                     TRAP  C$CLP1
3355
3356
3357 046344      250$:
3358 046344      FORCEEXIT 260$
3359 046354 005237 002310      INC  COUNT  ;GET NEXT BYTE COUNT
3360 046360 023727 002310 000101      CMP  COUNT,#65.  ;DONE ALL BYTES?

```

TEST 6: SUBTEST 6: FIFO VERIFY RESET FIFO TEST

```

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

```

TEST 6: SUBTEST 6: FIFO VERIFY RESET FIFO TEST

```

3414 047152      127      122      111 T176SSR: ASCIZ 'WRITE SUBSYSTEM (Read FIFO) Failed'
3415 047215      106      111      106 T171CMP: ASCIZ 'FIFO Status in WORD #9 Incorrect after Initialize'
3416 047277      122      145      141 T172CMP: ASCIZ 'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
3417 047373      106      111      106 T173CMP: ASCIZ 'FIFO Status (In WORD #9) Incorrect after WRITE FIFO'
3418 047457      106      111      106 T174CMP: ASCIZ 'FIFO Status (In WORD #9) Incorrect after READ FIFO'
3419 047542      122      145      141 T175CMP: ASCIZ 'Read FIFO Data not equal to Write FIFO Data'
3420 047616      106      111      106 T176CMP: ASCIZ 'FIFO Status (In WORD #9) Incorrect after READ FIFO from an Empty FIFO'
3421 047724      106      111      106 T177CMP: ASCIZ 'FIFO Status (In WORD #9) Incorrect after RESET FIFO'
3422                                     .EVEN
3423
3424
3425
3426
3427 050010
3428 050010
3429 050014      012701      050272
3430 050020      012702      000120
3431 050024      105021
3432 050026      005302
3433 050030      003375
3434 050032      000207
3435
3436
3437
3438
3439 050034
3440 050034      004737      050010
3441 050040      012700      050430
3442 050044      112720      000005
3443 050050      105010
3444 050052      000207
3445
3446
3447
3448
3449 050054
3450 050054      004737      050010
3451 050060      012700      050430
3452 050064      112720      000010
3453 050070      112710      000030
3454 050074      000207
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464 050076
3465 050076      004737      050010
3466 050102      012701      050430
3467 050106      112721      000011
3468 050112      052700      000020
3469 050116      110011
3470 050120      000207

;+
; CLEAR MESSAGE BUFFER
;-
T17CLRBUF:
    SAVREG
    MOV     #T17BFR,R1
    MOV     #T17BEND-T17BFR,R2
10$:   CLR    (R1)+
    DEC     R2
    BGT     10$
    RTS     PC

;+
; SETUP T17PK2 PACKET FOR READ STATUS
;-
T17SRD:
    JSR     PC,T17CLRBUF
    MOV     #T17DT2,R0
    MOV     #PW,RDSTATUS,(R0)+
    CLR     (R0)
    RTS     PC

;+
; SETUP T17PK2 PACKET FOR WRITE MISC RESET FIFO
;-
T17RSFIF:
    JSR     PC,T17CLRBUF
    MOV     #T17DT2,R0
    MOV     #PW,WMISC,(R0)+
    MOV     #MS,RSFIF:MS,RSTAP,(R0)
    RTS     PC

;+
; SETUP T17PK2 PACKET FOR WRITE NPR
; INPUT:
; R0 CONTAINS BSEL1 NPR DATA
; SETS NP,WRP SINCE IF 0 IT WRITES WRONG PARITY.
;-
T17SNPR:
    JSR     PC,T17CLRBUF
    MOV     #T17DT2,R1
    MOV     #PW,WNPR,(R1)+
    BIS     #NP,WRP,R0
    MOV     R0,(R1)
    RTS     PC

```

TEST 6: SUBTEST 6: FIFO VERIFY RESET FIFO TEST

```

3471
3472
3473      ;+
3474      ; SETUP T17PK2 PACKET FOR WRITE FIFO
3475      ;
3476      ; INPUT:
3477      ;      R0 CONTAINS BYTE COUNT
3478      ;      R1 CONTAINS DATA PATTERN BLOCK ADDRESS
3479      ;-
3479 050122 T17WFIF:
3480 050122      SAVREG
3481 050126 004737 050010      JSR      PC,T17CLRBUF      ;SAVE R1-R5 UNTIL NEXT RETURN
3482 050132 012702 050430      MOV      #T17DT2,R2      ;CLEAR MESSAGE BUFFER
3483 050136 112722 000004      MOV8     #PW.WFIFO,(R2)+      ;WRITE SUBSYSTEM DATA BUFFER
3484 050142 110022      MOV8     R0,(R2)+      ;STORE WRITE FIFO IN BSEL0
3485 050144 005022      CLR      (R2)+      ;STORE BYTE COUNT IN BSEL1
3486 050146 112122 10$:      MOV8     (R1)+,(R2)+      ;CLEAR SEL2 (UNUSED)
3487 050150 005300      DEC      R0      ;STORE DATA PATTERN BYTE
3488 050152 003375      BGT      10$      ;DONE ALL BYTES?
3489 050154 000207      RTS      PC      ;BR IF NO
3490
3491
3492      ;+
3493      ; SETUP T17PK2 PACKET FOR READ FIFO
3494      ;
3495      ; INPUT:
3496      ;      R0 CONTAINS SEL2 BYTE COUNT
3497      ;-
3497 050156 T17RFIF:
3498 050156 004737 050010      JSR      PC,T17CLRBUF      ;CLEAR MESSAGE BUFFER
3499 050162 012701 050430      MOV      #T17DT2,R1      ;WRITE SUBSYSTEM DATA BUFFER
3500 050166 112721 000003      MOV8     #PW.RFIFO,(R1)+      ;STORE READ FIFO IN BSEL0
3501 050172 110021      MOV8     R0,(R1)+      ;STORE BYTE COUNT IN BSEL1
3502 050174 060207      RTS      PC      ;RETURN
3503
3504      ;+
3505      ; CLEAR EXPECTED DATA MESSAGE BUFFER
3506      ;-
3506 050176 T17CLEXP:
3507 050176 012701 046432      MOV      #T17EXP,R1      ;GET EXPD ADDRESS
3508 050202 012700 000122      MOV      #T17EXEND-T17EXP,R0      ;GET EXPD SIZE
3509 050206 105021      CLR8     (R1)+      ;CLEAR A BYTE
3510 050210 005300 10$:      DEC      R0      ;DONE?
3511 050212 003375      BGT      10$      ;BR IF NO
3512 050214 000207      RTS      PC      ;RETURN
3513
3514
3515      ;+
3516      ; Set WORDS 0-7 of expd message buffer * to recv since not testing
3517      ;-
3517 050216 T17SETEXP:
3518 050216 012702 046432      MOV      #T17EXP,R2      ;GET EXPD
3519 050222 012703 050272      MOV      #T17BFR,R3      ;GET READ STATUS RECV BUFFER
3520 050226 012700 000010      MOV      #8.,R0      ;SET WORDS 0-7 EXP=RECV
3521 050232 012322 5$:      MOV8     (R3)+,(R2)+      ;SET EXPD=RECV
3522 050234 005300      DEC      R0      ;DONE WORDS 0-7 WORDS?
3523 050236 003375      BGT      5$      ;BR IF NO
3524 050240 000207      RTS      PC      ;RETURN
3525
3526      ;
3527      ; * < . + 10 > & 177770
3528
3529

```

814

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08126 PAGE 28-29

SEQ 0170

TEST 6: SUBTEST 6: FIFO VERIFY RESET FIFO TEST

```

3530      ;WRITE CHARACTERISTICS COMMAND PACKET
3531      ;
3532      050250      T17PACKET:      ;COMMAND PACKET FOR TEST
3533      050250      100004      ;WRITE CHARACTERISTICS COMMAND, WITH ACK
3534      050252      050260      ;ADDRESS OF CHARACTERISTICS BLOCK
3535      050254      000000      ;
3536      050256      000012      ;MINIMUM MESSAGE PACKET SIZE
3537
3538      050260      T17DATA:      ;CHARACTERISTICS DATA BLOCK
3539      050260      050272      ;ADDRESS OF MESSAGE BUFFER
3540      050262      000000      ;
3541      050264      000024      ;LENGTH OF MESSAGE BUFFER
3542      050266      000000      ;ESS,ENB,EAI,ERI
3543      050270      000000      ;EXTENDED FEATURES UNIT NO, ETC.
3544
3545
3546      ;MESSAGE BUFFER FOR ALL TEST 6 COMMANDS
3547
3548      050272      T17BFR:      ;BEGIN MESSAGE BUFFER
3549      050272      000000      ;MESSAGE TYPE
3550      050274      000000      ;DATA FIELD LENGTH
3551      050276      000000      ;RBPCR
3552      050300      000000      ;XST0
3553      050302      000000      ;XST1
3554      050304      000000      ;XST2
3555      050306      000000      ;XST3
3556      050310      000000      ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
3557      050312      T17BFSTA:      ;READ STATUS AND WRITE FIFO BUFFER
3558      050412      T17BEND:      ;END OF MESSAGE BUFFER
3559
3560      ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
3561      ;
3562      050420      ;...10>0177770
3563      050420      T17PK2:      ;WRITE SUBSYSTEM WITH ACK
3564      050422      100006      ;LOW ADDRESS OF DATA BLOCK
3565      050424      050430      ;HIGH ADDRESS OF DATA BLOCK
3566      050426      000000      ;MINIMUM MESSAGE PACKET SIZE
3567      050428      000012      ;
3568
3569      050430      T17DT2:      ;DATA BLOCK
3570      050430      000      ;BSEL0
3571      050431      000      ;BSEL1
3572      050432      000000      ;SEL2
3573      050434      000000      ;WRITE FIFO DATA OUTPUT BUFFER
3574
3575      050536      ;
3576      050536      ;
3577      050536      104401      ;
3578
3579      ;SBIT1 TEST 7: STATIC TRANSPORT BUS INTERFACE TEST
3580
3581      ;TEST DESCRIPTION:
3582
3583
3584      ;TEST STEPS:
3585
3586      ;REPEAT FOR LOOPCNT

```

C14

TSV5 - HARDWARE TESTS MACRO M1200 20 MAR 84 08:26 PAGE 28-30

SEQ 0171

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```

3587      | BEGIN
3588      | Write to TSSR register to soft initialize the controller
3589      | Do WRITE CHARACTERISTICS to check for Extended Features Switch
3590      | If Extended Features Hardware Switch Clear then:
3591      |   Do Write Subsystem Write Miscellaneous to Set Extended Features.
3592      | Do WRITE CHARACTERISTICS to select reserved unit 7
3593      | Do Write Subsystem READ STATUS
3594      | If any transport interface signals are asserted then Print Error
3595      | END
3596      |
3597      |
3598      |
3599 050540      RGNTST
3600 050540      |
3604 050540 012700 051246      MOV      #TST18ID,R0      |ASCII MESSAGE TO IDENTIFY TEST
3605 050544 004737 016510      JSR      PC,TSTSETUP      |DO INITIAL TEST SETUP
3606 050550 012737 000012 002216  MOV      #10,,LO(PCNT)      |PERFORM 10 ITERATIONS
3607 050556      T18LOOP:
3608      | Write to TSSR register to soft initialize the controller
3609 050556      |
3610 050556 004737 015774      JSR      PC,SOFINIT      |WRITE TO TSSR TO SOFT INITIALIZE
3611 050562 103405      BCS      101      |BR IF SOFT INIT OKAY
3612 050564 010001      MOV      R0,R1      |SAVE CONTENTS OF TSSR
3613 050566      ERDIF      ERRNO,SFIERR,SFIMSG      |DEVICE FATAL DURING INIT
3614      |
3615      |
3616 050576 005037 002222      CLR      FATFLG      |CLEAR FATAL ERROR FLAG
3617 050602 012704 051730      MOV      #T18PACKET,R4      |GET THE ADDRESS OF COMMAND PACKET
3618 050606 004737 010662      JSR      PC,WRTHCR      |DO WRITE CHARACTERISTICS COMMAND
3619 050612      FORCERROR      121      |DO FORCE ERROR IF FORCER=1
3620 050626 103407      BCS      151      |BR IF CARRY SET (GOOD RETURN)
3621 050630 010001      MOV      R0,R1      |SAVE CONTENTS OF TSSR
3622 050632      NEXT,ERRNO
3623 050632 121:      ERDIF      ERRNO,T18SSR,PKTSSR      |DEVICE FATAL SSR FAILED TO SET
3624      |
3625      |
3626      |
3627      |
3628      |
3629 050650 012701 051752      MOV      #T18BFR,R1      |MESSAGE BUFFER ADDRESS
3630 050654 032761 000200 000012  BIT      #X2,EXTF,XST2(R1)      |EXTENDED FEATURES SWITCH SET?
3631 050662 001026      BNE      301      |BR IF YES
3632 050664 004737 051576      JSR      PC,T18SMISC      |SETUP PACKET FOR WRITE MISCELLANEOUS
3633 050670 012704 052000      MOV      #T18PK2,R4      |GET WRITE SUBSYSTEM COMMAND PACKET
3634 050674 010465 000000      MOV      R4,TSD8(R5)      |SET THE PACKET ADDRESS TO EXECUTE
3635 050700 004737 016336      JSR      PC,CHKTSSR      |WAIT FOR SSR TO SET
3636 050704      FORCERROR      221      |DO FORCE ERROR IF FORCER=1
3637 050720 103407      BCS      301      |BR IF CARRY SET (GOOD RETURN)

```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```

3638 050722 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3639 050724      NEXT,ERRNO
3640 050724      221:  ERRDF  ERRNO,T182SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C1ERDF
                                .WORD     702
                                .WORD     T182SSR
                                .WORD     PKTSSR
3641 050734 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
3642 050740      301:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C1CLP1
3643
3644
3645      ; Do WRITE CHARACTERISTICS to select reserved unit 7
3646 050742 005037 002222      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
3647 050746 012704 051730      MOV      @T18PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
3648 050752 004737 010662      JSR      PC,WATCHR      ;DO WRITE CHARACTERISTICS COMMAND
3649 050756      FORCERROR 421      ;DO FORCE ERROR IF FORCER=1
3650 050772 103407      BCS      501      ;BR IF CARRY SET (GOOD RETURN)
3651 050774 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3652 050776      NEXT,ERRNO
3653 050776      421:  ERRDF  ERRNO,T18SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C1ERDF
                                .WORD     703
                                .WORD     T18SSR
                                .WORD     PKTSSR
3654 051006 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
3655 051012      501:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C1CLP1
3656
3657      ; Clear message buffer
3658 051014 012701 051752      MOV      @T18BFR,R1      ;GET MESSAGE BUFFER ADDRESS
3659 051020 013700 051744      MOV      T18DATA+4,R0      ;SIZE OF MESSAGE BUFFER IN BYTES
3660 051024 105021      601:  CLRB      (R1)+      ;CLEAR A BYTE
3661 051026 005300      DEC      R0      ;DONE?
3662 051030 003375      BGT      601      ;BR IF NO
3663      ; Do a Write Subsystem READ STATUS
3664 051032 004737 051556      JSR      PC,T18SRD      ;SETUP PACKET FOR READ STATUS
3665 051036 012704 052000      MOV      @T18PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
3666 051042 010465 000000      MOV      R4,T5DB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
3667 051046 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
3668 051052      FORCERROR 621      ;DO FORCE ERROR IF FORCER=1
3669 051066 103407      BCS      701      ;BR IF CARRY SET (GOOD RETURN)
3670 051070 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3671 051072      NEXT,ERRNO
3672 051072      621:  ERRDF  ERRNO,T183SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C1ERDF
                                .WORD     704
                                .WORD     T183SSR
                                .WORD     PKTSSR
3673 051102 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
3674 051106      701:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C1CLP1
3675
3676
3677      ; Set first 8 words of expd message buffer - to recv since not testing
3678      ; Set unused bits in Read Status expd equal rcvd
3679 051110 004737 051620      JSR      PC,T18SETEXP      ;SET SOME EXPD TO RECV

```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```

3680      ; If any transport interface signals are asserted then Print Error
3681 051114 005000      CLR      R0      ;HIGH RECV ADDRESS FOR CKMSG2
3682 051116 012701 051752  MOV     @T18BFR,R1 ;LOW RECV ADDRESS FOR CKMSG2
3683 051122 012702 051216  MOV     @T18EXP,R2 ;EXPD ADDRESS
3684 051126 012703 000012  MOV     @10.,R3 ;NUMBER OF WORDS TO COMPARE
3685 051132 004737 011500  JSR     PC,CKMSG2 ;EXPD EQUAL RECV?
3686 051136      FORCERROR 82$,NOTSSR ;880
3687 051146 103404      BCS     90$ ;BR IF YES
3688 051150      NEXT,ERRNO
3689 051150 82$: ERRHRD ERRNO,T18CMP,MSGSTAT ;REPORT ERROR
      ;
      ;
      TRAP      C1ERRHRD
      .WORD     705
      .WORD     T18CMP
      .WORD     MSGSTAT
      ;
3690 051160 90$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      TRAP      C1CLP1
      ;
3691      ;
3692 051162 005737 002222  TST     FATFIG ;ANY FATAL ERRORS ?
3693 051166 001402      BEQ     160$ ;BRANCH IF NOT
3694 051170 004737 017202  JSR     PC,CKDROP ;TRY TO DROP THE UNIT
3695 051174 004737 016456  JSR     PC,TSTLOOP ;DO ITERATIONS?
3696 051200 103002      BCC     165$ ;BR IF NO
3697 051202 000137 050556  JMP     T18LOOP ;LOOP UNTIL ITERATIONS DONE
3698 051206      165$:
3699 051206      EXIT     TST
      ;
      ;
      TRAP      C1EXIT
      .WORD     L10065-.
3700
3701
3702
3703      ; LOCAL STORAGE FOR THIS TEST
3704
3705
3706 051212      T18MSK: ;MASK OF UNUSED BITS IN READ STATUS BYTES
3707 051212      .BYTE     +C<000> ;BYTE 0 MASK
3708 051213      .BYTE     +C<340> ;BYTE 1
3709 051214      .BYTE     +C<277> ;BYTE 2
3710 051215      .BYTE     0 ;MAKE IT EVEN
3711
3712 051216      T18EXP: ;EXPECTED DATA BUFFER
3713 051216      .WORD     0 ;MESSAGE TYPE
3714 051220      .WORD     0 ;DATA FIELD LENGTH
3715 051222      .WORD     0 ;RBPCH
3716 051224      .WORD     0 ;XST0
3717 051226      .WORD     0 ;XST1
3718 051230      .WORD     0 ;XST2
3719 051232      .WORD     0 ;XST3
3720 051234      .WORD     0 ;XST4 (ALWAYS PRESENT FOR WRITE SUB)
3721 051236      .WORD     0 ;READ STATUS BYTE 1/0
3722 051240      .WORD     0 ;READ STATUS BYTE 2
3723
3724 051242      377      020      T18XS: .BYTE     377,020 ;READ STATUS BYTE 0/1 EXPECTED BASE
3725 051244      000000      .WORD     0 ;READ STATUS BYTE 2 EXPECTED BASE
3726
3727
3728      ; LOCAL TEXT MESSAGES FOR TEST
3729

```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```

3730
3731 051246      123      164      141 TST18ID:      ,ASCIIZ 'Static Transport Bus Interface'
3732 051305      127      122      111 T18SSR: ,ASCIIZ 'WRITE CHARACTERISTICS Failed'
3733 051342      127      122      111 T182SSR: ,ASCIIZ 'WRITE SUBSYSTEM (Write Misc) Failed'
3734 051406      127      122      111 T183SSR: ,ASCIIZ 'WRITE SUBSYSTEM (Read Status) Failed'
3735 051453      124      162      141 T18CMP: ,ASCIIZ 'Transport Bus Interface Signals NOT Negotiated After Unit 7 Selected'
3736      ,EVEN
3737
3738
3739      ;*
3740      ; SETUP T18PK2 PACKET FOR READ STATUS
3741      ;
3742      T18SRD:
3743      SAVREG
3744      MOV      @T18DT2,R0      ;SAVE R1-R5 UNTIL NEXT RETURN
3745      MOV      @PW,RDSTATUS,(R0)+ ;WRITE SUBSYSTEM DATA BUFFER
3746      CLRB      (R0)      ;STORE READ STATUS COMMAND IN BSEL0
3747      RTS      PC      ;CLEAR BSEL1
3748      ;
3749      ;*
3750      ; SETUP T18PK2 PACKET FOR WRITE MISC.
3751      ;
3752      T18SMISC:
3753      SAVREG
3754      MOV      @T18DT2,R0      ;SAVE R1-R5 UNTIL NEXT RETURN
3755      MOV      @PW,WMISC,(R0)+ ;WRITE SUBSYSTEM DATA BUFFER
3756      MOV      @MS,EXT,(R0)+ ;STORE WRITE MISCELLANEOUS IN BSEL0
3757      RTS      PC      ;STORE INVERT EXTENDED FEATURES IN BSEL1
3758      ;
3759      ;*
3760      ; Set first 8 words of expd message buffer = to recv since not testing
3761      ; Set unused bits in Read Status expd equal rcvd
3762      ;
3763      T18SETEXP:
3764      MOV      @T18EXP,R2      ;GET EXPD
3765      MOV      @T18BFR,R3      ;GET READ STATUS RECV BUFFER
3766      MOV      @B,,R0      ;SET FIRST 8 WORDS EXP=RCV
3767      54:      MOV      (R3),,(R2)+ ;SET EXPD=RCV
3768      DEC      R0      ;DONE FIRST 8 WORDS?
3769      BGT      54      ;BR IF NO
3770      MOV      @T18MSK,R1      ;GET UNUSED BIT MASK
3771      MOV      T18XS,(R2)      ;SETUP BASE EXPECTED BYTE 1/0
3772      MOV      T18XS+2,2(R2) ;SETUP BASE EXPECTED BYTE 2
3773      MOV      (R3),R0      ;GET RECV BYTE 1 AND BYTE 0
3774      BIC      (R1),R0      ;CLEAR ALL BUT UNUSED
3775      BIC      R0,(R2)      ;CLEAR UNUSED IN EXP
3776      BIS      R0,(R2)      ;SET UNUSED EXPD=RCV FOR COMPARE
3777      MOV      2(R3),R0      ;GET RECV BYTE 2
3778      BIC      2(R1),R0      ;CLEAR ALL BUT UNUSED
3779      BIC      R0,2(R2)      ;CLEAR UNUSED IN EXPD
3780      BIS      R0,2(R2)      ;SET UNUSED EXPD=RCV FOR COMPARE
3781      CLRB      3(R2)      ;CLEAR EXPD BYTE 3 (UNUSED)
3782      CLRB      3(R3)      ;CLEAR RECV BYTE 3 (UNUSED)
3783      RTS      PC      ;RETURN
3784
3785      .<<.,+10>>0177770
3786
3787      ;
3788      ;WRITE CHARACTERISTICS COMMAND PACKET

```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```

3789
3790 051730
3791 051730 100004
3792 051732 051740
3793 051734 000000
3794 051736 000012
3795
3796 051740
3797 051740 051752
3798 051742 000000
3799 051744 000024
3800 051746 000000
3801 051750 000007
3802
3803
3804 051752
3805 051752 000000
3806 051754 000000
3807 051756 000000
3808 051760 000000
3809 051762 000000
3810 051764 000000
3811 051766 000000
3812 051770 000000
3813 051772 000000
3814 051774 000000
3815
3816
3817
3819 052000 052000
3821 052000
3822 052000 100006
3823 052002 052010
3824 052004 000000
3825 052006 000010
3826
3827 052010
3828 052010 000
3829 052011 000
3830 052012 000000
3831 052014 000000
3832
3833
3834 052016
3835 052016 104401
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845

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

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

T18BFR:
;WORD 0
;WORD 0
;WORD 0
;WORD 0
;WORD 0
;WORD 0
;WORD 0
;WORD 0
;WORD 0
;WORD 0
;MESSAGE BUFFER
;MESSAGE TYPE
;DATA FIELD LENGTH
;RBPCR
;XST0
;XST1
;XST2
;XST3
;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM)
;READ STATUS BYTE 1/0 RETURNED
;READ STATUS BYTE 2

;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
;
;.=<.10>&177770
T18PK2:
;WORD P.WRTSUB:P.ACK
;WORD T18DT2
;WORD 0
;WORD 8.
;WRITE SUBSYSTEM WITH ACK
;LOW ADDRESS OF DATA BLOCK
;HIGH ADDRESS OF DATA BLOCK
;BUFFER EXTENT

T18DT2:
;BYTE 0
;BYTE 0
;WORD 0
;WORD 0
;DATA BLOCK
;BSELO
;BSEL1
;SEL2
;DATA

ENDTST

L10065: TRAP C$ETST

.SBTTL TEST 8: TRANSPORT BUS INTERFACE LOOPBACK TEST

;
;TEST DESCRIPTION:
;
;This test verifies the controller's Transport Bus
;drivers, receivers, and signal loopback logic. Note
;that the static Transport Bus test must have run
;correctly for this test to provide meaningful results.
;
;TEST STEPS:
;

```

H14

SEQ 0176

TEST 8: TRANSPORT BUS INTERFACE LOOPBACK TEST

```

3846      REPEAT FOR LOOPCNT
3847      BEGIN
3848      Do Subtest 1      - Loopback Control signals test
3849      Do Subtest 2      - Loopback Read/Write signals test
3850      Do Subtest 3      - Loopback Write Strobe test
3851      Do Subtest 4      - Loopback Read Strobe test
3852      END
3853      --
3854
3855
3856      BGNTST
3857      052020
3858      052020
3859      052020 012700 060232      MOV      #TST19ID,RO      ;ASCII MESSAGE TO IDENTIFY TEST
3860      052024 004737 016510      JSR      PC,TSTSETUP      ;DO INITIAL TEST SETUP
3861      052030 012737 000012 002216  MOV      #10.,LOOPCNT      ;PERFORM 10 ITERATIONS
3862      052036      T19LOOP:
3863
3864      .SBTTL TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST
3865
3866      --
3867      TEST 8: SUBTEST 1:
3868
3869      SUBTEST DESCRIPTION:
3870
3871      This subtest verifies the Transport Control loopback
3872      path can transmit and receive correctly. The
3873      control signals are all loopback signals other
3874      than the read/write data (IW<7:0> and IR<7:0>).
3875
3876      TEST STEPS:
3877
3878      The loopback signals IFAD,ITADO,ITAD1 are the tape unit select
3879      lines. Since reserved unit 7 must remain selected these signals
3880      are always set low. This further means the signals they drive
3881      (ISPEED,IRDY,IONL) are only tested in the low state.
3882
3883      BEGIN
3884      Write to TSSR register to soft initialize the controller
3885      Do WRITE CHARACTERISTICS to check for Extended Features Switch
3886      If Extended Features Hardware Switch Clear then:
3887      Do Write Subsystem Write Miscellaneous to Set Extended Features.
3888      Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
3889      Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
3890      Do Write Subsystem Write Control to CLEAR loopback signals group 1.
3891      Do Write Subsystem Write Format to CLEAR loopback signals group 2.
3892      (the loopback signals have to be cleared here due to the flip-flops
3893      that are set on a 1 to 0 transition (IHER,IFMK,ICER))
3894      Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
3895      Do a Write Subsystem READ STATUS
3896      If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
3897      signals NOT=0 Then Print Error
3898
3899      REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
3900
3901      BEGIN
3902      Do Write Subsystem Write Control to Drive loopback signals group 1.
3903      Do Write Subsystem Write Format to Drive loopback signals group 2.
3904      Do a Write Subsystem READ STATUS
3905

```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

3906      ;      If loopback data NOT= data sent Then Print Error
3907      ;      Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
3908      ;      Do a Write Subsystem READ STATUS
3909      ;      If all Tape Status 2 (ICER,IFMK,IHER) Flip-Flop
3910      ;      signals NOT=0 Then Print Error
3911      ;      END
3912      ;--
3913 052036      BGNSUB                      ;////////// BEGIN SUBTEST //////////
      052036                      T8.1:      TRAP      C#BSUB
      052036      104402
3914
3915      ;      Write to TSSR register to soft initialize the controller
3916 052040      5$:      JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
3917 052040      004737      015774      BCS      10$      ;BR IF SOFT INIT OKAY
3918 052044      103405      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3919 052046      010001      ERDIF      ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
3920 052050      104455      TRAP      C#ERDF
      052050      001440      .WORD      800
      052054      003652      .WORD      SFIERR
      052056      012034      .WORD      SFIMSG
3921      ;      Do WRITE CHARACTERISTICS to check for Extended Features Switch
3922 052060      10$:      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
3923 052064      012704      062360      MOV      @T19PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
3924 052070      004737      010662      JSR      PC,WRITCHR ;DO WRITE CHARACTERISTICS COMMAND
3925 052074      103407      FORCERROR      12$      ;GOODFORCE ERROR IF FORCER=1
3926 052110      010001      BCS      15$      ;BR IF CARRY SET (GOOD RETURN)
3927 052112      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3928 052114      NEXT,ERRNO
3929 052114      12$:      ERDIF      ERRNO,T19SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      052114      104455      TRAP      C#ERDF
      052116      001441      .WORD      801
      052120      060273      .WORD      T19SSR
      052122      012046      .WORD      PKTSSR
3930 052124      005237      002222      INC      FATFLG      ;SET FATAL ERROR FLAG
3931 052130      15$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      052130      104406      TRAP      C#CLP1
3932      ;      If Extended Features Hardware Switch Clear then:
3933      ;      Do Write Subsystem Write Miscellaneous to Set Extended Features.
3934 052132      012701      062402      MOV      @T198FR,R1 ;MESSAGE BUFFER ADDRESS
3935 052136      032761      000200      000012      BIT      @X2,EXTF,XST2(R1) ;EXTENDED FEATURES SWITCH SET?
3936 052144      001026      BNE      30$      ;BR IF YES
3937 052146      004737      062232      JSR      PC,T19SEXT ;SETUP PACKET FOR WRITE MISC INVERT
3938 052152      012704      062530      MOV      @T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
3939 052156      010465      000000      MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
3940 052162      004737      016336      JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
3941 052166      103407      FORCERROR      22$      ;GOODFORCE ERROR IF FORCER=1
3942 052202      010001      BCS      30$      ;BR IF CARRY SET (GOOD RETURN)
3943 052204      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3944 052206      22$:      NEXT,ERRNO
      052206      104455      ERDIF      ERRNO,T192SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      052210      001442      TRAP      C#ERDF
      052212      060330      .WORD      802
      052214      012046      .WORD      T192SSR
      052216      005237      002222      .WORD      PKTSSR
3946 052216      30$:      INC      FATFLG      ;SET FATAL ERROR FLAG
3947 052222      CKLOOP      ;LOOP ON ERROR, IF FLAG SET

```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

052222 104406                                TRAP      C1CLP1
3948      ; Do WRITE CHARACTERISTICS to select reserved unit 7
3949 052224 005037 002222      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
3950 052230 012704 062360      MOV      @T19PACKET,R4    ;GET THE ADDRESS OF COMMAND PACKET
3951 052234 004737 010662      JSR      PC,WRITCHR      ;DO WRITE CHARACTERISTICS COMMAND
3952 052240      FORCERROR      42$      ;SSO FORCE ERROR IF FORCER=1
3953 052254 103407      BCS      50$      ;BR IF CARRY SET (GOOD RETURN)
3954 052256 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3955 052260      NEXT,ERRNO
3956 052260 42$:      ERDF      ERRNO,T19SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C1ERDF
                                .WORD      803
                                .WORD      T19SSR
                                .WORD      PKTSSR
                                052260 104455
                                052262 001443
                                052264 060273
                                052266 012046
3957 052270 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
3958 052274 50$:      CKLOOP      ;LOOP ON ERROR. IF FLAG SET
                                TRAP      C1CLP1
3959      ; Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
3960 052276 012700 000100      MOV      @NP.OUT,R0      ;SET TAPE DIRECTION OUT
3961 052302 052700 000040      BIS      @NP.LOOP,R0      ;SET LOOPBACK ENABLE
3962 052306 004737 062072      JSR      PC,T19SNPR      ;SETUP T19PK2 FOR WRITE NPR
3963 052312 012704 062530      MOV      @T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
3964 052316 010465 000000      MOV      R1,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
3965 052322 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
3966 052326      FORCERROR      62$      ;SSO FORCE ERROR IF FORCER=1
3967 052342 103407      BCS      70$      ;BR IF CARRY SET (GOOD RETURN)
3968 052344 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3969 052346      NEXT,ERRNO
3970 052346 62$:      ERDF      ERRNO,T194SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C1ERDF
                                .WORD      804
                                .WORD      T194SSR
                                .WORD      PKTSSR
                                052346 104455
                                052350 001444
                                052352 060441
                                052354 012046
3971 052356 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
3972 052362 70$:      CKLOOP      ;LOOP ON ERROR. IF FLAG SET
                                TRAP      C1CLP1
3973      ; Do Write Subsystem Write Control to CLEAR loopback signals group 1.
3974      ; Do Write Subsystem Write Format to CLEAR loopback signals group 2.
3975      ; (the loopback signals have to be cleared here due to the flip-flops
3976      ; that are set on a 1 to 0 transition (IHER,IFMK,ICER))
3977 052364 005000      CLR      R0      ;WRITE 0'S
3978 052366 042700 000200      BIC      @WC.IFAD,R0      ;IFAD MUST ALWAYS =0
3979 052372 042700 000100      BIC      @WC.IOTAD,R0      ;ITADO MUST ALWAYS =0
3980 052376 042700 000040      BIC      @WC.IITAD,R0      ;ITADI MUST ALWAYS =0
3981 052402 004737 062172      JSR      PC,T19WCTL      ;SETUP PACKET FOR WRITE CONTROL
3982 052406 012704 062530      MOV      @T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
3983 052412 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
3984 052416 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
3985 052422      FORCERROR      82$      ;SSO FORCE ERROR IF FORCER=1
3986 052436 103407      BCS      90$      ;BR IF CARRY SET (GOOD RETURN)
3987 052440 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3988 052442      NEXT,ERRNO
3989 052442 82$:      ERDF      ERRNO,T197SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C1ERDF
                                .WORD      805
                                .WORD      T197SSR
                                .WORD      PKTSSR
                                052442 104455
                                052444 001445
                                052446 060613
                                052450 012046

```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

3990	052452	005237	002222		INC	FATFLG		;	SET FATAL ERROR FLAG
3991	052456			90\$:	CKLOOP			;	LOOP ON ERROR, IF FLAG SET
	052456	104406							TRAP C\$CLP1
3992	052460	005000			CLR	R0		;	SET FORMAT DRIVE DATA=0
3993	052462	004737	062212		JSR	PC,T19WFMT		;	SETUP PACKET FOR WRITE FORMAT
3994	052466	012704	062530		MOV	0T19PK2,R4		;	GET WRITE SUBSYSTEM COMMAND PACKET
3995	052472	010465	000000		MOV	R4,TSD8(R5)		;	SET THE PACKET ADDRESS TO EXECUTE
3996	052476	004737	016336		JSR	PC,CHKTSSR		;	WAIT FOR SSR TO SET
3997	052502				FORCERROR	102\$		;	000FORCE ERROR IF FORCER=1
3998	052516	103407			BCS	110\$		;	BR IF CARRY SET (GOOD RETURN)
3999	052520	010001			MOV	R0,R1		;	SAVE CONTENTS OF TSSR
4000	052522				NEXT,ERRNO				
4001	052522			102\$:	ERRDF	ERRNO,T198SSR,PKTSSR		;	DEVICE FATAL SSR FAILED TO SET
	052522	104455							TRAP C\$ERDF
	052524	001446							.WORD 806
	052526	060662							.WORD T198SSR
	052530	012046							.WORD PKTSSR
4002	052532	005237	002222		INC	FATFLG		;	SET FATAL ERROR FLAG
4003	052536			110\$:	CKLOOP			;	LOOP ON ERROR, IF FLAG SET
	052536	104406							TRAP C\$CLP1
4004				;	Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS				
4005	052540	004737	062050		JSR	PC,T19R5FIF		;	SETUP PKT FOR WRITE MISC Reset Tape Status F-FLOPS
4006	052544	012704	062530		MOV	0T19PK2,R4		;	GET WRITE SUBSYSTEM COMMAND PACKET
4007	052550	010465	000000		MOV	R4,TSD8(R5)		;	SET THE PACKET ADDRESS TO EXECUTE
4008	052554	004737	016336		JSR	PC,CHKTSSR		;	WAIT FOR SSR TO SET
4009	052560				FORCERROR	122\$		;	000FORCE ERROR IF FORCER=1
4010	052574	103407			BCS	130\$		;	BR IF CARRY SET (GOOD RETURN)
4011	052576	010001			MOV	R0,R1		;	SAVE CONTENTS OF TSSR
4012	052600				NEXT,ERRNO				
4013	052600			122\$:	ERRDF	ERRNO,T192SSR,PKTSSR		;	DEVICE FATAL SSR FAILED TO SET
	052600	104455							TRAP C\$ERDF
	052602	001447							.WORD 807
	052604	060330							.WORD T192SSR
	052606	012046							.WORD PKTSSR
4014	052610	005237	002222		INC	FATFLG		;	SET FATAL ERROR FLAG
4015	052614			130\$:	CKLOOP			;	LOOP ON ERROR, IF FLAG SET
	052614	104406							TRAP C\$CLP1
4016				;	Do a Write Subsystem READ STATUS				
4017				;	If all Tape Status 2 (ICER,IFMK,IHER) flip-flop				
4018				;	signals NOT=0 Then Print Error				
4019	052616	004737	062030		JSR	PC,T19SRD		;	SETUP PACKET FOR READ STATUS
4020	052622	012704	062530		MOV	0T19PK2,R4		;	GET WRITE SUBSYSTEM COMMAND PACKET
4021	052626	010465	000000		MOV	R4,TSD8(R5)		;	SET THE PACKET ADDRESS TO EXECUTE
4022	052632	004737	016336		JSR	PC,CHKTSSR		;	WAIT FOR SSR TO SET
4023	052636				FORCERROR	132\$		;	000FORCE ERROR IF FORCER=1
4024	052652	103407			BCS	140\$		;	BR IF CARRY SET (GOOD RETURN)
4025	052654	010001			MOV	R0,R1		;	SAVE CONTENTS OF TSSR
4026	052656				NEXT,ERRNO				
4027	052656			132\$:	ERRDF	ERRNO,T193SSR,PKTSSR		;	DEVICE FATAL SSR FAILED TO SET
	052656	104455							TRAP C\$ERDF
	052660	001450							.WORD 808
	052662	060374							.WORD T193SSR
	052664	012046							.WORD PKTSSR
4028	052666	005237	002222		INC	FATFLG		;	SET FATAL ERROR FLAG
4029	052672			140\$:	CKLOOP			;	LOOP ON ERROR, IF FLAG SET
	052672	104406							TRAP C\$CLP1
4030	052674	004737	062270		JSR	PC,T19SETEXP		;	SET WORDS 0-7 EXPD=RECV (NOT TESTING)

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

4031 052700 012701 060132      MOV      #T19EXSTA,R1      ;GET EXPECTED READ STATUS
4032 052704 012702 062422      MOV      #T19BFSTA,R2      ;GET RECV READ STATUS
4033 052710 011211              MOV      (R2),(R1)      ;SET EXPD WORD #8 = RECV TEMP
4034 052712 042711 002000      BIC      #S1.ICER,(R1)      ;SET EXPD ICER =0
4035 052716 042711 001000      BIC      #S1.IFMK,(R1)      ;SET EXPD IFMK =0
4036 052722 042711 000400      BIC      #S1.IHER,(R1)      ;SET EXPD IHER =0
4037 052726 016261 000002 000002 MOV      2(R2),2(R1)      ;SET EXPD WORD #9 = RECV (NOT TESTING)
4038 052734 005000              CLR      R0      ;HIGH RECV ADDRESS FOR CKMSG2
4039 052736 012701 062402      MOV      #T19BFR,R1      ;LOW RECV ADDRESS FOR CKMSG2
4040 052742 012702 060112      MOV      #T19EXP,R2      ;EXPD ADDRESS
4041 052746 012703 000024      MOV      #20,R3      ;NUMBER OF BYTES TO COMPARE
4042 052752 004737 011500      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
4043 052756              FORCERROR 152$,NOTSSR      ;000
4044 052766 103404              BCS      160$      ;BR IF YES
4045 052770              NEXT.ERRNO
4046 052770 152$: ERRHRD ERRNO,T197CMP,MSGLOOP ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD     809
                                .WORD     T197CMP
                                .WORD     MSGLOOP
                                TRAP      C$CLP1
                                052770 104456
                                052772 001451
                                052774 061333
                                052776 013064
4047 053000 160$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                053000 104406      TRAP      C$CLP1
                                ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4048              CLR      T19PREV      ;INIT 1-0 TRANSITION FLAG
4049 053002 005037 060044      MOV      #TSTBLK,R3      ;GET FIRST PATTERN ADDRESS
4050 053006 012703 002752 200$: MOV      (R3)+,R0      ;GET A TEST PATTERN
4051 053012 012300              MOV      R3,TSTPTR      ;SAVE POINTER INTO TSTBLK
4052 053014 010337 002316      BIC      #WC.IFAD,R0      ;IFAD MUST ALWAYS =0
4053 053020 042700 000200      BIC      #WC.IOTAD,R0      ;ITADO MUST ALWAYS =0
4054 053024 042700 000100      BIC      #WC.IITAD,R0      ;IITAD1 MUST ALWAYS =0
4055 053030 042700 000040      MOV      R0,DATA      ;SET DATA PATTERN
4056 053034 010037 002312      ; Do Write Subsystem Write Control to Drive loopback signals group 1.
4057              ;000 CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
4058              MOV      DATA,R0      ;GET TEST PATTERN
4059 053040 013700 002312      JSR      PC,T19CNVT      ;CONVERT PATTERN TO CONTROL DRIVE MASK
4060 053044 004737 062314      ;R0 CONTAINS WRITE CONTROL DATA HERE
4061              ;SETUP PACKET FOR WRITE CONTROL
4062 053050 004737 062172      MOV      #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
4063 053054 012704 062530      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
4064 053060 010465 000000      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
4065 053064 004737 016336      FORCERROR 212$      ;000FORCE ERROR IF FORCER=1
4066 053070              BCS      220$      ;BR IF CARRY SET (GOOD RETURN)
4067 053104 103407              MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4068 053106 010001              NEXT.ERRNO
4069 053110 212$: ERDF ERRNO,T197SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
4070 053110 104455              TRAP      C$ERDF
4071 053110 053112 001452              .WORD     810
4072 053110 053114 060613              .WORD     T197SSR
4073 053110 053116 012046              .WORD     PKTSSR
4074 053120 005237 002222 220$: INC      FATFLG      ;SET FATAL ERROR FLAG
4075 053124 104406      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
4076 053126 013700 002312              TRAP      C$CLP1
4077 053132 004737 062314      ; Do Write Subsystem Write Format to Drive loopback signals group 2.
                                ;000 CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
                                MOV      DATA,R0      ;GET TEST PATTERN
                                JSR      PC,T19CNVT      ;CONVERT PATTERN TO FORMAT DRIVE MASK

```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

4078 053136 000300          SWAB      R0          ;WRITE FORMAT DATA RETURNED IN HIGH BYTE
4079 053140 004737 062212    JSR       PC,T19WFMT ;SETUP PACKET FOR WRITE FORMAT
4080 053144 012704 062530    MOV       #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
4081 053150 010465 000000    MOV       R4,TSD8(R5) ;SET THE PACKET ADDRESS TO EXECUTE
4082 053154 004737 016336    JSR       PC,CHKTSSR ;WAIT FOR SSR TO SET
4083 053160          FORCERROR 232$          ;DO FORCE ERROR IF FORCER=1
4084 053174 103407          BCS       240$ ;BR IF CARRY SET (GOOD RETURN)
4085 053176 010001          MOV       R0,R1 ;SAVE CONTENTS OF TSSR
4086 053200          NEXT.ERRNO
4087 053200 232$:          ERDF      ERRNO,T198SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     811
                                .WORD     T198SSR
                                .WORD     PKTSSR
                                053200 104455
                                053202 001453
                                053204 060662
                                053206 012046
4088 053210 005237 002222    INC       FATFLG ;SET FATAL ERROR FLAG
4089 053214 240$:          CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                053214 104406
4090          ; Do a Write Subsystem READ STATUS
4091 053216 004737 062030    JSR       PC,T19SRD ;SETUP PACKET FOR READ STATUS
4092 053222 012704 062530    MOV       #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
4093 053226 010465 000000    MOV       R4,TSD8(R5) ;SET THE PACKET ADDRESS TO EXECUTE
4094 053232 004737 016336    JSR       PC,CHKTSSR ;WAIT FOR SSR TO SET
4095 053236          FORCERROR 252$          ;DO FORCE ERROR IF FORCER=1
4096 053252 103407          BCS       260$ ;BR IF CARRY SET (GOOD RETURN)
4097 053254 010001          MOV       R0,R1 ;SAVE CONTENTS OF TSSR
4098 053256          NEXT.ERRNO
4099 053256 252$:          ERDF      ERRNO,T193SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     812
                                .WORD     T193SSR
                                .WORD     PKTSSR
                                053256 104455
                                053260 001454
                                053262 060374
                                053264 012046
4100 053266 005237 002222    INC       FATFLG ;SET FATAL ERROR FLAG
4101 053272 260$:          CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                053272 104406
4102          ; If loopback data NOT= data sent Then Print Error
4103 053274 004737 062270    JSR       PC,T19SETEXP ;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
4104 053300 012701 060132    MOV       #T19EXSTA,R1 ;GET EXPECTED READ STATUS
4105 053304 012702 062422    MOV       #T19BFSTA,R2 ;GET RCV READ STATUS
4106 053310 013711 002312    MOV       DATA,(R1) ;SET EXPD WORD #8 TO TEST DATA FIRST
4107 053314 013700 060044    MOV       T19PREV,R0 ;GET PREVIOUS DATA PATTERN
4108 053320 013703 002312    MOV       DATA,R3 ;GET CURRENT PATTERN
4109 053324 012704 000400    MOV       #S1.IHER,R4 ;SETUP IHER EXPECTED
4110 053330 040411          BIC       R4,(R1) ;SET EXPD IHER =0
4111 053332 030400          BIT       R4,R0 ;PREVIOUS =1?
4112 053334 001403          BEQ       275$ ;BR IF NO
4113 053336 030403          BIT       R4,R3 ;CURRENT =0?
4114 053340 001001          BNE       275$ ;BR IF NO
4115 053342 050411          BIS       R4,(R1) ;SET EXPD IHER =1
4116 053344 012704 001000 275$:          MOV       #S1.IFMK,R4 ;SETUP IFMK EXPECTED
4117 053350 040411          BIC       R4,(R1) ;SET EXPD IFMK =0
4118 053352 030400          BIT       R4,R0 ;PREVIOUS =1?
4119 053354 001403          BEQ       280$ ;BR IF NO
4120 053356 030403          BIT       R4,R3 ;CURRENT =0?
4121 053360 001001          BNE       280$ ;BR IF NO
4122 053362 050411          BIS       R4,(R1) ;SET EXPD IFMK =1
4123 053364 012704 002000 280$:          MOV       #S1.ICER,R4 ;SETUP ICER EXPECTED
4124 053370 040411          BIC       R4,(R1) ;SET EXPD ICER =0

```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

4125 053372 030400      BIT      R4,R0      ;PREVIOUS =1?
4126 053374 001403      BEQ      285$      ;BR IF NO
4127 053376 030403      BIT      R4,R3      ;CURRENT =0?
4128 053400 001001      BNE      285$      ;BR IF NO
4129 053402 050411      BIS      R4,(R1)    ;SET EXPD ICER =1
4130 053404 011100      285$: MOV      (R1),R0 ;GET EXPD WORD
4131      ; If previous IIDENT=1 and current is IIDENT=1 then EXPD= 0 else 1
4132 053406 012704 004000      MOV      $S1.IIDENT,R4 ;IIDENT
4133 053412 050400      BIS      R4,R0      ;ASSUME EXPD=1
4134 053414 030437 060044      BIT      R4,T19PREV ;PREVIOUS IIDENT=1?
4135 053420 001403      BEQ      288$      ;BR IF NO
4136 053422 030403      BIT      R4,R3      ;IS CURRENT IIDENT=1?
4137 053424 001401      BEQ      288$      ;BR IF NO
4138 053426 040400      BIC      R4,R0      ;SET EXPD=0
4139 053430 052700 040000      288$: BIS      $S1.I2RES,R0 ;I2RESV2 EXPD ALWAYS=1
4140 053434 052700 020000      BIS      $S1.I1RES,R0 ;I1RESV1 EXPD ALWAYS=1
4141 053440 042700 100000      BIC      $S1.PARERR,R0 ;IGNORE PARERR
4142 053444 032712 100000      BIT      $S1.PARERR,(R2) ;IS PARERR SET IN RECV?
4143 053450 001402      BEQ      290$      ;BR IF NO
4144 053452 052700 100000      BIS      $S1.PARERR,R0 ;SET IN EXPD
4145 053456 010011      290$: MOV      R0,(R1) ;SETUP FINAL EXPD IN WORD #8
4146 053460 016261 000002 000002      MOV      2(R2),2(R1) ;SET EXPD WORD #9 = RECV (NOT TESTING)
4147 053466 005000      CLR      R0      ;HIGH RECV ADDRESS FOR CKMSG2
4148 053470 012701 062402      MOV      $T19BFR,R1 ;LOW RECV ADDRESS FOR CKMSG2
4149 053474 012702 060112      MOV      $T19EXP,R2 ;EXPD ADDRESS
4150 053500 012703 000024      MOV      $20.,R3 ;NUMBER OF BYTES TO COMPARE
4151 053504 004737 011500      JSR      PC,CK1SG2 ;EXPD EQUAL RECV?
4152 053510      FORCERROR 302$,NOTSSR ;880
4153 053520 103404      BCS      310$      ;BR IF YES
4154 053522      NEXT.ERRNO
4155 053522      302$: ERRHRD ERRNO,T198CMP,MSGLOOP ;REPORT ERROR
      TRAP      C$ERHRD
      .WORD      813
      .WORD      T198CMP
      .WORD      MSGLOOP
4156 053532      310$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
      TRAP      C$CLP1
4157      ; Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
4158 053534 004737 062050      JSR      PC,T19RSFIF ;SETUP PKT FOR WRITE MISC Reset STATUS
4159 053540 012704 062530      MOV      $T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
4160 053544 010465 000000      MOV      R4,TSD8(R5) ;SET THE PACKET ADDRESS TO EXECUTE
4161 053550 004737 016336      JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
4162 053554      FORCERROR 322$ ;880FORCE ERROR IF FORCER=1
4163 053570 103407      BCS      330$      ;BR IF CARRY SET (GOOD RETURN)
4164 053572 010001      MOV      R0,R1 ;SAVE CONTENTS OF TSSR
4165 053574      NEXT.ERRNO
4166 053574      322$: ERDF ERRNO,T192SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      TRAP      C$ERDF
      .WORD      814
      .WORD      T192SSR
      .WORD      PKTSSR
4167 053604 005237 002222      330$: INC      FATELG ;SET FATAL ERROR FLAG
4168 053610      CKLOOP ;LOOP ON ERROR, IF FLAG SET
      TRAP      C$CLP1
4169      ; Do a Write Subsystem READ STATUS
4170 053612 004737 062030      JSR      PC,T19SRD ;SETUP PACKET FOR READ STATUS
4171 053616 012704 062530      MOV      $T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET

```

TEST A: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

4172	053622	010465	000000		MOV R4,TSD8(R5)	SET THE PACKET ADDRESS TO EXECUTE			
4173	053626	004737	016336		JSR PC,CHKTSSR	WAIT FOR SSR TO SET			
4174	053632				FORCERROR 3421	GOODFORCE ERROR IF FORCER=1			
4175	053646	103407			BCS 3501	BR IF CARRY SET (GOOD RETURN)			
4176	053650	010001			MOV RO,R1	SAVE CONTENTS OF TSSR			
4177	053652				NEXT.ERRNO				
4178	053652			3421:	ERRDF ERRNO,T193SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET			
	053652	104455					TRAP	C\$ERDF	
	053654	001457					.WORD	815	
	053656	060374					.WORD	T193SSR	
	053660	012046					.WORD	PKTSSR	
4179	053662	005237	002222		INC FATFLG	SET FATAL ERROR FLAG			
4180	053666			3501:	CKLOOP	LOOP ON ERROR, IF FLAG SET			
	053666	104406					TRAP	C\$CLP1	
4181	053670	004737	062270		JSR PC,T19SETEXP	SET WORDS 0-7 EXPD-RCV (NOT TESTING)			
4182	053674	012701	060132		MOV #T19EXSTA,R1	GET EXPECTED READ STATUS			
4183	053700	012702	062422		MOV #T19BFSTA,R2	GET RCV READ STATUS			
4184	053704	011211			MOV (R2),(R1)	SET EXPD WORD #8 = RCV TEMP			
4185	053706	042711	002000		BIC #S1.ICER,(R1)	SET EXPD ICER =0			
4186	053712	042711	001000		BIC #S1.IFMK,(R1)	SET EXPD IFMK =0			
4187	053716	042711	000400		BIC #S1.IHER,(R1)	SET EXPD IHER =0			
4188	053722	016261	000002	000002	MOV 2(R2),2(R1)	SET EXPD WORD #9 = RCV (NOT TESTING)			
4189	053730	005000			CLR RO	HIGH RCV ADDRESS FOR CKMSG2			
4190	053732	012701	062402		MOV #T19BFR,R1	LOW RCV ADDRESS FOR CKMSG2			
4191	053736	012702	060112		MOV #T19EXP,R2	EXPD ADDRESS			
4192	053742	012703	000024		MOV #20.,R3	NUMBER OF BYTES TO COMPARE			
4193	053746	004737	011500		JSR PC,CKMSG2	EXPD EQUAL RCV?			
4194	053752				FORCERROR 3621,NOTSSR	END			
4195	053762	103404			BCS 3701	BR IF YES			
4196	053764				NEXT.ERRNO				
4197	053764			3621:	ERRHRD ERRNO,T197CMP,MSGSTAT	REPORT ERROR			
	053764	104456					TRAP	C\$ERHRD	
	053766	001460					.WORD	816	
	053770	061333					.WORD	T197CMP	
	053772	012350					.WORD	MSGSTAT	
4198	053774			3701:	CKLOOP	LOOP ON ERROR, IF FLAG SET			
	053774	104406					TRAP	C\$CLP1	
4199									
4200	053776	013737	002312	060044	MOV DATA,T19PREV	SETUP PREVIOUS DATA FOR EXPD CALC.			
4201	054004	013703	002316		MOV TSTPTR,R3	RESTORE CURRENT TSTBLK POINTER			
4202	054010	020327	003062		CHP R3,#TBLEND	END OF TSTBLK?			
4203	054014	103002			BHIS 4001	BR IF YES			
4204	054016	000137	053012		JMP 2001	DO NEXT TSTBLK PATTERN			
4205	054022			4001:					
4206									
4207	054022				ENDSUB	////////// END SUBTEST //////////			
	054022					L10067:			
	054022	104403					TRAP	C\$ESUB	
4208									
4209	054024	005737	002222		TST FATFLG	ANY FATAL ERRORS ?			
4210	054030	001402			BEQ 4601	BRANCH IF NOT			
4211	054032	004737	017202		JSR PC,CKDROP	TRY TO DROP THE UNIT			
4212	054036			4601:					
4213				</					

C15

TSV5 - HARDWARE TESTS MACRO M1200 20 MAR-84 08:26 PAGE 28-43

SEQ 0184

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253

4254 054036
054036
054036 104402
4255
4256 054040
4257 054040 004737 015774
4258 054044 103405
4259 054046 010001
4260 054050
054050 104455
054052 001460
054054 003652
054056 012034
4261
4262 054060 005037 002222
4263 054064 012704 062360
4264 054070 004737 010662
4265 054074
4266 054110 103407
4267 054112 010001

```

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

**
TEST 8: SUBTEST 2:
SUBTEST DESCRIPTION:
    This subtest verifies the Read/Write data loopback path.
    The Read/Write data signals are IR<7:0> and IW<7:0>
    respectively.
TEST STEPS:
    REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
    BEGIN
        Write to TSSR register to soft initialize the controller
        Do WRITE CHARACTERISTICS to check for Extended Features Switch
        If Extended Features Hardware Switch Clear then:
            Do Write Subsystem Write Miscellaneous to Set Extended Features.
        Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
        Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
        Do a WRITE NPR to set loopback and tape direction OUT
        Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
        Do a READ FIFO with tape direction OUT to load tape out write latch
        Do a WRITE NPR to set loopback and tape direction IN
        Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
        to strobe loopback data into FIFO.
        Do a READ FIFO with tape direction IN to read data
        If Data read from FIFO NOT= to Data sent Then Print Error
        Do a Write Subsystem READ STATUS
        If Input Ready NOT=1 Then Print Error
        If Output Ready NOT=0 Then Print Error
        If Data In Miss NOT=0 Then Print Error
    END
**
BGNSUB                                // BEGIN SUBTEST //
                                     TB.2:
                                     TRAP C1BSUB
    Write to TSSR register to soft initialize the controller
51:
    JSR    PC,SOFINIT                ;WRITE TO TSSR TO SOFT INITIALIZE
    BCS    101                      ;BR IF SOFT INIT OKAY
    MOV    R0,R1                    ;SAVE CONTENTS OF TSSR
    ERDIF  ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
                                     TRAP C1ERDIF
                                     .WORD 816
                                     .WORD SFIERR
                                     .WORD SFIMSG
101:
    Do WRITE CHARACTERISTICS to check for Extended Features Switch
    CLR    FATELG                    ;CLEAR FATAL ERROR FLAG
    MOV    @T19PACKET,R4             ;GET THE ADDRESS OF COMMAND PACKET
    JSR    PC,WRITCHR                ;DO WRITE CHARACTERISTICS COMMAND
    FORCERR 121                      ;DO FORCE ERROR IF FORCER=1
    BCS    151                      ;BR IF CARRY SET (GOOD RETURN)
    MOV    R0,R1                    ;SAVE CONTENTS OF TSSR

```

D15

TSV5 - HARDWARE TESTS MACRO M1200 20 MAR 84 08:26 PAGE 28-44

SEQ 0185

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

```

4268 054114
4269 054114
      054114 104455
      054116 001461
      054120 060273
      054122 012046
4270 054124 005237 002222
4271 054130
      054130 104406
4272
4273
4274 054132 012701 062402
4275 054136 032761 000200 000012
4276 054144 001026
4277 054146 004737 062232
4278 054152 012704 062530
4279 054156 010465 000000
4280 054162 004737 016336
4281 054166
4282 054202 103407
4283 054204 010001
4284 054206
4285 054206
      054206 104455
      054210 001462
      054212 060330
      054214 012046
4286 054216 005237 002222
4287 054222
      054222 104406
4288
4289 054224 012704 062360
4290 054230 004737 010662
4291 054234
4292 054250 103407
4293 054252 010001
4294 054254
4295 054254
      054254 104455
      054256 001463
      054260 060273
      054262 012046
4296 054264 005237 002222
4297 054270
      054270 104406
4298
4299
4300
4301 054272 012703 002752
4302 054276 012337 002312
4303 054302 042737 177400 002312
4304 054310 010337 002316
4305
4306 054314 012700 000100
4307 054320 052700 000040
4308 054324 004737 062072
4309 054330 012704 062530
      NEXT,ERRNO
121:  ERRDF  ERRNO,T195SR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                         TRAP  C1ERDF
                                         .WORD 817
                                         .WORD T195SR
                                         .WORD PKTSSR
      INC  FATFLG                      ;SET FATAL ERROR FLAG
151:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                         TRAP  C1CLP1
      ; If Extended Features Hardware Switch Clear then:
      ; Do Write Subsystem Write Miscellaneous to Set Extended Features.
      MOV  #T19BFR,R1                  ;MESSAGE BUFFER ADDRESS
      BIT  #X2.EXTF,XST2(R1)          ;EXTENDED FEATURES SWITCH SET?
      BNE  301                         ;BR IF YES
      JSR  PC,T19SEXT                  ;SETUP PACKET FOR WRITE MISC INVERT
      MOV  #T19PK2,R4                 ;GET WRITE SUBSYSTEM COMMAND PACKET
      MOV  R4,TSD8(R5)                 ;SET THE PACKET ADDRESS TO EXECUTE
      JSR  PC,CHKTSSR                  ;WAIT FOR SSR TO SET
      FORCERROR 221                    ;BDDFORCE ERROR IF FORCER=1
      BCS  301                         ;BR IF CARRY SET (GOOD RETURN)
      MOV  R0,R1                       ;SAVE CONTENTS OF TSSR
      NEXT,ERRNO
221:  ERRDF  ERRNO,T192SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                         TRAP  C1ERDF
                                         .WORD 818
                                         .WORD T192SSR
                                         .WORD PKTSSR
      INC  FATFLG                      ;SET FATAL ERROR FLAG
301:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                         TRAP  C1CLP1
      ; Do WRITE CHARACTERISTICS to select reserved unit 7
      MOV  #T19PACKET,R4               ;GET THE ADDRESS OF COMMAND PACKET
      JSR  PC,WRCHR                    ;DO WRITE CHARACTERISTICS COMMAND
      FORCERROR 421                    ;BDDFORCE ERROR IF FORCER=1
      BCS  501                         ;BR IF CARRY SET (GOOD RETURN)
      MOV  R0,R1                       ;SAVE CONTENTS OF TSSR
      NEXT,ERRNO
421:  ERRDF  ERRNO,T195SR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                         TRAP  C1ERDF
                                         .WORD 819
                                         .WORD T195SR
                                         .WORD PKTSSR
      INC  FATFLG                      ;SET FATAL ERROR FLAG
501:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                         TRAP  C1CLP1
      ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
      INC  #TSTBLK,R3                  ;GET FIRST PATTERN ADDRESS
1001: MOV  (R3)+,DATA                  ;GET A TEST PATTERN
      BIC  #C<377>,DATA                ;DATA IS BYTE
      MOV  R3,TSTPTR                   ;SETUP CURRENT TSTBLK POINTER
      ; Do a WRITE NPR to set loopback and tape direction OUT
      MOV  #NPR.OUT,R0                 ;SET TAPE DIRECTION OUT
      BIS  #NPR.LOOP,R0               ;SET LOOPBACK
      JSR  PC,T19SNPR                  ;SETUP T19PK2 FOR WRITE NPR
      MOV  #T19PK2,R4                 ;GET WRITE SUBSYSTEM COMMAND PACKET

```

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

4310	054334	010465	000000	MOV	R4,TSDB(R5)	;	SET THE PACKET ADDRESS TO EXECUTE
4311	054340	004737	016336	JSR	PC,CHKTSSR	;	WAIT FOR SSR TO SET
4312	054344			FORCERROR	102;	;	FORCE ERROR IF FORCER=1
4313	054360	103407		BCS	105;	;	BR IF CARRY SET (GOOD RETURN)
4314	054362	010001		MOV	R0,R1	;	SAVE CONTENTS OF TSSR
4315	054364			NEXT,ERRNO			
4316	054364			102;	ERRDF ERRNO,T194SSR,PKTSSR	;	DEVICE FATAL SSR FAILED TO SET
	054364	104455				TRAP	C1ERDF
	054366	001464				.WORD	820
	054370	060441				.WORD	T194SSR
	054372	012046				.WORD	PKTSSR
4317	054374	005237	002222	INC	FATFLG	;	SET FATAL ERROR FLAG
4318	054400			105;	CKLOOP	;	LOOP ON ERROR, IF FLAG SET
	054400	104406				TRAP	C1CLP1
4319				;	Do a WRITE FIFO with byte count equal to 1 and tape direction OUT		
4320	054402	012700	000001	MOV	#1,R0	;	WRITE 1 BYTE
4321	054406	012701	002312	MOV	#DATA,R1	;	FIFO WRITE DATA ADDRESS
4322	054412	004737	062136	JSR	PC,T19WFIF	;	SETUP T19PK2 FOR WRITE FIFO
4323	054416	012704	062530	MOV	#T19PK2,R4	;	GET WRITE SUBSYSTEM COMMAND PACKET
4324	054422	010465	000000	MOV	R4,TSDB(R5)	;	SET THE PACKET ADDRESS TO EXECUTE
4325	054426	004737	016336	JSR	PC,CHKTSSR	;	WAIT FOR SSR TO SET
4326	054432			FORCERROR	107;	;	FORCE ERROR IF FORCER=1
4327	054446	103407		BCS	110;	;	BR IF CARRY SET (GOOD RETURN)
4328	054450	010001		MOV	R0,R1	;	SAVE CONTENTS OF TSSR
4329	054452			NEXT,ERRNO			
4330	054452			107;	ERRDF ERRNO,T195SSR,PKTSSR	;	DEVICE FATAL SSR FAILED TO SET
	054452	104455				TRAP	C1ERDF
	054454	001465				.WORD	821
	054456	060504				.WORD	T195SSR
	054460	012046				.WORD	PKTSSR
4331	054462	005237	002222	INC	FATFLG	;	SET FATAL ERROR FLAG
4332	054466			110;	CKLOOP	;	LOOP ON ERROR, IF FLAG SET
	054466	104406				TRAP	C1CLP1
4333				;	Do a READ FIFO with tape direction OUT to load tape out write latch		
4334	054470	012700	000001	MOV	#1,R0	;	SET READ BYTE COUNT
4335	054474	004737	062116	JSR	PC,T19RFIF	;	SETUP T19PK2 FOR READ FIFO
4336	054500	012704	062530	MOV	#T19PK2,R4	;	GET WRITE SUBSYSTEM COMMAND PACKET
4337	054504	010465	000000	MOV	R4,TSDB(R5)	;	SET THE PACKET ADDRESS TO EXECUTE
4338	054510	004737	016336	JSR	PC,CHKTSSR	;	WAIT FOR SSR TO SET
4339	054514			FORCERROR	122;	;	FORCE ERROR IF FORCER=1
4340	054530	103407		BCS	130;	;	BR IF CARRY SET (GOOD RETURN)
4341	054532	010001		MOV	R0,R1	;	SAVE CONTENTS OF TSSR
4342	054534			NEXT,ERRNO			
4343	054534			122;	ERRDF ERRNO,T196SSR,PKTSSR	;	DEVICE FATAL SSR FAILED TO SET
	054534	104455				TRAP	C1ERDF
	054536	001466				.WORD	822
	054540	060550				.WORD	T196SSR
	054542	012046				.WORD	PKTSSR
4344	054544	005237	002222	INC	FATFLG	;	SET FATAL ERROR FLAG
4345	054550			130;	CKLOOP	;	LOOP ON ERROR, IF FLAG SET
	054550	104406				TRAP	C1CLP1
4346				;	Do a WRITE NPR to set loopback and tape direction IN		
4347	054552	005000		CLR	R0	;	CLR NP.OUT TO SET TAPE DIRECTION IN
4348	054554	052700	000040	BIS	#NP.LOOP,R0	;	SET LOOPBACK
4349	054560	004737	062072	JSR	PC,T19SNPR	;	SETUP T19PK2 FOR WRITE NPR
4350	054564	012704	062530	MOV	#T19PK2,R4	;	GET WRITE SUBSYSTEM COMMAND PACKET
4351	054570	010465	000000	MOV	R4,TSDB(R5)	;	SET THE PACKET ADDRESS TO EXECUTE

F15

TSVS - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 28-46

SEQ 0187

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

4352	054574	004737	016336	JSR	PC,CHKTSSR	!WAIT FOR SSR TO SET
4353	054600			FORCERROR	142!	!BDDFORCE ERROR IF FORCER=1
4354	054614	103407		BCS	150!	!BR IF CARRY SET (GOOD RETURN)
4355	054616	010001		MOV	R0,R1	!SAVE CONTENTS OF TSSR
4356	054620			NEXT,ERRNO		
4357	054620			142!:	ERRDF ERRNO,T194SSR,PKTSSR	!DEVICE FATAL SSR FAILED TO SET
	054620	104455				TRAP C!ERDF
	054622	001467				.WORD 823
	054624	060441				.WORD T194SSR
	054626	012046				.WORD PKTSSR
4358	054630	005237	002222	INC	FATFLG	!SET FATAL ERROR FLAG
4359	054634			150!:	CKLOOP	!LOOP ON ERROR, IF FLAG SET
	054634	104406				TRAP C!CLP1
4360				!	Do a WRITE FIFO with byte count equal to 1 and Tape direction IN	
4361	054636	012700	000001	MOV	#1,R0	!WRITE 1 BYTE
4362	054642	012701	002312	MOV	#DATA,R1	!FIFO WRITE DATA ADDRESS
4363	054646	004737	062136	JSR	PC,T19WFIF	!SETUP T19PK2 FOR WRITE FIFO
4364	054652	012704	062530	MOV	#T19PK2,R4	!GET WRITE SUBSYSTEM COMMAND PACKET
4365	054656	010465	000000	MOV	R4,TSD8(R5)	!SET THE PACKET ADDRESS TO EXECUTE
4366	054662	004737	016336	JSR	PC,CHKTSSR	!WAIT FOR SSR TO SET
4367	054666			FORCERROR	162!	!BDDFORCE ERROR IF FORCER=1
4368	054702	103407		BCS	170!	!BR IF CARRY SET (GOOD RETURN)
4369	054704	010001		MOV	R0,R1	!SAVE CONTENTS OF TSSR
4370	054706			NEXT,ERRNO		
4371	054706			162!:	ERRDF ERRNO,T195SSR,PKTSSR	!DEVICE FATAL SSR FAILED TO SET
	054706	104455				TRAP C!ERDF
	054710	001470				.WORD 824
	054712	060504				.WORD T195SSR
	054714	012046				.WORD PKTSSR
4372	054716	005237	002222	INC	FATFLG	!SET FATAL ERROR FLAG
4373	054722			170!:	CKLOOP	!LOOP ON ERROR, IF FLAG SET
	054722	104406				TRAP C!CLP1
4374				!	Do a READ FIFO with tape direction IN to read c a	
4375				!	If Data read from FIFO NOT= to Data sent Then Print Error	
4376	054724	012700	000001	MOV	#1,R0	!SET READ BYTE COUNT
4377	054730	004737	062116	JSR	PC,T19RFIF	!SETUP T19PK2 FOR READ FIFO
4378	054734	012704	062530	MOV	#T19PK2,R4	!GET WRITE SUBSYSTEM COMMAND PACKET
4379	054740	010465	000000	MOV	R4,TSD8(R5)	!SET THE PACKET ADDRESS TO EXECUTE
4380	054744	004737	016336	JSR	PC,CHKTSSR	!WAIT FOR SSR TO SET
4381	054750			FORCERROR	182!	!BDDFORCE ERROR IF FORCER=1
4382	054764	103407		BCS	190!	!BR IF CARRY SET (GOOD RETURN)
4383	054766	010001		MOV	R0,R1	!SAVE CONTENTS OF TSSR
4384	054770			NEXT,ERRNO		
4385	054770			182!:	ERRDF ERRNO,T196SSR,PKTSSR	!DEVICE FATAL SSR FAILED TO SET
	054770	104455				TRAP C!ERDF
	054772	001471				.WORD 825
	054774	060550				.WORD T196SSR
	054776	012046				.WORD PKTSSR
4386	055000	005237	002222	INC	FATFLG	!SET FATAL ERROR FLAG
4387	055004			190!:	CKLOOP	!LOOP ON ERROR, IF FLAG SET
	055004	104406				TRAP C!CLP1
4388	055006	004737	062270	JSR	PC,T19SETEXP	!SET WORDS 0-7 EXPD=RCV (NOT TESTING)
4389	055012	012701	060132	MOV	#1,9EXSTA,R1	!GET EXPECTED READ STATUS
4390	055016	012702	062422	MOV	#T198FSTA,R2	!GET RCV READ STATUS
4391	055022	013711	002312	MOV	DATA,(R1)	!SET EXPD WORD #8 = DATA
4392	055026	016261	000002	MOV	2(R2),2(R1)	!SET EXPD WORD #9 = RCV (NOT TESTING)
4393	055034	005000		CLR	R0	!HIGH RCV ADDRESS FOR CKMSG2

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

```

4394 055036 012701 062402      MOV      #T19BFR,R1      ;LOW RECV ADDRESS FOR CKMSG2
4395 055042 012702 060112      MOV      #T19EXP,R2     ;EXPD ADDRESS
4396 055046 012703 000022      MOV      #18.,R3      ;NUMBER OF BYTES TO COMPARE
4397 055052 004737 011500      JSR       PC,CKMSG2     ;EXPD EQUAL RECV?
4398 055056                FORCERROR 202$,NOTSSR ;###
4399 055066 103404                BCS      210$      ;BR IF YES
4400 055070                NEXT,ERRNO
4401 055070 202$: ERRHRD  ERRNO,T199CMP,MSGSUB ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD     826
                                .WORD     T199CMP
                                .WORD     MSGSUB
                                .WORD     C$CLP1
                                055070 104456
                                055072 001472
                                055074 061510
                                055076 013742
4402 055100 210$: CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                055100 104406
                                TRAP      C$CLP1
4403                ; Do a Write Subsystem READ STATUS
4404                ; If Input Ready NOT=1 Then Print Error
4405                ; If Output Ready NOT=0 Then Print Error
4406                ; If Data In Miss NOT=0 Then Print Error
4407 055102 004737 062030      JSR       PC,T19SRD     ;SETUP PACKET FOR READ STATUS
4408 055106 012704 062530      MOV      #T19PK2,R4     ;GET WRITE SUBSYSTEM COMMAND PACKET
4409 055112 010465 000000      MOV      #4,TSD8(R5)   ;SET THE PACKET ADDRESS TO EXECUTE
4410 055116 004737 016336      JSR       PC,CHKTSSR    ;WAIT FOR SSR TO SET
4411 055122                FORCERROR 212$      ;###FORCE ERROR IF FORCER=1
4412 055136 103407                BCS      220$      ;BR IF CARRY SET (GOOD RETURN)
4413 055140 010001                MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4414 055142                NEXT,ERRNO
4415 055142 212$: ERRDF  ERRNO,T193SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     827
                                .WORD     T193SSR
                                .WORD     PKTSSR
                                055142 104455
                                055144 001473
                                055146 060374
                                055150 012046
4416 055152 005237 002222      INC      FATEFLG     ;SET FATAL ERROR FLAG
4417 055156 220$: CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                055156 104406
                                TRAP      C$CLP1
4418 055160 004737 062270      JSR       PC,T19SETEXP  ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4419 055164 012701 060132      MOV      #T19EXSTA,R1 ;GET EXPECTED READ STATUS
4420 055170 012702 062422      MOV      #T19BFSTA,R2  ;GET RECV READ STATUS
4421 055174 012221                MOV      (R2),R1     ;SET EXPD WORD #8 = RECV TEMP
4422 055176 011211                MOV      (R2),R1     ;SET EXPD WORD #9 = RECV TEMP
4423 055200 052711 000020      BIS      #S2.INRDY,(R1) ;SET EXP INPUT READY= 1
4424 055204 042711 000040      BIC      #S2.OTRDY,(R1) ;SET EXP OUTPUT READY= 0
4425 055210 042711 000200      BIC      #S2.DIM,(R1) ;SET EXP DATA IN MISS = 0
4426 055214 005000      CLR      R0      ;HIGH RECV ADDRESS FOR CKMSG2
4427 055216 012701 062402      MOV      #T19BFR,R1   ;LOW RECV ADDRESS FOR CKMSG2
4428 055222 012702 060112      MOV      #T19EXP,R2   ;EXPD ADDRESS
4429 055226 012703 000024      MOV      #20.,R3      ;NUMBER OF BYTES TO COMPARE
4430 055232 004737 011500      JSR       PC,CKMSG2     ;EXPD EQUAL RECV?
4431 055236                FORCERROR 232$,NOTSSR ;###
4432 055246 103404                BCS      240$      ;BR IF YES
4433 055250                NEXT,ERRNO
4434 055250 232$: ERRHRD  ERRNO,T196CMP,MSGSTAT ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD     828
                                .WORD     T196CMP
                                .WORD     MSGSTAT
                                .WORD     C$CLP1
                                055250 104456
                                055252 001474
                                055254 061250
                                055256 012350
4435 055260 240$: CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                055260 104406
                                TRAP      C$CLP1

```

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

```

4436
4437
4438
4439 055262          FORCEEXIT          255$          ;000
4440 055272 013703 002316      MOV      TSTPTR,R3      ;RESTORE CURRENT TSTBLK POINTER
4441 055276 020327 003062      CMP      R3,0TBLEND      ;END OF TSTBLK?
4442 055302 103002          BHS      255$          ;BR IF YES
4443 055304 000137 054276      JMP      100$          ;DO ANOTHER TSTBLK PATTERN
4444 055310          255$;
4445
4446 055310          ENDSUB                      ;////////// END SUBTEST //////////
         055310                      L10070:
         055310 104403                      TRAP      C$ESUB
4447
4448 055312 005737 002222      TST      FATFLG          ;ANY FATAL ERRORS ?
4449 055316 001402          BEQ      260$          ;BRANCH IF NOT
4450 055320 004737 017202      JSR      PC,CKDROP      ;TRY TO DROP THE UNIT
4451 055324          260$;
4452
4453
4454          .SBTTL  TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489 055324
         055324

```

```

***
; TEST 8: SUBTEST 3:
; SUBTEST DESCRIPTION:
;
;   This subtest verifies the Write Strobe loopback path
;   can strobe data from the FIFO to the Data lines.
;   The signal IRESV3 drives IWSTR (write strobe) to write
;   data from the FIFO to the tape data out latch.
;
; TEST STEPS:
;
; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
; BEGIN
;   Write to TSSR register to soft initialize the controller
;   Do WRITE CHARACTERISTICS to check for Extended Features Switch
;   If Extended Features Hardware Switch Clear then:
;     Do Write Subsystem Write Miscellaneous to Set Extended Features.
;   Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
;   Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
;   Do a WRITE NPR to set loopback and tape direction OUT
;   Do a WRITE FORMAT to set IRESV3-->IWSTR = 1
;   Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
;   Do a WRITE FORMAT to set IRESV3-->IWSTR = 0 to load write data latch
;   Do a WRITE FORMAT to set IRESV3-->IWSTR = 1
;   Do a WRITE NPR to set loopback and tape direction IN
;   Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
;     to strobe loopback data into FIFO.
;   Do a READ FIFO with tape direction IN to read data
;   If Data read from FIFO NOT= to Data sent Then Print Error
; END
;--
; BGNSUB                      ;////////// BEGIN SUBTEST //////////
;                               T8.3:

```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

Address	Offset	PC	SR	DR	Instruction	Comment	Trap	Offset
4490	055324	104402				Write to TSSR register to soft initialize the controller	TRAP	C1BSUB
4491	055326							
4492	055326	004737	015774		JSR PC,SOFINIT	WRITE TO TSSR TO SOFT INITIALIZE		
4493	055332	103405			BCS 10	BR IF SOFT INIT OKAY		
4494	055334	010001			MOV R0,R1	SAVE CONTENTS OF TSSR		
4495	055336				ERRDF ERRNO,SFIERR,SFIMSG	DEVICE FATAL DURING INIT		
	055336	104455					TRAP	C1ERDF
	055340	001474					.WORD	828
	055342	003652					.WORD	SFIERR
	055344	012034					.WORD	SFIMSG
4496						Do WRITE CHARACTERISTICS to check for Extended Features Switch		
4497	055346	005037	002222		CLR FATFLG	CLEAR FATAL ERROR FLAG		
4498	055352	012704	062360		MOV #T19PACKET,R4	GET THE ADDRESS OF COMMAND PACKET		
4499	055356	004737	010662		JSR PC,WRCHRR	DO WRITE CHARACTERISTICS COMMAND		
4500	055362				FORCERROR 12	FORCE ERROR IF FORCER=1		
4501	055376	103407			BCS 15	BR IF CARRY SET (GOOD RETURN)		
4502	055400	010001			MOV R0,R1	SAVE CONTENTS OF TSSR		
4503	055402				NEXT,ERRNO			
4504	055402				ERRDF ERRNO,T19SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET		
	055402	104455					TRAP	C1ERDF
	055404	001475					.WORD	829
	055406	060273					.WORD	T19SSR
	055410	012046					.WORD	PKTSSR
4505	055412	005237	002222		INC FATFLG	SET FATAL ERROR FLAG		
4506	055416				CKLOOP	LOOP ON ERROR, IF FLAG SET		
	055416	104406					TRAP	C1CLP1
4507						If Extended Features Hardware Switch Clear then:		
4508						Do Write Subsystem Write Miscellaneous to Set Extended Features.		
4509	055420	012701	062402		MOV #T19BFR,R1	MESSAGE BUFFER ADDRESS		
4510	055424	032761	000200	000012	BIT #X2.EXTF,XST2(R1)	EXTENDED FEATURES SWITCH SET?		
4511	055432	001026			BNE 30	BR IF YES		
4512	055434	004737	062232		JSR PC,T19SEXT	SETUP PACKET FOR WRITE MISC INVERT		
4513	055440	012704	062530		MOV #T19PK2,R4	GET WRITE SUBSYSTEM COMMAND PACKET		
4514	055444	010465	000000		MOV R4,TSD8(R5)	SET THE PACKET ADDRESS TO EXECUTE		
4515	055450	001737	016336		JSR PC,CHKTSSR	WAIT FOR SSR TO SET		
4516	055454				FORCERROR 22	FORCE ERROR IF FORCER=1		
4517	055470	103407			BCS 30	BR IF CARRY SET (GOOD RETURN)		
4518	055472	010401			MOV R0,R1	SAVE CONTENTS OF TSSR		
4519	055474				NEXT,ERRNO			
4520	055474				ERRDF ERRNO,T192SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET		
	055474	104455					TRAP	C1ERDF
	055476	001476					.WORD	830
	055500	060430					.WORD	T192SSR
	055502	012046					.WORD	PKTSSR
4521	055504	005237	002222		INC FATFLG	SET FATAL ERROR FLAG		
4522	055510				CKLOOP	LOOP ON ERROR, IF FLAG SET		
	055510	104406					TRAP	C1CLP1
4523						Do WRITE CHARACTERISTICS to select reserved unit 7		
4524	055512	012704	062360		MOV #T19PACKET,R4	GET THE ADDRESS OF COMMAND PACKET		
4525	055516	004737	010662					

Address	Offset	Hex	Label	Operation	Comment	SSR
055544	001477					831
055546	060273					T19SSR
055550	012046					PKTSSR
4531	055552	005237	002222	INC FATFLG	SET FATAL ERROR FLAG	
4532	055556			504: CKLOOP	LOOP ON ERROR, IF FLAG SET	
055556	104406				TRAP	C4CLP1
4533						
4534						
4535	055560	012703	002752			
4536	055564	012337	002312	1004: MOV #TSTBLK,R3	GET FIRST PATTERN ADDRESS	
4537	055570	042737	177400	MOV (R3)+,DATA	GET A TEST PATTERN	
4538	055576	010337	002316	BIC #C<377>,DATA	DATA IS BYTE	
4539				MOV R3,TSTPTR	SETUP CURRENT TSTBLK POINTER	
4540	055602	012700	000100		Do a WRITE NPR to set loopback and tape direction OUT	
4541	055606	052700	000040	MOV #NPR,OUT,R0	SET TAPE DIRECTION OUT	
4542	055612	004737	062072	BIS #NPR.LOOP,R0	SET LOOPBACK	
4543	055616	012704	062530	JSR PC,T19SNPR	SETUP T19PK2 FOR WRITE NPR	
4544	055622	010465	000000	MOV #T19PK2,R4	GET WRITE SUBSYSTEM COMMAND PACKET	
4545	055626	004737	016336	MOV R4,TSDB(R5)	SET THE PACKET ADDRESS TO EXECUTE	
4546	055632			JSR PC,CHKTSSR	WAIT FOR SSR TO SET	
4547	055646	103407		FORCERROR 1024	FORCE ERROR IF FORCER=1	
4548	055650	010001		BCS 1054	BR IF CARRY SET (GOOD RETURN)	
4549	055652			MOV R0,R1	SAVE CONTENTS OF TSSR	
4550	055652			NEXT,ERRNO		
055652	104455			1024: ERRDF ERRNO,T194SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET	
055654	001500				TRAP	C4ERDF
055656	060441				WORD	832
055660	012046				WORD	T194SSR
4551	055662	005237	002222	INC FATFLG	SET FATAL ERROR FLAG	
4552	055666			1054: CKLOOP	LOOP ON ERROR, IF FLAG SET	
055666	104406				TRAP	C4CLP1
4553						
4554	055670	012700	000002		Do a WRITE FORMAT to set IRESV3-->IWSTR = 1	
4555	055674	004737	062212	MOV #WF.I3RES,R0	IRESV3-->IWSTR=1	
4556	055700	012704	062530	JSR PC,T19WFMT	SETUP T19PK2 FOR WRITE FORMAT	
4557	055704	010465	000000	MOV #T19PK2,R4	GET WRITE SUBSYSTEM COMMAND PACKET	
4558	055710	004737	016336	MOV R4,TSDB(R5)	SET THE PACKET ADDRESS TO EXECUTE	
4559	055714			JSR PC,CHKTSSR	WAIT FOR SSR TO SET	
4560	055730	103407		FORCERROR 1124	FORCE ERROR IF FORCER=1	
4561	055732	010001		BCS 1204	BR IF CARRY SET (GOOD RETURN)	
4562	055734			MOV R0,R1	SAVE CONTENTS OF TSSR	
4563	055734			NEXT,ERRNO		
055734	104455			1124: ERRDF ERRNO,T198SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET	
055736	001501				TRAP	C4ERDF
055740	060662				WORD	833
055742	012046				WORD	T198SSR
4564	055744	005237	002222	INC FATFLG	SET FATAL ERROR FLAG	
4565	055750			1204: CKLOOP	LOOP ON ERROR, IF FLAG SET	
055750	104406				TRAP	C4CLP1
4566						
4567	055752	012700	000001		Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT	
4568	055756	012701	002312	MOV #1,R0	WRITE 1 BYTE	
4569	055762	004737	062136	MOV #DATA,R1	FIFO WRITE DATA ADDRESS	
4570	055766	012704	062530	JSR PC,T19WFIF	SETUP T19PK2 FOR WRITE FIFO	
4571	055772	010465	000000	MOV #T19PK2,R4	GET WRITE SUBSYSTEM COMMAND PACKET	
4572	055776					

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```

4574 056016 103407      BCS      140$      ;BR IF CARRY SET (GOOD RETURN)
4575 056020 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4576 056022      NEXT,ERRNO
4577 056022 132$:      ERRDF      ERRNO,T195SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      834
                                .WORD      T195SSR
                                .WORD      PKTSSR
                                056022 104455
                                056024 001502
                                056026 060504
                                056030 012046
4578 056032 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
4579 056036 140$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                056036 104406
;      Do a WRITE FORMAT to set IRESV3-->IWSTR = 0
4580      CLR      R0      ;SET IRESV3-->IWSTR=0
4581 056040 005000      JSR      PC,T19WFMT      ;SETUP T9PK2 FOR WRITE FORMAT
4582 056042 004737 062212      MOV      @T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
4583 056046 012704 062530      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
4584 056052 010465 000000      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
4585 056056 004737 016336      FORCERROR      152$      ;DO FORCE ERROR IF FORCER=1
4586 056062      BCS      160$      ;BR IF CARRY SET (GOOD RETURN)
4587 056076 103407      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4588 056100 010001      NEXT,ERRNO
4589 056102 152$:      ERRDF      ERRNO,T198SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      835
                                .WORD      T198SSR
                                .WORD      PKTSSR
                                056102 104455
                                056104 001503
                                056106 060662
                                056110 012046
4591 056112 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
4592 056116 160$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                056116 104406
;      Do a WRITE FORMAT to set IRESV3 --> IWSTR = 1
4593      MOV      @WF,I3RES,R0      ;IRESV3-->IWSTR=1
4594 056120 012700 000002      JSR      PC,T19WFMT      ;SETUP T9PK2 FOR WRITE FORMAT
4595 056124 004737 062212      MOV      @T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
4596 056130 012704 062530      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
4597 056134 010465 000000      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
4598 056140 004737 016336      FORCERROR      172$      ;DO FORCE ERROR IF FORCER=1
4599 056144      BCS      180$      ;BR IF CARRY SET (GOOD RETURN)
4600 056160 103407      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4601 056162 010001      NEXT,ERRNO
4602 056164 172$:      ERRDF      ERRNO,T198SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      836
                                .WORD      T198SSR
                                .WORD      PKTSSR
                                056164 104455
                                056166 001504
                                056170 060662
                                056172 012046
4604 056174 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
4605 056200 180$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                056200 104406
4606      Do a WRITE NPR to set loopback and tape direction IN
4607      CLR      R0      ;CLR NP.OUT TO SET TAPE DIRECTION IN
4608 056202 005000      BIS      @NP,LOOP,R0      ;SET LOOPBACK
4609 056204 052700 000040      JSR      PC,T19SNPR      ;SETUP T19PK2 FOR WRITE NPR
4610 056210 004737 062072      MOV      @T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
4611 056214 012704 062530      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
4612 056220 010465 000000      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
4613 056224 004737 016336      FORCERROR      182$      ;DO FORCE ERROR IF FORCER=1
4614 056230      BCS      190$      ;BR IF CARRY SET (GOOD RETURN)
4615 056244 103407

```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```

4616 056246 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4617 056250      NEXT,ERRNO
4618 056250      182$:  ERRDF  ERRNO,T194SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                        TRAP      C$ERDF
                        .WORD      837
                        .WORD      T194SSR
                        .WORD      PKTSSR
                        056250 104455
                        056252 001505
                        056254 060441
                        056256 012046
4619 056260 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
4620 056264      190$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
4621      ; Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
4622 056266 012700 000001      MOV      #1,R0      ;WRITE 1 BYTE
4623 056272 012701 002312      MOV      #DATA,R1      ;WRITE DATA ADDRESS
4624 056276 004737 062136      JSR      PC,T19WFIF      ;SETUP T19PK2 FOR WRITE FIFO
4625 056302 012704 062530      MOV      #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
4626 056306 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
4627 056312 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
4628 056316      FORCERROR 202$      ;GOODFORCE ERROR IF FORCER=1
4629 056332 103407      BCS      210$      ;BR IF CARRY SET (GOOD RETURN)
4630 056334 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4631 056336      NEXT,ERRNO
4632 056336      202$:  ERRDF  ERRNO,T195SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                        TRAP      C$ERDF
                        .WORD      838
                        .WORD      T195SSR
                        .WORD      PKTSSR
                        056336 104455
                        056340 001506
                        056342 060504
                        056344 012046
4633 056346 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
4634 056352      210$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
4635      ; Do a READ FIFO with tape direction IN to read data
4636 056354 012700 000001      MOV      #1,R0      ;SET READ BYTE COUNT
4637 056360 004737 062116      JSR      PC,T19RFIF      ;SETUP T19PK2 FOR READ FIFO
4638 056364 012704 062530      MOV      #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
4639 056370 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
4640 056374 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
4641 056400      FORCERROR 222$      ;GOODFORCE ERROR IF FORCER=1
4642 056414 103407      BCS      230$      ;BR IF CARRY SET (GOOD RETURN)
4643 056416 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4644 056420      NEXT,ERRNO
4645 056420      222$:  ERRDF  ERRNO,T196SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                        TRAP      C$ERDF
                        .WORD      839
                        .WORD      T196SSR
                        .WORD      PKTSSR
                        056420 104455
                        056422 001507
                        056424 060550
                        056426 012046
4646 056430 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
4647 056434      230$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
4648      ; If Data read from FIFO NOT= to Data sent Then Print Error
4649 056436 004737 062270      JSR      PC,T19SETEXP      ;SET WORDS 0-7 EXPD-RECV (NOT TESTING)
4650 056442 012701 060132      MOV      #T19EXSTA,R1      ;GET EXPECTED READ STATUS
4651 056446 012702 062422      MOV      #T198FSTA,R2      ;GET RECV READ STATUS
4652 056452 013711 002312      MOV      DATA,(R1)      ;SET EXPD WORD #8 = DATA
4653 056456 016261 000002 000002      MOV      2(R2),2(R1)      ;SET EXPD WORD #9 = RECV (NOT TESTING)
4654 056464 005000      CLR      R0      ;HIGH RECV ADDRESS FOR CKMSG2
4655 056466 012701 062402      MOV      #T198FR,R1      ;LOW RECV ADDRESS FOR CKMSG2
4656 056472 012702 060112      MOV      #T19EXP,R2      ;EXPD ADDRESS
4657 056476 012703 000022      MOV      #18,,R3      ;NUMBER OF BYTES TO COMPARE

```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```

4658 056502 004737 011500      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
4659 056506                    FORCERROR 242$,NOTSSR      ;000
4660 056516 103404              BCS      250$              ;BR IF YES
4661 056520                    NEXT,ERRNO
4662 056520 242$:              ERRHRD  ERRNO,T19WSTR,MSGSUB ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD     840
                                .WORD     T19WSTR
                                .WORD     MSGSUB
4663 056530 250$:              CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
4664 056530 104406
4665
4666 056532                    FORCEEXIT 255$              ;000
4667 056542 013703 002316      MOV      TSTPTR,R3          ;RESTORE CURRENT TSTBLK POINTER
4668 056546 020327 003062      CMP      R3,0TBLND          ;END OF TSTBLK?
4669 056552 103002              BHIS     255$              ;BR IF YES
4670 056554 000137 055564      JMP      100$              ;DO ANOTHER TSTBLK PATTERN
4671 056560 255$:
4672
4673 056560                    ENDSUB                      ;////////// END SUBTEST //////////
                                L10071:
                                TRAP      C$ESUB
4674 056560 104403
4675 056562 005737 002222      TST      FATFLG              ;ANY FATAL ERRORS ?
4676 056566 001402              BEQ      260$              ;BRANCH IF NOT
4677 056570 004737 017202      JSR      PC,CKDROP          ;TRY TO DROP THE UNIT
4678 056574 260$:

```

```

.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 1STBLK 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

```

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

```

B16

TSV5 - HARDWARE TESTS MACRO M1200 20 MAR-84 08:26 PAGE 28-55

SEQ 0196

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

4749	056762	012704	062360	MOV	#T19PACKET,R4	GET THE ADDRESS OF COMMAND PACKET
4750	056766	004737	010662	JSR	PC,WRTCHR	DO WRITE CHARACTERISTICS COMMAND
4751	056772			FORCERROR	42#	FORCE ERROR IF FORCER=1
4752	057006	103407		BCS	50#	BR IF CARRY SET (GOOD RETURN)
4753	057010	010001		MOV	R0,R1	SAVE CONTENTS OF TSSR
4754	057012			NEXT	ERRNO	
4755	057012			42#:	ERRDF ERRNO,T19SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET
	057012	104455				TRAP C#ERDF
	057014	001513				.WORD 843
	057016	060273				.WORD T19SSR
	057020	012046				.WORD PKTSSR
4756	057022	005237	002222	INC	FATFLG	SET FATAL ERROR FLAG
4757	057026			50#:	CKLOOP	LOOP ON ERROR, IF FLAG SET
	057026	104406				TRAP C#CLP1
4758						
4759						
4760	057030	012703	002752			
4761	057034	012337	002312	100#:	MOV #TSTBLK,R3	GET FIRST PATTERN ADDRESS
4762	057040	042737	177400		MOV (R3),DATA	GET A TEST PATTERN
4763	057046	010337	002316		BIC #C<377>,DATA	DATA IS BYTE
4764					MOV R3,TSTPTR	SETUP CURRENT TSTBLK POINTER
4765	057052	012700	000100		Do a WRITE NPR to set loopback and tape direction OUT	
4766	057056	052700	000040		MOV #NP,OUT,R0	SET TAPE DIRECTION OUT
4767	057062	004737	062072		BIS #NP,LOOP,R0	SET LOOPBACK
4768	057066	012704	062530		JSR PC,T19SNPR	SETUP T19PK2 FOR WRITE NPR
4769	057072	010465	000000		MOV #T19PK2,R4	GET WRITE SUBSYSTEM COMMAND PACKET
4770	057076	004737	016336		MOV R4,TSD8(R5)	SET THE PACKET ADDRESS TO EXECUTE
4771	057102				JSR PC,CHKTSSR	WAIT FOR SSR TO SET
4772	057116	103407			FORCERROR 102#	FORCE ERROR IF FORCER=1
4773	057120	010001			BCS 105#	BR IF CARRY SET (GOOD RETURN)
4774	057122				MOV R0,R1	SAVE CONTENTS OF TSSR
4775	057122				NEXT,ERRNO	
	057122	104455		102#:	ERRDF ERRNO,T194SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET
	057124	001514				TRAP C#ERDF
	057126	060441				.WORD 844
	057130	012046				.WORD T194SSR
						.WORD PKTSSR
4776	057132	005237	002222		INC FATFLG	SET FATAL ERROR FLAG
4777	057136			105#:	CKLOOP	LOOP ON ERROR, IF FLAG SET
	057136	104406				TRAP C#CLP1
4778						
4779	057140	012700	000001		Do a WRITE FORMAT to set IRESV4-->IRSTR = 1	
4780	057144	004737	062212		MOV #WF,I4RES,R0	IRESV4-->IRSTR=1
4781	057150	012704	062530		JSR PC,T19WFM	SETUP T19PK2 FOR WRITE FORMAT
4782	057154	010465	000000		MOV #T19PK2,R4	GET WRITE SUBSYSTEM COMMAND PACKET
4783	057160	004737	016336		MOV R4,TSD8(R5)	SET THE PACKET ADDRESS TO EXECUTE
4784	057164				JSR PC,CHKTSSR	WAIT FOR SSR TO SET
4785	057200	103407			FORCERROR 112#	FORCE ERROR IF FORCER=1
4786	057202	010001			BCS 120#	BR IF CARRY SET (GOOD RETURN)
4787	057204				MOV R0,R1	SAVE CONTENTS OF TSSR
4788	057204				NEXT,ERRNO	
	057204	104455		112#:	ERRDF ERRNO,T198SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET
	057206	001515				TRAP C#ERDF
	057210	060662				.WORD 845
	057212	012046				.WORD T198SSR
						.WORD PKTSSR
4789	057214	005237	002222		INC FATFLG	SET FATAL ERROR FLAG
4790	057220			120#:	CKLOOP	LOOP ON ERROR, IF FLAG SET
	057220	104406				TRAP C#CLP1

TEST 8. SUBTEST 4 LOOPBACK READ STROBE TEST

Address	Offset	Hex	Dec	Label	Operation	Comment
4791	057222	012700	000001		MOV #1,R0	Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4793	057226	012701	002312		MOV #DATA,R1	WRITE 1 BYTE
4794	057232	004737	062136		JSR PC,T19WFIF	FIFO WRITE DATA ADDRESS
4795	057236	012704	062530		MOV #T19PK2,R4	SETUP T19PK2 FOR WRITE FIFO
4796	057242	010465	000000		MOV R4,TSD8(R5)	GET WRITE SUBSYSTEM COMMAND PACKET
4797	057246	004737	016336		JSR PC,CHKTSSR	SET THE PACKET ADDRESS TO EXECUTE
4798	057252				FORCERROR 1321	WAIT FOR SSR TO SET
4799	057266	103407			BCS 1401	FORCE ERROR IF FORCER=1
4800	057270	010001			MOV R0,R1	BR IF CARRY SET (GOOD RETURN)
4801	057272				NEXT,ERRNO	SAVE CONTENTS OF TSSR
4802	057272			1321:	ERRDF ERRNO,T195SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET
	057272	104455				TRAP C1ERDF
	057274	001516				.WORD 846
	057276	060504				.WORD T195SSR
	057300	012046				.WORD PKTSSR
4803	057302	005237	002222		INC FATFLG	SET FATAL ERROR FLAG
4804	057306			1401:	CKLOOP	LOOP ON ERROR, IF FLAG SET
	057306	104406				TRAP C1CLP1
4805						
4806	057310	012700	000001		MOV #1,R0	Do a READ FIFO with tape direction OUT to load tape out write latch
4807	057314	004737	062116		JSR PC,T19RFIF	SET READ BYTE COUNT
4808	057320	012704	062530		MOV #T19PK2,R4	SETUP T19PK2 FOR READ FIFO
4809	057324	010465	000000		MOV R4,TSD8(R5)	GET WRITE SUBSYSTEM COMMAND PACKET
4810	057330	004737	016336		JSR PC,CHKTSSR	SET THE PACKET ADDRESS TO EXECUTE
4811	057334				FORCERROR 1521	WAIT FOR SSR TO SET
4812	057350	103407			BCS 1601	FORCE ERROR IF FORCER=1
4813	057352	010001			MOV R0,R1	BR IF CARRY SET (GOOD RETURN)
4814	057354				NEXT,ERRNO	SAVE CONTENTS OF TSSR
4815	057354			1521:	ERRDF ERRNO,T196SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET
	057354	104455				TRAP C1ERDF
	057356	001517				.WORD 847
	057360	060550				.WORD T196SSR
	057362	012046				.WORD PKTSSR
4816	057364	005237	002222		INC FATFLG	SET FATAL ERROR FLAG
4817	057370			1601:	CKLOOP	LOOP ON ERROR, IF FLAG SET
	057370	104406				TRAP C1CLP1
4818						
4819	057372	005000			CLR R0	Do a WRITE NPR to set loopback and tape direction IN
4820	057374	052700	000040		BIS #NPR.LOOP,R0	CLR NP.OUT TO SET TAPE DIRECTION IN
4821	057400	004737	062072		JSR PC,T19SNPR	SET LOOPBACK
4822	057404	012704	062530		MOV #T19PK2,R4	SETUP T19PK2 FOR WRITE NPR
4823	057410	010465	000000		MOV R4,TSD8(R5)	GET WRITE SUBSYSTEM COMMAND PACKET
4824	057414	004737	016336		JSR PC,CHKTSSR	SET THE PACKET ADDRESS TO EXECUTE
4825	057420				FORCERROR 1821	WAIT FOR SSR TO SET
4826	057434	103407			BCS 1901	FORCE ERROR IF FORCER=1
4827	057436	010001			MOV R0,R1	BR IF CARRY SET (GOOD RETURN)
4828	057440				NEXT,ERRNO	SAVE CONTENTS OF TSSR
4829	057440			1821:	ERRDF ERRNO,T194SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET
	057440	104455				TRAP C1ERDF
	057442	001520				.WORD 848
	057444	060441				.WORD T194SSR
	057446	012046				.WORD PKTSSR
4830	057450	005237	002222		INC FATFLG	SET FATAL ERROR FLAG
4831	057454			1901:	CKLOOP	LOOP ON ERROR, IF FLAG SET
	057454	104406				TRAP C1CLP1
4832						
					Do a WRITE FORMAT to set IRESV4**>IRSTR * 0	

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

4833	057456	005000		CLR	R0		SET IRESV4==>IRSTR=0
4834	057460	004737	062212	JSR	PC,T19WFM		SETUP T9PK2 FOR WRITE FORMAT
4835	057464	012704	062530	MOV	#T19PK2,R4		GET WRITE SUBSYSTEM COMMAND PACKET
4836	057470	010465	000000	MC	R4,TSD8(R5)		SET THE PACKET ADDRESS TO EXECUTE
4837	057474	004737	016336	JSR	PC,CHKTSSR		WAIT FOR SSR TO SET
4838	057500			FORCERROR	202		FORCE ERROR IF FORCER=1
4839	057514	103407		BCS	210		BR IF CARRY SET (GOOD RETURN)
4840	057516	010001		MOV	R0,R1		SAVE CONTENTS OF TSSR
4841	057520			NEXT,ERRNO			
4842	057520			202:	ERRDF	ERRNO,T198SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET
	057520	104455					TRAP C#ERDF
	057522	001521					.WORD 849
	057524	060662					.WORD T198SSR
	057526	012046					.WORD PKTSSR
4843	057530	005237	002222	INC	FATFLG		SET FATAL ERROR FLAG
4844	057534			210:	CKLOOP		LOOP ON ERROR, IF FLAG SET
	057534	104406					TRAP C#CLP1
4845				1	Do a WRITE FORMAT to set IRESV4==>IRSTR = 1		
4846	057536	012700	000001	MOV	#WF,I4RES,R0		IRESV4==>IRSTR=1
4847	057542	004737	062212	JSR	PC,T19WFM		SETUP T9PK2 FOR WRITE FORMAT
4848	057546	012704	062530	MOV	#T19PK2,R4		GET WRITE SUBSYSTEM COMMAND PACKET
4849	057552	010465	000000	MOV	R4,TSD8(R5)		SET THE PACKET ADDRESS TO EXECUTE
4850	057556	004737	016336	JSR	PC,CHKTSSR		WAIT FOR SSR TO SET
4851	057562			FORCERROR	222		FORCE ERROR IF FORCER=1
4852	057576	103407		BCS	230		BR IF CARRY SET (GOOD RETURN)
4853	057600	010001		MOV	R0,R1		SAVE CONTENTS OF TSSR
4854	057602			NEXT,ERRNO			
4855	057602			222:	ERRDF	ERRNO,T198SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET
	057602	104455					TRAP C#ERDF
	057604	001522					.WORD 850
	057606	060662					.WORD T198SSR
	057610	012046					.WORD PKTSSR
4856	057612	005237	002222	INC	FATFLG		SET FATAL ERROR FLAG
4857	057616			230:	CKLOOP		LOOP ON ERROR, IF FLAG SET
	057616	104406					TRAP C#CLP1
4858				1	Do a READ FIFO with tape direction IN to read de		
4859	057620	012700	000001	MOV	#1,R0		SET READ BYTE C ANT
4860	057624	004737	062116	JSR	PC,T19RFIF		SETUP T19PK2 FOR READ FIFO
4861	057630	012704	062530	MOV	#T19PK2,R4		GET WRITE SUBSYSTEM COMMAND PACKET
4862	057634	010465	000000	MOV	R4,TSD8(R5)		SET THE PACKET ADDRESS TO EXECUTE
4863	057640	004737	016336	JSR	PC,CHKTSSR		WAIT FOR SSR TO SET
4864	057644			FORCERROR	282		FORCE ERROR IF FORCER=1
4865	057660	103407		BCS	290		BR IF CARRY SET (GOOD RETURN)
4866	057662	010001		MOV	R0,R1		SAVE CONTENTS OF TSSR
4867	057664			NEXT,ERRNO			
4868	057664			282:	ERRDF	ERRNO,T196SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET
	057664	104455					TRAP C#ERDF
	057666	001523					.WORD 851
	057670	060550					.WORD T196SSR
	057672	012046					.WORD PKTSSR
4869	057674	005237	002222	INC	FATFLG		SET FATAL ERROR FLAG
4870	057700			290:	CKLOOP		LOOP ON ERROR, IF FLAG SET
	057700	104406					TRAP C#CLP1
4871				1	If Data read from FIFO NOT= to Data sent Then Print Error		
4872	057702	004737	062270	JSR	PC,T19SETEXP		SET WORDS 0-7 EXPD=RCV (NOT TESTING)
4873	057706	012701	060152	MOV	#T19EXSTA,R1		GET EXPECTED READ STATUS
4874	057712	012702	062422	MOV	#T19BFSTA,R2		GET RCV READ STATUS

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

4875 057716 013711 002312      MOV      DATA,(R1)      ;SET EXPD WORD #8 = DATA
4876 057722 016261 000002 000002  MOV      2(R2),2(R1)    ;SET EXPD WORD #9 = RECV (NOT TESTING)
4877 057730 005000              CLR      RO              ;HIGH RECV ADDRESS FOR CKMSG2
4878 057732 012701 062402      MOV      #T198FR,R1      ;LOW RECV ADDRESS FOR CKMSG2
4879 057736 012702 060112      MOV      #T19EXP,R2      ;EXPD ADDRESS
4880 057742 012703 000022      MOV      #18,,R3      ;NUMBER OF BYTES TO COMPARE
4881 057746 004737 011500      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
4882 057752              FORCERROR 302$,NOTSSR ;###
4883 057762 103404              BCS      310$      ;BR IF YES
4884 057764              NEXT,ERRNO
4885 057764 302$:  ERRHRD  ERRNO,T19RSTR,MSGSUB ;REPORT ERROR
              TRAP      C$ERHRD
              .WORD      852
              .WORD      T19RSTR
              .WORD      MSGSUB
              SET
4886 057774 104406 310$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
              TRAP      C$CLP1
4887
4888
4889 057776              FORCEEXIT 355$
4890 060006 013703 002316      MOV      TSTPTR,R3      ;###
4891 060012 020327 003062      CMP      R3,#TBLEND    ;RESTORE CURRENT TSTBLK POINTER
4892 060016 103002              BHS      355$      ;END OF TSTBLK?
4893 060020 000137 057034      JMP      100$      ;BR IF YES
4894 060024 355$:              ;DO ANOTHER TSTBLK PATTERN
4895
4896 060024              ENDSUB      ;//////////////// END SUBTEST //////////////////
              L10072:
              TRAP      C$ESUB
4897 060024 104403
4898 060026 005737 002222      TST      FATFLG      ;ANY FATAL ERRORS ?
4899 060032 001402              BEQ      360$      ;BRANCH IF NOT
4900 060034 004737 017202      JSR      PC,CKDROP    ;TRY TO DROP THE UNIT
4901 060040 360$:
4902
4903 060040 104432              EXIT      TST      ;//////////////// EXIT TEST //////////////////
              TRAP      C$EXIT
              .WORD      L10066-
4904 060042 002602
4905
4906
4907 ;LOCAL STORAGE FOR THIS TEST
4908 ;
4909
4910 060044 000000      T19PREV:      .WORD      0      ;DRIVE SIGNAL 1-0 TRANSITION FLAG
4911
4912 ;
4913 ; LOOPBACK DRIVE SIGNAL TABLE
4914 ; THIS TABLE IS USED BY T19CNVI TO SETUP
4915 ; A DRIVE PATTERN FROM THE TEST DATA INPUT PATTERN.
4916 ;
4917 ; WRITE CONTROL SIGNALS ARE OF FORM WC,XXX
4918 ; WRITE FORMAT SIGNALS ARE OF FORM WF,XXXX
4919
4919 060046              T19BCTL:
4920 060046 000001      WC,IGN      ;WRITE CONTROL DRIVE SIGNALS
4921 060050 000002      WC,IFEN     ;IGO==>IFPT      DATA<0>
4922 060052 000004      WC,IRWU     ;IFEN==>IFBY      DATA<1>
              ;IRWU==>IRWD      DATA<2>

```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

4923 060054 000010 WC.IREW ;IREW==>IDBY DATA<3>
4924 060056 002000 WF.IERASE*256. ;IFAD==>ILOP DATA<4>
4925 060060 000040 WC.I1TAD ;ITAD1==>IONL DATA<5>
4926 060062 000100 WC.IOTAD ;ITAD0==>IRDY DATA<6>
4927 060064 000200 WC.IFAD ;IERASE==>ISPEED DATA<7>
4928 060066 004000 WF.IEDIT*256. ;IEDIT==>IHER DATA<8>
4929 060070 010000 WF.IWFM*256. ;IWFM==>IFMK DATA<9>
4930 060072 020000 WF.IREV*256. ;IREV==>ICER DATA<10>
4931 060074 040000 WF.IWRT*256. ;IWRT==>IIDENT DATA<11>
4932 060076 100000 WF.IHISP*256. ;IHISP==>IEOT DATA<12>
4933 060100 000000 .WORD 0 ;IRESV2 (UNUSED)DATA<13>
4934 060102 000000 .WORD 0 ;IRESV1 (UNUSED)DATA<14>
4935 060104 000000 .WORD 0 ;PARERR (UNTESTED)DATA<15>
4936
4937 060106 T19MSK: ;MASK OF UNTESTED BITS IN READ STATUS BYTES
4938 ;UNTESTED BITS ARE SET TO 1
4939 060106 377 .BYTE 1C<000> ;BYTE 0 MASK
4940 060107 037 .BYTE 1C<340> ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
4941 060110 360 .BYTE 1C<017> ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
4942 060111 000 .BYTE 0 ;MAKE IT EVEN
4943
4944 060112 T19EXP: ;BEGIN EXPECTED DATA BUFFER
4945 060112 000000 .WORD 0 ;MESSAGE TYPE
4946 060114 000000 .WORD 0 ;DATA FIELD LENGTH
4947 060116 000000 .WORD 0 ;RBPGR
4948 060120 000000 .WORD 0 ;XST0
4949 060122 000000 .WORD 0 ;XST1
4950 060124 000000 .WORD 0 ;XST2
4951 060126 000000 .WORD 0 ;XST3
4952 060130 000000 .WORD 0 ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
4953 060132 T19EXSTA: .BLKB 64. ;EXPECTED READ STATUS AND WRITE FIFO DATA
4954 060232 T19EXEND: ;END EXPECTED DATA BUFFER
4955
4956 ;
4957 ;LOCAL TEXT MESSAGES FOR TEST
4958 ;
4959 060232 124 162 141 T19ID: .ASCIZ 'Transport Bus Interface Loopback'
4960 060273 127 122 111 T19SSR: .ASCIZ 'WRITE CHARACTERISTICS Failed'
4961 060330 127 122 111 T192SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Misc) Failed'
4962 060374 127 122 111 T193SSR: .ASCIZ 'WRITE SUBSYSTEM (Read Status) Failed'
4963 060441 127 122 111 T194SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Npr) Failed'
4964 060504 127 122 111 T195SSR: .ASCIZ 'WRITE SUBSYSTEM (Write FIFO) Failed'
4965 060550 127 122 111 T196SSR: .ASCIZ 'WRITE SUBSYSTEM (Read FIFO) Failed'
4966 060613 127 122 111 T197SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Control) Failed'
4967 060662 127 122 111 T198SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Format) Failed'
4968 060730 106 111 106 T191CMP: .ASCIZ 'FIFO Status in WORD #9 Incorrect after Initialize'
4969 061012 122 145 141 T192CMP: .ASCIZ 'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
4970 061106 124 141 160 T193CMP: .ASCIZ 'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
4971 061174 122 145 141 T195CMP: .ASCIZ 'Read FIFO Data not equal to Write FIFO Data'
4972 061250 106 111 106 T196CMP: .ASCIZ 'FIFO Status (in WORD #9) Incorrect after READ FIFO'
4973 061333 124 141 160 T197CMP: .ASCIZ 'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
4974 061421 103 157 156 T198CMP: .ASCIZ 'Control Signal Loopback Data Error, Data is in WORD #8'
4975 061510 122 145 141 T199CMP: .ASCIZ 'Read/Write Loopback Data Error, Data is in WORD #8'
4976 061573 114 157 157 T19WSTR: .ASCIZ 'Loopback Data Error when strobed by Write strobe, Data is in WORD #8'
4977 061700 114 157 157 T19RSTR: .ASCIZ 'Loopback Data Error when strobed by Read Strobe, Data is in WORD #8'
4978
4979 .EVEN

```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

4980
4981
4982
4983
4984 062004
4985 062004
4986 062010 012701 062402
4987 062014 012702 000120
4988 062020 105021
4989 062022 005302
4990 062024 005375
4991 062026 000207
4992
4993
4994
4995
4996 062030
4997 062030 004737 062004
4998 062034 012700 062540
4999 062040 112720 000005
5000 062044 105010
5001 062046 000207
5002
5003
5004
5005
5006 062050
5007 062050 004737 062004
5008 062054 012700 062540
5009 062060 112720 000010
5010 062064 112710 000030
5011 062070 000207
5012
5013
5014
5015
5016
5017
5018
5019
5020
5021 062072
5022 062072 004737 062004
5023 062076 012701 062540
5024 062102 112721 000011
5025 062106 052700 000020
5026 062112 110011
5027 062114 000207
5028
5029
5030
5031
5032
5033
5034
5035 062116
5036 062116 004737 062004

;
; CLEAR MESSAGE BUFFER
;
T19CLRBUF:
    SAVREG
    MOV     @T19BFR,R1
    MOV     @T19BEND-T19BFR,R2
10$:   CLRB   (R1)+
    DEC     R2
    BGT     10$
    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
    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 T19PK2 PACKET FOR WRITE MISC Reset Tape Status F-FLOPS
;
T19RSFIF:
    JSR     PC,T19CLRBUF
    MOV     @T19DT2,R0
    MOVB    @PW,WMISC,(R0)+
    MOVB    @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
    MOVB    @PW,WNPR,(R1)+
    BIS     @NP,WRP,R0
    MOVB    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

5037	062122	012701	062540	MOV	T19DT2,R1	;WRITE SUBSYSTEM DATA BUFFER
5038	062126	112721	000003	MOVB	@PW.RFIFO,(R1)+	;STORE READ FIFO IN BSEL0
5039	062132	110021		MOVB	RO,(R1)+	;STORE BYTE COUNT IN BSEL1
5040	062130	000207		RTS	PC	;RETURN
5041				;*		
5042				; SETUP T19PK2 PACKET FOR WRITE FIFO		
5043				; INPUT:		
5044				; R0 CONTAINS BYTE COUNT		
5045				; R1 CONTAINS DATA PATTERN BLOCK ADDRESS		
5046				;*		
5047				T19WFIF:		
5048	062136			SAVREG		;SAVE R1-R5 UNTIL NEXT RETURN
5049	062136			JSR	PC,T19CLRBUF	;CLEAR MESSAGE BUFFER
5050	062142	004737	062004	MOV	@T19DT2,R2	;WRITE SUBSYSTEM DATA BUFFER
5051	062146	012702	062540	MOVB	@PW.WFIFO,(R2)+	;STORE WRITE FIFO IN BSEL0
5052	062152	112722	000004	MOVB	RO,(R2)+	;STORE BYTE COUNT IN BSEL1
5053	062156	110022		CLR	(R2)+	;CLEAR SEL2 (UNUSED)
5054	062160	005022		10\$:	MOVB (R1)+,(R2)+	;STORE DATA PATTERN BYTE
5055	062162	112122		DEC	RO	;DONE ALL BYTES?
5056	062164	005300		BGT	10\$	;BR IF NO
5057	062166	003375		RTS	PC	;RETURN
5058	062170	000207		;*		
5059				; SETUP T19PK2 FOR WRITE CONTROL		
5060				; INPUT:		
5061				; R0 CONTAINS DRIVING DATA PATTERN		
5062				;*		
5063				T19WCTL:		
5064				JSR	PC,T19CLRBUF	;CLEAR MESSAGE BUFFER
5065	062172	004737	062004	MOV	@T19DT2,R1	;WRITE SUBSYSTEM DATA BUFFER
5066	062176	012701	062540	MOVB	@PW.WCTL,(R1)+	;STORE WRITE CONTROL IN BSEL0
5067	062202	112721	000006	MOVB	RO,(R1)+	;STORE DATA WORD IN BSEL1
5068	062206	110021		RTS	PC	;RETURN
5069	062210	000207		;*		
5070				; SETUP T19PK2 FOR WRITE FORMAT TRANSPORT REGISTER		
5071				; INPUT:		
5072				; R0 CONTAINS DRIVING DATA PATTERN		
5073				;*		
5074				T19WFMT:		
5075				JSR	PC,T19CLRBUF	;CLEAR MESSAGE BUFFER
5076	062212			MOV	@T19DT2,R1	;WRITE SUBSYSTEM DATA BUFFER
5077	062212	004737	062004	MOVB	@PW.WFMT,(R1)+	;STORE WRITE FORMAT IN BSEL0
5078	062216	012701	062540	MOVB	RO,(R1)+	;STORE DATA WORD IN BSEL1
5079	062222	112721	000007	RTS	PC	;RETURN
5080	062226	110021		;*		
5081	062230	000207		; SETUP T19PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH		
5082				;*		
5083				T19SEXT:		
5084	062232			MOV	@T19DT2,RO	;WRITE SUBSYSTEM DATA BUFFER
5085	062232	012700	062540	MOVB	@PW.WMISC,(RO)+	;STORE WRITE MISCELLANEOUS IN BSEL0
5086	062236	112720	000010	MOVB	@MS.EXT,(RO)	;STORE INVERT EXTENDED FEATURES IN BSEL1
5087	062242	112710	000200	RTS	PC	;RETURN
5088	062246	000207		;*		
5089				; CLEAR EXPECTED DATA MESSAGE BUFFER		
5090				;*		
5091						
5092						
5093						

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

5094 062250
5095 062250 012701 060112
5096 062254 012700 000120
5097 062260 105021
5098 062262 005300
5099 062264 003375
5100 062266 000207
5101
5102
5103
5104
5105 062270
5106 062270 012702 060112
5107 062274 012703 062402
5108 062300 012700 000010
5109 062304 012322
5110 062306 005300
5111 062310 003375
5112 062312 000207
5113
5114
5115
5116
5117
5118
5119
5120
5121
5122
5123
5124
5125
5126
5127
5128
5129
5130 062314
5131 062314
5132 062320 012701 060046
5133 062324 005002
5134 062326 012703 000020
5135 062332 006000
5136 062334 103001
5137 062336 051102
5138 062340 005721
5139 062342 005303
5140 062344 003372
5141 062346 010200
5142 062350 000207
5143
5144
5145
5147 062360
5149
5150
5151
5152 062360

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 EXP=RECV
5$:    MOV    (R3)+,(R2)+ ;SET EXPD=RECV
        DEC    R0         ;DONE WORDS 0-7 WORDS?
        BGT    5$         ;BR IF NO
        RTS    PC         ;RETURN

;
; CONVERT A TEST PATTERN DATA WORD TO LOOPBACK DRIVE SIGNALS
;
; INPUTS:
;
;     R0      TEST PATTERN
;
; IMPLICIT INPUTS:
;
;     T19BCTL - CONTAINS WRITE CONTROL / WRITE FORMAT CONVERSION BITS
;
; OUTPUTS:
;
;     R0      - LOW BYTE CONTAINS WRITE CONTROL DATA
;              - HIGH BYTE CONTAINS WRITE FORMAT DATA
;
T19CNVT:
SAVREG
MOV    #T19BCTL,R1      ;SAVE R1-R5 UNTIL NEXT RETURN
CLR    R2               ;CONVERSION TABLE ADDRESS
MOV    #16,R3           ;INIT RESULT OF CONVERSION
10$:   ROR     R0         ;BIT COUNT
        BCC    20$        ;IS THIS BIT EQUAL TO 1?
        BIS    (R1),R2    ;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

```

5153 062360 100004      .WORD 100004      ;WRITE CHARACTERISTICS COMMAND, WITH ACK
5154 062362 062370      .WORD T19DATA     ;ADDRESS OF CHARACTERISTICS BLOCK
5155 062364 000000      .WORD 0          ;
5156 062366 000012      .WORD 10.         ;MINIMUM MESSAGE PACKET SIZE
5157
5158 062370              T19DATA:          ;CHARACTERISTICS DATA BLOCK
5159 062370 062402      .WORD T198FR      ;ADDRESS OF MESSAGE BUFFER
5160 062372 000000      .WORD 0          ;
5161 062374 000024      .WORD 20.         ;LENGTH OF MESSAGE BUFFER
5162 062376 000000      .WORD 0          ;ESS,ENB,EAI,ERI
5163 062400 000007      .WORD 7          ;EXTENDED FEATURES UNIT NO.
5164
5165
5166                      ;MESSAGE BUFFER FOR ALL TEST 8 COMMANDS
5167
5168 062402              T198FR:          ;BEGIN MESSAGE BUFFER
5169 062402 000000      .WORD 0          ;MESSAGE TYPE
5170 062404 000000      .WORD 0          ;DATA FIELD LENGTH
5171 062406 000000      .WORD 0          ;RBPCR
5172 062410 000000      .WORD 0          ;XST0
5173 062412 000000      .WORD 0          ;XST1
5174 062414 000000      .WORD 0          ;XST2
5175 062416 000000      .WORD 0          ;XST3
5176 062420 000000      .WORD 0          ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
5177 062422              T198FSTA: .BLKB 64. ;READ STATUS AND WRITE FIFO BUFFER
5178 062522              T198END:         ;END OF MESSAGE BUFFER
5179
5180                      ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
5181
5183                      ;* < . * 10 > E177770
5185 062530              T19PK2:          ;
5186 062530 100006      .WORD P.WRISUB!P.ACK ;WRITE SUBSYSTEM WITH ACK
5187 062532 062540      .WORD T19DT2      ;LOW ADDRESS OF DATA BLOCK
5188 062534 000000      .WORD 0          ;HIGH ADDRESS OF DATA BLOCK
5189 062536 000012      .WORD 10.         ;MINIMUM MESSAGE PACKET SIZE
5190
5191 062540              T19DT2:          ;DATA BLOCK
5192 062540 000          .BYTE 0          ;BSEL0
5193 062541 000          .BYTE 0          ;BSEL1
5194 062542 000000      .WORD 0          ;SEL2
5195 062544              .BLKB 64.         ;WRITE FIFO DATA OUTPUT BUFFER
5196
5197
5198 062644              ENDTST
5198 062644
5198 062644 104401      L10066:          TRAP C$ETST

```

.SBTTL TEST 9: READ/WRITE DATA PARITY TEST

```

***
; TEST DESCRIPTION:
;

```

```

; This test verifies that the Write Data Parity generator
; and the Read Data Parity checker operate properly. The
; Transport Bus signal loopback mode is enabled and a
; Set Wrong parity function is executed. Then various
; Write Subsystem Memory functions are performed to
; write data to and from the FIFO in loopback mode.
; The program then checks to insure a Read Data parity

```

TEST 9: READ/WRITE DATA PARITY TEST

```

5210      error occurred.
5211      A Reset FIFO is done and the Read Data parity
5212      error bit is again tested to insure it cleared.
5213      Finally a Clear wrong parity function is done
5214      and it is verified the data word can pass in loopback
5215      mode without setting Read Data parity error.
5216
5217      TEST STEPS:
5218
5219      REPEAT FOR LOOPCNT
5220      BEGIN
5221          Write to TSSR register to soft initialize the controller
5222          Do WRITE CHARACTERISTICS to check for Extended Features Switch
5223          If Extended Features Hardware Switch Clear then:
5224              Do Write Subsystem Write Miscellaneous to Set Extended Features.
5225          Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
5226      REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
5227      BEGIN
5228          (* Verify Write Wrong Parity Sets Parity Error *)
5229          Do a WRITE NPR to set loopback and tape direction OUT
5230              and SET Write Wrong Parity.
5231          Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
5232          Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
5233          Do a READ FIFO with tape direction OUT to load tape out write latch
5234              (this is when wrong parity (IWP) is set)
5235          Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 (sets read strobe low)
5236              (Read Strobe sets PAR IN H [Parity Error])
5237          Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
5238          Do a Write Subsystem READ STATUS
5239          If Read Data parity error NOT=1 Then Print Error
5240          Do a Write Misc to RESET FIFO
5241          Do a Write Subsystem READ STATUS
5242          If Read Data parity error NOT=0 Then Print Error
5243
5244          (* Verify Data can be transferred without a Parity Error *)
5245          Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
5246          Do a WRITE NPR to set loopback and tape direction OUT
5247              and CLEAR Write Wrong Parity.
5248          Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
5249          Do a READ FIFO with tape direction OUT to load tape out write latch
5250          Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 (sets read strobe low)
5251              (Read Strobe should NOT set PAR IN H [Parity Error] here)
5252          Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
5253          Do a Write Subsystem READ STATUS
5254          If Read Data parity error NOT=0 Then Print Error
5255
5256      END
5257  ---
5258
5259
5260      062646      BGNIST
5261      062646
5262      062646      012700      065232      MOV      @TST201D,RO
5263      062652      004737      016510      JSR      PC,TSTSETUP
5264      062656      012737      000012      002216      MOV      @10,,LOOPCNT
5265      062664      T20LOOP:
5266
5267      T9::
5268      ASCII MESSAGE TO IDENTIFY TEST
5269      DO INITIAL TEST SETUP
5270      PERFORM 10 ITERATIONS

```

TEST 9: READ/WRITE DATA PARITY TEST

```

5270 062664          BGNSUB                      // BEGIN SUBTEST //
      062664          T9.1:                      TRAP      C$BSUB
      062664 104402          Write to TSSR register to soft initialize the controller
5271          ;
5272 062666          5$:
5273 062666 004737 015774      JSR      PC,SOFINIT          ;WRITE TO TSSR TO SOFT INITIALIZE
5274 062672 103405          BCS      10$                  ;BR IF SOFT INIT OKAY
5275 062674 010001          MOV      R0,R1                  ;SAVE CONTENTS OF TSSR
5276 062676          ERRDF      ERRNO,SFIERR,SFIMSG        ;DEVICE FATAL DURING INIT
      062676 104455          TRAP      C$ERDF
      062700 001604          .WORD    900
      062702 003652          .WORD    SFIERR
      062704 012034          .WORD    SFIMSG
5277          ; Do WRITE CHARACTERISTICS to check for Extended Features Switch
5278 062706 005037 002222      10$: CLR      FATFLG          ;CLEAR FATAL ERROR FLAG
5279 062712 012704 066430      MOV      @T20PACKET,R4        ;GET THE ADDRESS OF COMMAND PACKET
5280 062716 004737 010662      JSR      PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
5281 062722          FORCERROR 12$          ;GOODFORCE ERROR IF FORCER=1
5282 062736 103407          BCS      15$                  ;BR IF CARRY SET (GOOD RETURN)
5283 062740 010001          MOV      R0,R1                  ;SAVE CONTENTS OF TSSR
5284 062742          NEXT,ERRNO
5285 062742          12$: ERRDF      ERRNO,T20SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      062742 104455          TRAP      C$ERDF
      062744 001605          .WORD    901
      062746 065261          .WORD    T20SSR
      062750 012046          .WORD    PKTSSR
5286 062752 005237 002222      INC      FATFLG          ;SET FATAL ERROR FLAG
5287 062756          15$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      062756 174406          TRAP      C$CLP1
5288          ; If Extended Features Hardware Switch Clear then:
5289          ; Do Write Subsystem Write Miscellaneous to Set Extended Features.
5290 062760 012701 066452      MOV      @T20BFR,R1          ;MESSAGE BUFFER ADDRESS
5291 062764 032761 000200 000012      BIT      @X2.EXIF,XST2(R1) ;EXTENDED FEATURES SWITCH SET?
5292 062772 001026          BNE      30$                  ;BR IF YES
5293 062774 004737 066346      JSR      PC,T20SEXT          ;SETUP PACKET FOR WRITE MISC INVERT
5294 063000 012704 066600      MOV      @T20PK2,R4          ;GET WRITE SUBSYSTEM COMMAND PACKET
5295 063004 010465 000000      MOV      R4,T5DB(R5)          ;SET THE PACKET ADDRESS TO EXECUTE
5296 063010 004737 016336      JSR      PC,CHKTSSR          ;WAIT FOR SSR TO SET
5297 063014          FORCERROR 22$          ;GOODFORCE ERROR IF FORCER=1
5298 063030 103407          BCS      30$                  ;BR IF CARRY SET (GOOD RETURN)
5299 063032 010001          MOV      R0,R1                  ;SAVE CONTENTS OF TSSR
5300 063034          NEXT,ERRNO
5301 063034          22$: ERRDF      ERRNO,T202SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      063034 104455          TRAP      C$ERDF
      063036 001606          .WORD    902
      063040 065316          .WORD    T202SSR
      063042 012046          .WORD    PKTSSR
5302 063044 005237 002222      INC      FATFLG          ;SET FATAL ERROR FLAG
5303 063050          30$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      063050 104406          TRAP      C$CLP1
5304          ; Do WRITE CHARACTERISTICS to select reserved unit 7
5305 063052 012704 066430      MOV      @T20PACKET,R4        ;GET THE ADDRESS OF COMMAND PACKET
5306 063056 004737 010662      JSR      PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
5307 063062          FORCERROR 42$          ;GOODFORCE ERROR IF FORCER=1
5308 063076 103407          BCS      50$                  ;BR IF CARRY SET (GOOD RETURN)
5309 063100 010001          MOV      R0,R1                  ;SAVE CONTENTS OF TSSR
5310 063102          NEXT,ERRNO

```

TEST 9: READ/WRITE DATA PARITY TEST

```

5311 063102          42$:  ERRDF  ERRNO,T20SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      063102 104455                                     TRAP C$ERDF
      063104 001607                                     .WORD 903
      063106 065261                                     .WORD T20SSR
      063110 012046                                     .WORD PKTSSR
5312 063112 005237 002222          50$:  INC  FATFLG      ;SET FATAL ERROR FLAG
5313 063116          50$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      063116 104406                                     TRAP C$CLP1
5314
5315
5316      ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
5317 063120 012703 002752          100$:  MOV  @TSTBLK,R3      ;GET FIRST PATTERN ADDRESS
5318 063124 012337 002312          100$:  MOV  (R3)+,DATA      ;GET A TEST PATTERN
5319 063130 042737 177400 002312          100$:  BIC  @C<377>,DATA      ;DATA IS BYTE
5320 063136 010337 002316          100$:  MOV  R3,TSTPTR      ;SETUP CURRENT TSTBLK POINTER
5321      ; Do a WRITE NPR to set loopback and tape direction OUT and
5322      ; and SET Write Wrong Parity.
5323 063142 012700 000100          100$:  MOV  @NP,OUT,R0      ;SET TAPE DIRECTION OUT
5324 063146 052700 000040          100$:  BIS  @NP,LOOP,R0      ;SET LOOPBACK
5325 063152 042700 000020          100$:  BIC  @NP,WRP,R0      ;SET WRITE WRONG PARITY (INVERTED)
5326 063156 004737 066216          100$:  JSR  PC,T20WNPR      ;SETUP T20PK2 FOR WRITE NPR
5327 063162 012704 066600          100$:  MOV  @T20PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
5328 063166 010465 000000          100$:  MOV  R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
5329 063172 004737 016336          100$:  JSR  PC,CHKTSSR      ;WAIT FOR SSR TO SET
5330 063176          100$:  FORCERROR 102$      ;FORCE ERROR IF FORCER=1
5331 063212 103407          100$:  BCS  105$      ;BR IF CARRY SET (GOOD RETURN)
5332 063214 010001          100$:  MOV  R0,R1      ;SAVE CONTENTS OF TSSR
5333 063216          100$:  NEXT,ERRNO
5334 063216          102$:  ERRDF  ERRNO,T204SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      063216 104455                                     TRAP C$ERDF
      063220 001610                                     .WORD 904
      063222 065427                                     .WORD T204SSR
      063224 012046                                     .WORD PKTSSR
5335 063226 005237 002222          105$:  INC  FATFLG      ;SET FATAL ERROR FLAG
5336 063232          105$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      063232 104406                                     TRAP C$CLP1
5337      ; Do a WRITE FORMAT to set IRESV4-->IRSTR = 1 (sets read strobe high)
5338 063234 012700 000001          105$:  MOV  @WF,I4RES,R0      ;IRESV4-->IRSTR = 1
5339 063240 004737 066312          105$:  JSR  PC,T20WFMT      ;SETUP T20PK2 FOR WRITE FORMAT
5340 063244 012704 066600          105$:  MOV  @T20PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
5341 063250 010465 000000          105$:  MOV  R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
5342 063254 004737 016336          105$:  JSR  PC,CHKTSSR      ;WAIT FOR SSR TO SET
5343 063260          105$:  FORCERROR 112$      ;FORCE ERROR IF FORCER=1
5344 063274 103407          105$:  BCS  120$      ;BR IF CARRY SET (GOOD RETURN)
5345 063276 010001          105$:  MOV  R0,R1      ;SAVE CONTENTS OF TSSR
5346 063300          105$:  NEXT,ERRNO
5347 063300          112$:  ERRDF  ERRNO,T208SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      063300 104455                                     TRAP C$ERDF
      063302 001611                                     .WORD 905
      063304 065601                                     .WORD T208SSR
      063306 012046                                     .WORD PKTSSR
5348 063310 005237 002222          120$:  INC  FATFLG      ;SET FATAL ERROR FLAG
5349 063314          120$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      063314 104406                                     TRAP C$CLP1
5350      ; Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
5351 063316 012700 000001          120$:  MOV  @1,R0      ;WRITE 1 BYTE
5352 063322 012701 002312          120$:  MOV  @DATA,R1      ;FIFO WRITE DATA ADDRESS

```

W WTD V7

....B1
....C1
....D1
....E1
....F1
....G1
....H1
....I1
....J1
....K1
....L1
....M1
....N1

....B5
....C5
....D5
....E5
....F5
....G5
....H5
....I5
....J5
....K5
....L5
....M5
....N5

....B9
....C9
....D9
....E9
....F9
....G9
....H9
....I9
....J9
....K9
....L9
....M9
....N9

....B13
....C13
....D13
....E13
....F13
....G13
....H13
....I13
....J13
....K13
....L13
....M13
....N13

....B2
....C2
....D2
....E2
....F2
....G2
....H2
....I2
....J2
....K2
....L2
....M2
....N2

....B6
....C6
....D6
....E6
....F6
....G6
....H6
....I6
....J6
....K6
....L6
....M6
....N6

....B10
....C10
....D10
....E10
....F10
....G10
....H10
....I10
....J10
....K10
....L10
....M10
....N10

....B14
....C14
....D14
....E14
....F14
....G14
....H14
....I14
....J14
....K14
....L14
....M14
....N14

....B3
....C3
....D3
....E3
....F3
....G3
....H3
....I3
....J3
....K3
....L3
....M3
....N3

....B7
....C7
....D7
....E7
....F7
....G7
....H7
....I7
....J7
....K7
....L7
....M7
....N7

....B11
....C11
....D11
....E11
....F11
....G11
....H11
....I11
....J11
....K11
....L11
....M11
....N11

....B15
....C15
....D15
....E15
....F15
....G15
....H15
....I15
....J15
....K15
....L15
....M15
....N15

....B4
....C4
....D4
....E4
....F4
....G4
....H4
....I4
....J4
....K4
....L4
....M4
....N4

....B8
....C8
....D8
....E8
....F8
....G8
....H8
....I8
....J8
....K8
....L8
....M8
....N8

....B12
....C12
....D12
....E12
....F12
....G12
....H12
....I12
....J12
....K12
....L12
....M12
....N12

....B16
....C16
....D16
....E16
....F16
....G16
....H16
....I16
....J16
....K16
....L16
....M16

TEST 9: READ/WRITE DATA PARITY TEST

5353	063326	004737	066256	JSR	PC,T20WFIF	SETUP T20PK2 FOR WRITE FIFO
5354	063332	012704	066600	MOV	#T20PK2,R4	GET WRITE SUBSYSTEM COMMAND PACKET
5355	063336	010465	000000	MOV	R4,TSD8(R5)	SET THE PACKET ADDRESS TO EXECUTE
5356	063342	004737	016336	JSR	PC,CHKTSSR	WAIT FOR SSR TO SET
5357	063346			FORCERROR	1524	FORCE ERROR IF FORCER=1
5358	063362	103407		BCS	1604	BR IF CARRY SET (GOOD RETURN)
5359	063364	010001		MOV	R0,R1	SAVE CONTENTS OF TSSR
5360	063366			NEXT,ERRNO		
5361	063366			1524:	ERRDF ERRNO,T205SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET
	063366	104455				TRAP C1ERDF
	063370	001612				.WORD 906
	063372	065472				.WORD T205SSR
	063374	012046				.WORD PKTSSR
5362	063376	005237	002222	INC	FATFLG	SET FATAL ERROR FLAG
5363	063402			1604:	CKLOOP	LOOP ON ERROR, IF FLAG SET
	063402	104406				TRAP C1CLP1
5364						
5365						Do a READ FIFO with tape direction OUT to load tape out write latch
						(this is when wrong parity (115) is set)
5366	063404	012700	000001	MOV	#1,R0	SET READ BYTE COUNT
5367	063410	004737	066236	JSR	PC,T20RFIF	SETUP T20PK2 FOR READ FIFO
5368	063414	012704	066600	MOV	#T20PK2,R4	GET WRITE SUBSYSTEM COMMAND PACKET
5369	063420	010465	000000	MOV	R4,TSD8(R5)	SET THE PACKET ADDRESS TO EXECUTE
5370	063424	004737	016336	JSR	PC,CHKTSSR	WAIT FOR SSR TO SET
5371	063430			FORCERROR	1724	FORCE ERROR IF FORCER=1
5372	063444	103407		BCS	1804	BR IF CARRY SET (GOOD RETURN)
5373	063446	010001		MOV	R0,R1	SAVE CONTENTS OF TSSR
5374	063450			NEXT,ERRNO		
5375	063450			1724:	ERRDF ERRNO,T206SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET
	063450	104455				TRAP C1ERDF
	063452	001613				.WORD 907
	063454	065536				.WORD T206SSR
	063456	012046				.WORD PKTSSR
5376	063460	005237	002222	INC	FATFLG	SET FATAL ERROR FLAG
5377	063464			1804:	CKLOOP	LOOP ON ERROR, IF FLAG SET
	063464	104406				TRAP C1CLP1
5378						
5379						Do a WRITE FORMAT to set IRESV4-->IRSTR = 0 (sets read strobe low)
						(Read Strobe sets PAR IN H (Parity Error))
5380	063466	005000		CLR	R0	IRESV4-->IRSTR = 0
5381	063470	004737	066312	JSR	PC,T20WFMT	SETUP T20PK2 FOR WRITE FORMAT
5382	063474	012704	066600	MOV	#T20PK2,R4	GET WRITE SUBSYSTEM COMMAND PACKET
5383	063500	010465	000000	MOV	R4,TSD8(R5)	SET THE PACKET ADDRESS TO EXECUTE
5384	063504	004737	016336	JSR	PC,CHKTSSR	WAIT FOR SSR TO SET
5385	063510			FORCERROR	1924	FORCE ERROR IF FORCER=1
5386	063524	103407		BCS	2004	BR IF CARRY SET (GOOD RETURN)
5387	063526	010001		MOV	R0,R1	SAVE CONTENTS OF TSSR
5388	063530			NEXT,ERRNO		
5389	063530			1924:	ERRDF ERRNO,T208SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET
	063530	104455				TRAP C1ERDF
	063532	001614				.WORD 908
	063534	065601				.WORD T208SSR
	063536	012046				.WORD PKTSSR
5390	063540	005237	002222	INC	FATFLG	SET FATAL ERROR FLAG
5391	063544			2004:	CKLOOP	LOOP ON ERROR, IF FLAG SET
	063544	104406				TRAP C1CLP1
5392						
						Do a WRITE FORMAT to set IRESV4-->IRSTR = 1 (sets read strobe high)
5393	063546	012700	000001	MOV	#WF,I4RES,R0	IRESV4-->IRSTR = 1
5394	063552	004737	066312	JSR	PC,T20WFMT	SETUP T20PK2 FOR WRITE FORMAT

Address	Offset	Hex Data	Assembly	Comments
5395	063556	012704 066600	MOV #T20PK2,R4	GET WRITE SUBSYSTEM COMMAND PACKET
5396	063562	010465 000000	MOV R4,TSD8(R5)	SET THE PACKET ADDRESS TO EXECUTE
5397	063566	004737 016336	JSR PC,CHKTSSR	WAIT FOR SSR TO SET
5398	063572		FORCERROR 212#	FORCE ERROR IF FORCER=1
5399	063606	103407	BCS 220#	BR IF CARRY SET (GOOD RETURN)
5400	063610	010001	MOV R0,R1	SAVE CONTENTS OF TSSR
5401	063612		NEXT,ERRNO	
5402	063612		ERRDF 212#	DEVICE FATAL SSR FAILED TO SET
	063612	104455		TRAP C#ERDF
	063614	001615		.WORD 909
	063616	065601		.WORD T208SSR
	063620	012046		.WORD PKTSSR
5403	063622	005237 002222	INC FATFLG	SET FATAL ERROR FLAG
5404	063626		CKLOOP 220#	LOOP ON ERROR, IF FLAG SET
	063626	104406		TRAP C#CLP1
5405				
5406	063630	004737 066176	JSR PC,T20SRD	SETUP PACKET FOR READ STATUS
5407	063634	012704 066600	MOV #T20PK2,R4	GET WRITE SUBSYSTEM COMMAND PACKET
5408	063640	010465 000000	MOV R4,TSD8(R5)	SET THE PACKET ADDRESS TO EXECUTE
5409	063644	004737 016336	JSR PC,CHKTSSR	WAIT FOR SSR TO SET
5410	063650		FORCERROR 232#	FORCE ERROR IF FORCER=1
5411	063664	103407	BCS 240#	BR IF CARRY SET (GOOD RETURN)
5412	063666	010001	MOV R0,R1	SAVE CONTENTS OF TSSR
5413	063670		NEXT,ERRNO	
5414	063670		ERRDF 232#	DEVICE FATAL SSR FAILED TO SET
	063670	104455		TRAP C#ERDF
	063672	001616		.WORD 910
	063674	065362		.WORD T203SSR
	063676	012046		.WORD PKTSSR
5415	063700	005237 002222	INC FATFLG	SET FATAL ERROR FLAG
5416	063704		CKLOOP 240#	LOOP ON ERROR, IF FLAG SET
	063704	104406		TRAP C#CLP1
5417				
5418	063706	004737 066404	JSR PC,T20SETEXP	SET WORDS 0-7 EXPD=RCV (NOT TESTING)
5419	063712	012701 065132	MOV #T20EXSTA,R1	GET EXPECTED READ STATUS
5420	063716	012702 066472	MOV #T20BFSTA,R2	GET RCV READ STATUS
5421	063722	011211	MOV (R2),(R1)	SET EXPD WORD #8 = RCV TEMP
5422	063724	016261 000002 000002	MOV 2(R2),2(R1)	SET EXPD WORD #9 = RCV (NOT TESTED)
5423	063732	052711 100000	BIS #S1.PARERR,(R1)	SET EXP PAR ERR =1
5424	063736	005000	CLR R0	HIGH RCV ADDRESS FOR CKMSG2
5425	063740	012701 066452	MOV #T20BFR,R1	LOW RCV ADDRESS FOR CKMSG2
5426	063744	012702 065112	MOV #T20EXP,R2	EXPD ADDRESS
5427	063750	012703 000024	MOV #20,R3	NUMBER OF BYTES TO COMPARE
5428	063754	004737 011500	JSR PC,CKMSG2	EXPD EQUAL RCV?
5429	063760		FORCERROR 252#,NOTSSR	NO
5430	063770	103404	BCS 260#	BR IF YES
5431	063772		NEXT,ERRNO	
5432	063772		ERRHRD 252#	REPORT ERROR
	063772	104456		TRAP C#ERHRD
	063774	001617		.WORD 911
	063776	065647		.WORD T20SWP
	064000	012350		.WORD MSGSTAT
5433	064002		CKLOOP 260#	LOOP ON ERROR, IF FLAG SET
	064002	104406		TRAP C#CLP1
5434				
5435	064004	012700 000020	MOV #MS.RSFIF,R0	SET RESET FIFO COMMAND
5436	064010	004737 066332	JSR PC,T20WMISC	SETUP T20PK2 FOR WRITE MISC

D1.

SEQ 0210

TEST 9: READ/WRITE DATA PARITY TEST

```

5437 064014 012704 066600      MOV      #T20PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
5438 064020 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
5439 064024 004737 01 336      JSR      PC,CHKTSSR    ;WAIT FOR SSR TO SET
5440 064030                      FORCERROR 282#      ;###FORCE ERROR IF FORCER=1
5441 064044 103407                      BCS      290#      ;BR IF CARRY SET (GOOD RETURN)
5442 064046 010001                      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
5443 064050                      NEXT,ERRNO
5444 064050 282#; ERRDF  ERRNO,T202SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      064050 104455                      TRAP      C#ERDF
      064052 001620                      .WORD      912
      064054 065316                      .WORD      T202SSR
      064056 012046                      .WORD      PKTSSR
5445 064060 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
5446 064064 290#; CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
      064064 104406                      TRAP      C#CLP1
5447                      ; Do a Write Subsystem READ STATUS
5448                      ; If Read Data parity error NOT=0 Then Print Error
5449 064066 004737 066404      JSR      PC,T20SETEXP    ;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
5450 064072 012701 065132      MOV      #T20EXSTA,R1    ;GET EXPECTED READ STATUS
5451 064076 012702 066472      MOV      #T20BFSTA,R2    ;GET RCV READ STATUS
5452 064102 011211                      MOV      (R2),(R1)      ;SET EXPD WORD #8 = RCV TEMP
5453 064104 016261 000002 000002      MOV      2(R2),2(R1)    ;SET EXPD WORD #9 = RCV (NOT TESTED)
5454 064112 042711 100000      BIC      #S1.PARERR,(R1)    ;SET EXP PAR ERR =0
5455 064116 005000      CLR      R0      ;HIGH RCV ADDRESS FOR CKMSG2
5456 064120 012701 066452      MOV      #T20BFR,R1      ;LOW RCV ADDRESS FOR CKMSG2
5457 064124 012702 065112      MOV      #T20EXP,R2      ;EXPD ADDRESS
5458 064130 012703 000024      MOV      #20,,R3      ;NUMBER OF BYTES TO COMPARE
5459 064134 004737 011500      JSR      PC,CKMSG2      ;EXPD EQUAL RCV?
5460 064140                      FORCERROR 302#;NOTSSR    ;###
5461 064150 103404                      BCS      320#      ;BR IF YES
5462 064152                      NEXT,ERRNO
5463 064152 302#; ERHRD  ERRNO,T20RSF,MSGSTAT    ;REPORT ERROR
      064152 104456                      TRAP      C#ERHRD
      064154 001621                      .WORD      913
      064156 065756                      .WORD      T20RSF
      064160 012350                      .WORD      MSGSTAT
5464 064162 320#; CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
      064162 104406                      TRAP      C#CLP1
5465                      ; (* Verify Data can be transferred without a Parity Error *)
5466                      ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
5467 064164 012700 000001      MOV      #WF.I4RES,R0    ;IRESV4==>IRSTR = 1
5468 064170 004737 066317      JSR      PC,T20WFMT    ;SETUP T20PK2 FOR WRITE FORMAT
5469 064174 012704 066600      MOV      #T20PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
5470 064200 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
5471 064204 004737 016336      JSR      PC,CHKTSSR    ;WAIT FOR SSR TO SET
5472 064210                      FORCERROR 332#      ;###FORCE ERROR IF FORCER=1
5473 064224 103407                      BCS      340#      ;BR IF CARRY SET (GOOD RETURN)
5474 064226 010001                      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
5475 064230                      NEXT,ERRNO
5476 064230 332#; ERDF  ERRNO,T203SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
      064230 104455                      TRAP      C#ERDF
      064232 001622                      .WORD      914
      064234 065601                      .WORD      T203SSR
      064236 012046                      .WORD      PKTSSR
5477 064240 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
5478 064244 340#; CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
      064244 104406                      TRAP      C#CLP1

```

TEST 9: READ/WRITE DATA PARITY TEST

```

5479      ; Do a WRITE NPR to set loopback and tape direction OUT and
5480      ; and CLEAR Write Wrong Parity.
5481 054246 012700 000100      MOV    #NPR,OUT,R0      ;SET TAPE DIRECTION OUT
5482 064252 052700 000040      BIS    #NPR,LOOP,R0      ;SET LOOPBACK
5483 064256 052700 000020      BIS    #NPR,WRP,R0      ;CLEAR WRITE WRONG PARITY (INVERTED)
5484 064262 004737 066216      JSR     PC,T20WNPR      ;SETUP T20PK2 FOR WRITE NPR
5485 064266 012704 066600      MOV     #T20PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
5486 064272 010465 000000      MOV     R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
5487 064276 004737 016336      JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
5488 064302      FORCERROR 352$      ;BDFORCE ERROR IF FORCER=1
5489 064316 103407      BCS     360$      ;BR IF CARRY SET (GOOD RETURN)
5490 064320 010001      MOV     R0,R1      ;SAVE CONTENTS OF TSSR
5491 064322      NEXT,ERRNO
5492 064322 352$: ERRDF  ERRNO,T204SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      TRAP    C$ERDF
      .WORD   915
      .WORD   T204SSR
      .WORD   PKTSSR
5493 064332 005237 002222      INC     FATFLG      ;SET FATAL ERROR FLAG
5494 064336 360$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      TRAP    C$CLP1
5495      ; Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
5496 064340 012700 000001      MOV     #1,R0      ;WRITE 1 BYTE
5497 064344 012701 002312      MOV     #DATA,R1      ;FIFO WRITE DATA ADDRESS
5498 064350 004737 066256      JSR     PC,T20WFIF      ;SETUP T20PK2 FOR WRITE FIFO
5499 064354 012704 066600      MOV     #T20PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
5500 064360 010465 000000      MOV     R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
5501 064364 004737 016336      JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
5502 064370      FORCERROR 372$      ;BDFORCE ERROR IF FORCER=1
5503 064404 103407      BCS     380$      ;BR IF CARRY SET (GOOD RETURN)
5504 064406 010001      MOV     R0,R1      ;SAVE CONTENTS OF TSSR
5505 064410      NEXT,ERRNO
5506 064410 372$: ERRDF  ERRNO,T205SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      TRAP    C$ERDF
      .WORD   916
      .WORD   T205SSR
      .WORD   PKTSSR
5507 064420 005237 002222      INC     FATFLG      ;SET FATAL ERROR FLAG
5508 064424 380$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      TRAP    C$CLP1
5509      ; Do a READ FIFO with tape direction OUT to load tape out write latch
5510 064426 012700 000001      MOV     #1,R0      ;SET READ BYTE COUNT
5511 064432 004737 066236      JSR     PC,T20RFIF      ;SETUP T20PK2 FOR READ FIFO
5512 064436 012704 066600      MOV     #T20PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
5513 064442 010465 000000      MOV     R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
5514 064446 004737 016336      JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
5515 064452      FORCERROR 392$      ;BDFORCE ERROR IF FORCER=1
5516 064460 103407      BCS     400$      ;BR IF CARRY SET (GOOD RETURN)
5517 064470 010001      MOV     R0,R1      ;SAVE CONTENTS OF TSSR
5518 064472      NEXT,ERRNO
5519 064472 392$: ERRDF  ERRNO,T206SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      TRAP    C$ERDF
      .WORD   917
      .WORD   T206SSR
      .WORD   PKTSSR
5520 064502 005237 002222      INC     FATFLG      ;SET FATAL ERROR FLAG
5521 064506 400$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET

```

F1

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 28-71

SEQ 0212

TEST 9: READ/WRITE DATA PARITY TEST

```

064506 104406
5522 ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 (sets read strobe low) TRAP C$CLP1
5523 ; (Read Strobe sets PAR IN H [Parity Error])
5524 064510 005000 CLR R0 ; IRESV4==>IRSTR = 0
5525 064512 004737 066312 JSR PC,T20WFMT ; SETUP T20PK2 FOR WRITE FORMAT
5526 064516 012704 066600 MOV @T20PK2,R4 ; GET WRITE SUBSYSTEM COMMAND PACKET
5527 064522 010465 000000 MOV R4,TSD8(R5) ; SET THE PACKET ADDRESS TO EXECUTE
5528 064526 004737 016336 JSR PC,CHKTSSR ; WAIT FOR SSR TO SET
5529 064532 FORCERROR 412$ ; GOODFORCE ERROR IF FORCER=1
5530 064546 103407 BCS 420$ ; BR IF CARRY SET (GOOD RETURN)
5531 064550 010001 MOV R0,R1 ; SAVE CONTENTS OF TSSR
5532 064552 NEXT.ERRNO
5533 064552 412$: ERRDF ERRNO,T208SSR,PKTSSR ; DEVICE FATAL SSR FAILED TO SET
064552 104455 TRAP C$ERDF
064554 001626 .WORD 918
064556 065601 .WORD T208SSR
064560 012046 .WORD PKTSSR
5534 064562 005237 002222 INC FATFLG ; SET FATAL ERROR FLAG
5535 064566 420$: CKLOOP ; LOOP ON ERROR, IF FLAG SET
064566 104406 TRAP C$CLP1
5536 ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
5537 064570 012700 000001 MOV @WF.I4RES,R0 ; IRESV4==>IRSTR = 1
5538 064574 004737 066312 JSR PC,T20WFMT ; SETUP T20PK2 FOR WRITE FORMAT
5539 064600 012704 066600 MOV @T20PK2,R4 ; GET WRITE SUBSYSTEM COMMAND PACKET
5540 064604 010465 000000 MOV R4,TSD8(R5) ; SET THE PACKET ADDRESS TO EXECUTE
5541 064610 004737 016336 JSR PC,CHKTSSR ; WAIT FOR SSR TO SET
5542 064614 FORCERROR 432$ ; GOODFORCE ERROR IF FORCER=1
5543 064630 103407 BCS 440$ ; BR IF CARRY SET (GOOD RETURN)
5544 064632 010001 MOV R0,R1 ; SAVE CONTENTS OF TSSR
5545 064634 NEXT.ERRNO
5546 064634 432$: ERRDF ERRNO,T208SSR,PKTSSR ; DEVICE FATAL SSR FAILED TO SET
064634 104455 TRAP C$ERDF
064636 001627 .WORD 919
064640 065601 .WORD T208SSR
064642 012046 .WORD PKTSSR
5547 064644 005237 002222 INC FATFLG ; SET FATAL ERROR FLAG
5548 064650 440$: CKLOOP ; LOOP ON ERROR, IF FLAG SET
064650 104406 TRAP C$CLP1
5549
5550 ; Do a Write Subsystem READ STATUS
5551 064652 004737 066176 JSR PC,T20SRD ; SETUP PACKET FOR READ STATUS
5552 064656 012704 066600 MOV @T20PK2,R4 ; GET WRITE SUBSYSTEM COMMAND PACKET
5553 064662 010465 000000 MOV R4,TSD8(R5) ; SET THE PACKET ADDRESS TO EXECUTE
5554 064666 004737 016336 JSR PC,CHKTSSR ; WAIT FOR SSR TO SET
5555 064672 FORCERROR 452$ ; GOODFORCE ERROR IF FORCER=1
5556 064706 103407 BCS 460$ ; BR IF CARRY SET (GOOD RETURN)
5557 064710 010001 MOV R0,R1 ; SAVE CONTENTS OF TSSR
5558 064712 NEXT.ERRNO
5559 064712 452$: ERRDF ERRNO,T203SSR,PKTSSR ; DEVICE FATAL SSR FAILED TO SET
064712 104455 TRAP C$ERDF
064714 001630 .WORD 920
064716 065362 .WORD T203SSR
064720 012046 .WORD PKTSSR
5560 064722 005237 002222 INC FATFLG ; SET FATAL ERROR FLAG
5561 064726 460$: CKLOOP ; LOOP ON ERROR, IF FLAG SET
064726 104406 TRAP C$CLP1
5562 ; If Read Data parity error NOT=0 Then Print Error

```

G1

SEQ 0213

TEST 9: READ/WRITE DATA PARITY TEST

5563	064730	004737	066404	JSR	PC,T20SETEXP	;	SET WORDS 0-7 EXPD=RECV (NOT TESTING)
5564	064734	012701	065132	MOV	*T20EXSTA,R1	;	GET EXPECTED READ STATUS
5565	064740	012702	066472	MOV	*T20BFSTA,R2	;	GET RECV READ STATUS
5566	064744	011211		MOV	(R2),(R1)	;	SET EXPD WORD *8 = RECV TEMP
5567	064746	016261	000002	MOV	2(R2),2(R1)	;	SET EXPD WORD *9 = RECV (NOT TESTED)
5568	064754	042711	100000	BIC	*S1,PARERR,(R1)	;	SET EXP PAR ERR = 0
5569	064760	005000		CLR	R0	;	HIGH RECV ADDRESS FOR CKMSG2
5570	064762	012701	066452	MOV	*T20BFR,R1	;	LOW RECV ADDRESS FOR CKMSG2
5571	064766	012702	065112	MOV	*T20EXP,R2	;	EXPD ADDRESS
5572	064772	012703	000024	MOV	*20,R3	;	NUMBER OF BYTES TO COMPARE
5573	064776	004737	011500	JSR	PC,CKMSG2	;	EXPD EQUAL RECV?
5574	065002			FORCERR	472*,NOTSSR	;	END
5575	065012	103404		BCS	480*	;	BR IF YES
5576	065014			NEXT,ERRNO		;	
5577	065014			ERRHRD	ERRNO,T20CWP,MSGSTAT	;	REPORT ERROR
	065014	104456					TRAP C1ERHRD
	065016	001631					.WORD 921
	065020	066057					.WORD T20CWP
	065022	012350					.WORD MSGSTAT
5578	065024			480*:	CKLOOP	;	LOOP ON ERROR, IF FLAG SET
	065024	104406					TRAP C1CLP1
5579							
5580	065026			FORCEXIT	555*	;	END
5581	065036	013703	002316	MOV	TSTPTR,R5	;	RESTORE CURRENT TSTBLK POINTER
5582	065042	020327	003062	CMR	R3,*TBLEND	;	END OF TSTBLK?
5583	065046	103002		BHIS	555*	;	BR IF YES
5584	065050	000137	063124	JMP	100*	;	DO ANOTHER TSTBLK PATTERN
5585	065054			555*:			
5586							
5587	065054			ENDSUB		;	////////// END SUBTEST //////////
	065054						L10074:
	065054	104403					TRAP C1ESUB
5588							
5589	065056	005737	002222	TST	FATFLG	;	ANY FATAL ERRORS ?
5590	065062	001402		BEQ	560*	;	BRANCH IF NOT
5591	065064	004737	017202	JSR	PC,CKDROP	;	TRY TO DROP THE UNIT
5592	065070			560*:			
5593	065070	004737	016456	JSR	PC,TSTLOOP	;	DO ITERATIONS?
5594	065074	103002		BCC	565*	;	BR IF NO
5595	065076	000137	050556	JMP	T18LOOP	;	LOOP UNTIL ITERATIONS DONE
5596	065102			565*:			
5597	065102			EXIT	TST	;	////////// EXIT TEST //////////
	065102	104432					TRAP C1EXIT
	065104	001610					.WORD L10073-
5598							
5599							
5600				;			
5601				;	LOCAL STORAGE FOR THIS TEST		
5602				;			
5603							
5604							
5605	065106			T20MSK:		;	MASK OF UNTESTED BITS IN READ STATUS
5606						;	UNTESTED BITS ARE SET TO 1
5607	065106	377		.BYTE	*C<000>	;	BYTE 0 MASK
5608	065107	037		.BYTE	*C<340>	;	BYTE 1 MASK (PARERR,IRESV2,IRESV1)
5609	065110	360		.BYTE	*C<017>	;	BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
5610	065111	000		.BYTE	0	;	MAKE IT EVEN

TEST 9: READ/WRITE DATA PARITY TEST

```

5611
5612 065112          T20EXP:          ;BEGIN EXPECTED DATA BUFFER
5613 065112 000000    .WORD 0          ;MESSAGE TYPE
5614 065114 000000    .WORD 0          ;DATA FIELD LENGTH
5615 065116 000000    .WORD 0          ;RBPGR
5616 065120 000000    .WORD 0          ;XST0
5617 065122 000000    .WORD 0          ;XST1
5618 065124 000000    .WORD 0          ;XST2
5619 065126 000000    .WORD 0          ;XST3
5620 065130 000000    .WORD 0          ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
5621 065132          T20EXSTA: .BLKB 64. ;EXPECTED READ STATUS AND WRITE FIFO DATA
5622 065232          T20XEND:          ;END EXPECTED DATA BUFFER
5623
5624          ;*
5625          ;LOCAL TEXT MESSAGES FOR TEST
5626          ;-
5627 065232      122      145      141  TST20ID:      .ASCIIZ 'Read/Write Data Parity'
5628 065261      127      122      111  T20SSR: .ASCIIZ 'WRITE CHARACTERISTICS Failed'
5629 065316      127      122      111  T202SSR: .ASCIIZ 'WRITE SUBSYSTEM (Write Misc) Failed'
5630 065362      127      122      111  T203SSR: .ASCIIZ 'WRITE SUBSYSTEM (Read Status) Failed'
5631 065427      127      122      111  T204SSR: .ASCIIZ 'WRITE SUBSYSTEM (Write Npr) Failed'
5632 065472      127      122      111  T205SSR: .ASCIIZ 'WRITE SUBSYSTEM (Write FIFO) Failed'
5633 065536      127      122      111  T206SSR: .ASCIIZ 'WRITE SUBSYSTEM (Read FIFO) Failed'
5634 065601      127      122      111  T208SSR: .ASCIIZ 'WRITE SUBSYSTEM (Write Format) Failed'
5635 065647      122      145      141  T20SWP: .ASCIIZ 'Read Data Parity Error (PARERR) Failed to Set after Write Wrong Parity'
5636 065755      122      145      141  T20RSF: .ASCIIZ 'Read Data Parity Error (PARERR) Failed to Clear after RESET FIFO'
5637 066057      122      145      141  T20CWP: .ASCIIZ 'Read Data Parity Error (PARERR) occurred in Data Loopback'
5638
5639
5640
5641          ;*
5642          ; CLEAR MESSAGE BUFFER
5643          ;-
5643 066152          T20CLRBUF:
5644 066152          SAVREG
5645 066156 012701 066452    MOV      @T20BFR,R1
5646 066162 012702 000120    MOV      @T20BEND-T20BFR,R2
5647 066166 105021          104:    CLRB      (R1)+
5648 066170 005302          DEC      R2
5649 066172 003375          BGT      104
5650 066174 000207          RTS      PC
5651
5652
5653          ;*
5654          ; SETUP T20PK2 PACKET FOR READ STATUS
5655          ;-
5655 066176          T20SRD:
5656 066176 004737 066152    JSR      PC,T20CLRBUF
5657 066202 012700 066610    MOV      @T20OT2,R0
5658 066206 112720 000005    MOVB     @PW,RDSTATUS,(R0)+
5659 066212 105010          CLRB      (R0)
5660 066214 000207          RTS      PC
5661
5662
5663          ;*
5664          ; SETUP T20PK2 PACKET FOR WRITE NPR
5665          ;
5666          ; INPUT:
5667          ;      R0 CONTAINS BSEL1 NPR DATA

```

TEST 9: READ/WRITE DATA PARITY TEST

```

5668
5669
5670 066216
5671 066216 004737 066152
5672 066222 012701 066610
5673 066226 112721 000011
5674 066232 110011
5675 066234 000207
5676
5677
5678
5679
5680
5681
5682
5683 066236
5684 066236 004737 066152
5685 066242 012701 066610
5686 066246 112721 000003
5687 066252 110021
5688 066254 000207
5689
5690
5691
5692
5693
5694
5695
5696 066256
5697 066256
5698 066262 004737 066152
5699 066266 012702 066610
5700 066272 112722 000004
5701 066276 110022
5702 066300 005022
5703 066302 112122
5704 066304 005300
5705 066306 003375
5706 066310 000207
5707
5708
5709
5710
5711
5712
5713
5714 066312
5715 066312 004737 066152
5716 066316 012701 066610
5717 066322 112721 000007
5718 066326 110021
5719 066330 000207
5720
5721
5722
5723
5724

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

```

5725 066332
5726 066332 012701 066610
5727 066336 112721 000010
5728 066342 110011
5729 066344 000207
5730
5731
5732
5733 066346
5734 066346 012700 066610
5735 066352 112720 000010
5736 066356 112710 000200
5737 066362 000207
5738
5739
5740
5741 066364
5742 066364 012701 065112
5743 066370 012700 000120
5744 066374 105021
5745 066376 005300
5746 066400 003375
5747 066402 000207
5748
5749
5750
5751
5752 066404
5753 066404 012702 065112
5754 066410 012703 066452
5755 066414 012700 000010
5756 066420 012322
5757 066422 005300
5758 066424 003375
5759 066426 000207
5760
5761
5762
5766
5767
5768
5769 066430
5770 066430 100004
5771 066432 066440
5772 066434 000000
5773 066436 000012
5774
5775 066440
5776 066440 066452
5777 066442 000000
5778 066444 000024
5779 066446 000000
5780 066450 000007
5781
5782
5783
5784

T20WMISC:
    MOV     @T20DT2,R1          ;WRITE SUBSYSTEM DATA BUFFER
    MOVB    @PW.WMISC,(R1)+     ;STORE WRITE MISCELLANEOUS IN BSEL0
    MOVB    R0,(R1)            ;STORE INVERT EXTENDED FEATURES IN BSEL1
    RTS     PC                  ;RETURN

; *
; SETUP T20PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
; -
T20SEXT:
    MOV     @T20DT2,R0          ;WRITE SUBSYSTEM DATA BUFFER
    MOVB    @PW.WMISC,(R0)+     ;STORE WRITE MISCELLANEOUS IN BSEL0
    MOVB    @MS.EXT,(R0)       ;STORE INVERT EXTENDED FEATURES IN BSEL1
    RTS     PC                  ;RETURN

; *
; CLEAR EXPECTED DATA MESSAGE BUFFER
; -
T20CLEXP:
    MOV     @T20EXP,R1          ;GET EXPD ADDRESS
    MOV     @T20EXEND-T20EXP,R0 ;GET EXPD SIZE
10$:    CLRB    (R1)+           ;CLEAR A BYTE
    DEC     R0                  ;DONE?
    BGT     10$                ;BK IF NO
    RTS     PC                  ;RETURN

; *
; Set WORDS 0-7 of expd message BUFFER = to recv since not testing
; -
T20SETEXP:
    MOV     @T20EXP,R2          ;GET EXPD
    MOV     @T20BFR,R3         ;GET READ STATUS RECV BUFFER
    MOV     @B.,R0              ;SET WORDS 0-7 EXP=RECV
5$:    MOV     (R3)+,(R2)+      ;SET EXPD=RECV
    DEC     R0                  ;DONE WORDS 0-7 WORDS?
    BGT     5$                 ;BR IF NO
    RTS     PC                  ;RETURN

;
; WRITE CHARACTERISTICS COMMAND PACKET
;
T20PACKET:
    .WORD    100004             ;COMMAND PACKET FOR TEST
    .WORD    T20DATA            ;WRITE CHARACTERISTICS COMMAND, WITH ACK
    .WORD    0                  ;ADDRESS OF CHARACTERISTICS BLOCK
    .WORD    10.               ;MINIMUM MESSAGE PACKET SIZE

T20DATA:
    .WORD    T20BFR             ;CHARACTERISTICS DATA BLOCK
    .WORD    0                  ;ADDRESS OF MESSAGE BUFFER
    .WORD    20.                ;LENGTH OF MESSAGE BUFFER
    .WORD    0                  ;ESS,ENB,EAI,ERI
    .WORD    7                  ;EXTENDED FEATURES UNIT NO.

;MESSAGE BUFFER FOR ALL TEST 17 COMMANDS

```

TEST 9: READ/WRITE DATA PARITY TEST

```

5785 066452          T20BFR:          ;BEGIN MESSAGE BUFFER
5786 066452 000000          ;WORD 0          ;MESSAGE TYPE
5787 066454 000000          ;WORD 0          ;DATA FIELD LENGTH
5788 066456 000000          ;WORD 0          ;RBPCR
5789 066460 000000          ;WORD 0          ;XST0
5790 066462 000000          ;WORD 0          ;XST1
5791 066464 000000          ;WORD 0          ;XST2
5792 066466 000000          ;WORD 0          ;XST3
5793 066470 000000          ;WORD 0          ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
5794 066472          T20BFSTA: .BLKB 64. ;READ STATUS AND WRITE FIFO BUFFER
5795 066572          T20BEND:          ;END OF MESSAGE BUFFER
5796
5797          ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
5798
5800          ;.=<.+10>6177770
5802 066600          T20PK2:          ;WRITE SUBSYSTEM WITH ACK
5803 066600 100006          ;WORD P.WRTSUB:P.ACK ;LOW ADDRESS OF DATA BLOCK
5804 066602 066610          ;WORD T20DT2      ;HIGH ADDRESS OF DATA BLOCK
5805 066604 000000          ;WORD 0          ;MINIMUM MESSAGE PACKET SIZE
5806 066606 000012          ;WORD 10.
5807
5808 066610          T20DT2:          ;DATA BLOCK
5809 066610          ;BYTE 0          ;BSEL0
5810 066611          ;BYTE 0          ;BSEL1
5811 066612 000000          ;WORD 0          ;SEL2
5812 066614          ;BLKB 64.        ;WRITE FIFO DATA OUTPUT BUFFER
5813
5814
5815 066714          ENDTST
5816 066714
5817 066714 104401          L10073:      TRAP      C#ETST
5818
5819          .SBITL TEST 10: MANUAL INTERVENTION
5820
5821          ;THE MANUAL INTERVENTION TEST IS A STANDALONE ROUTINE (NOT REALLY A "TEST")
5822          ;THAT ALLOWS THE OPERATOR TO CHECK OUT VARIOUS ELEMENTS AND FUNCTIONS OF
5823          ;THE SUBSYSTEM THAT CANNOT BE MANIPULATED BY THE PROGRAM ALONE. WHEN
5824          ;THIS ROUTINE IS STARTED, IT FIRST PRINTS OUT A MENU OF SELECTABLE
5825          ;SUBTESTS AND THEN WAITS FOR THE OPERATOR TO TYPE IN A SELECTION CODE.
5826          ;THE ONLY WAYS TO EXIT THIS ROUTINE AND RETURN TO THE DIAGNOSTIC SUPERVISOR
5827          ;ARE BY TYPING (CTRL-C) OR SELECTING CODE 7.
5828          ;SELECTION CODES AND SUBROUTINES ARE:
5829
5830          ;
5831          ;      CODE      ROUTINE
5832          ;
5833          ;      0      HELP. PRINTS THIS MENU.
5834          ;      1      TURN ON ALL M7196 LED INDICATORS
5835          ;      2      TURN OFF ALL M7196 LED INDICATORS
5836          ;      3      OFFLINE/ONLINE ATTENTION TEST
5837          ;      4      WRITE-PROTECT TEST
5838          ;      5      INITIATE TRANSPORT SERVO EXERCISER
5839          ;      6      PRINT EXTENDED TRANSPORT STATUS
5840          ;      7      EXIT (RETURN TO SUPERVISOR)
5841
5842          ;EACH MENU ITEM CORRESPONDS TO A SUBTEST, AS FOLLOWS:

```

TEST 10: MANUAL INTERVENTION

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

TEST 10: MANUAL INTERVENTION

```

5899      ;
5900      ;
5901      ;
5902      ;
5903 066716      BGNTSi
      066716
5908 066716      RFLAGS      RO
      066716      104421
5909 066720      001403
5910 066722      012700      072300
5911 066726      000402
5912 066730      21$:
5913 066730      012700      073415
5914 066734      004737      016510
5915 066740      004737      020500
5916 066744      103402
5917 066746      000137      071500
5918 066752      22$:
5922 066752      005037      002222
5923 066756      012737      176750      071512
5924 066764      004737      015774
5925 066770      103427
5926 066772      010001
5927 066774      032701      000200
5928 067000      001023
5929 067002      012727      000250
      067006      000000
      067010      013727      002116
      067014      000000
      067016      005367      177772
      067022      001375
      067024      005367      177756
      067030      001367
5930 067032      005337      071512
5931 067036      001352
5932 067040      DEC      T38DLY
      067040      104455
      067042      001751
      067044      003652
      067046      012034
5933 067050      012700      073442
5934 067054      012701      000006
5935 067060      004737      020256
5936 067064      010004
5937 067066      006304
5938 067070      000174      067074
5939 067074      066752
5940 067076      067112
5941 067100      067374
5942 067102      067626
5943 067104      070262
5944 067106      071216
5945 067110      071474
5946 067112      10$:
      067112      012746      073311
      067116      012746      000001
      067116      012746      000001

      BGNTSi
      RFLAGS      RO
      BEQ      21$
      MOV      0T38NE,RO
      BR      3$
      MOV      0T38ID,RO
      JSR      PC,TSTSETUP
      JSR      PC,CHKMAN
      BCS      22$
      JMP      64$
      CLR      FATFLG
      MOV      065000.,T38DLY
      JSR      PC,SOFINIT
      BCS      23$
      MOV      RO,R1
      BIT      0SSR,R1
      BNE      23$
      DELAY      250
      MOV      0250,(PC)+
      .WORD      0
      MOV      L$DLY,(PC)+
      .WORD      0
      DEC      -6(PC)
      BNE      .-4
      DEC      -22(PC)
      BNE      .-20
      DEC      T38DLY
      BNE      5$
      ERROF      ERRNO,SFIERR,SFIMSG
      TRAP      C$ERDF
      .WORD      1001
      .WORD      SFIERR
      .WORD      SFIMSG
      MOV      0MIMENU,RO
      MOV      06,R1
      JSR      PC,GETSEL
      MOV      RO,R4
      ASL      R4
      JMP      06$(R4)
      .WORD      2$
      .WORD      10$
      .WORD      15$
      .WORD      20$
      .WORD      25$
      .WORD      35$
      .WORD      63$
      PRINTF      0T38MS2
      MOV      0T38MS2,-(SP)
      MOV      01,-(SP)
      T10::
      GET OPERATOR FLAGS
      TRAP      C$RFLA
      BR, IF OK TO RUN
      "TEST NOT EXECUTED"
      JUMP IF NOT FIRST TEST
      TEST ID MESSAGE
      DO THE COMMON SETUP
      IS MANUAL INTERVENTION ALLOWED?
      BR, IF MANUAL INTER ALLOWED
      JUMP IF NOT ALLOWED
      CLEAR THE FATAL ERROR FLAG
      SET UP DELAY COUNTER
      DO A SOFT INIT
      BRANCH IF OK
      CONTENTS OF TSSR REGISTER
      CHECK FOR TSSR SET
      KEEP GOING IF NOT SET
      CALL DELAY ROUTINE
      BUMP COUNTER DOWN
      BR, IF MORE TIME LEFT
      REPORT FATAL ERROR
      MENU OF MANUAL INTERVENTIONS
      MAXIMUM ALLOWED SELECTION
      GO GET THE OPERATORS SELECTION
      GET NUMBER FROM ROUTINE
      CONVERT TO WORD OF FSET
      JUMP TO PROPER LOOP
      RETYPE THE MENU
      1 TURN ON LED'S
      2 TURN OFF LED'S
      3 ONLINE ATTENTION
      4 WRITE PROTECT
      5 EXTENDED TRANSPORT STATUS
      6 LEAVE THE TEST
      TELL OPERATOR TO CNTRL-C FOR EXIT

```

TEST 10: MANUAL INTERVENTION

067122	010600								MOV	SP,R0
067124	104417								TRAP	C\$PNTF
067126	062706	000004							ADD	#4,SP
5947	067132	004737	074046			JSR	PC,T38REST			;SET P KET TO INITIAL VALUES
5948	067136	004737	015774			JSR	PC,SOFINIT			;DO SC. INIT OF CONTROLLER
5949	067142	103405				BCS	100\$			;BR IF SOFT INIT = OK
5953	067144	010001				MOV	R0,R1			;SAVE CONTENTS OF TSSR
5954	067146					ERRDF	ERRNO,SFIERR,SFIMSG			;DEVICE FATAL ERROR DURING INIT
	067146	104455							TRAP	C\$ERDF
	067150	001752							.WORD	1002
	067152	003652							.WORD	SFIERR
	067154	012034							.WORD	SFIMSG
5955	067156	013737	002202	072240	100\$:	MOV	UNITN,T38DSW			;SET UNIT NUMBER
5956										
5957	067164	012704	072220			MOV	#T38PK2,R4			;SUBROUTINE NEEDS PACKET ADDRESS
5958	067170	004737	010662			JSR	PC,WRTCHR			;ISSUE WRITE CHARACTERISTICS
5959	067174	103405				BCS	110\$			;BR. IF COMMAND ISSUED OK
5963	067176	010001				MOV	R0,R1			;SAVE CONTENTS OF TSSR
5964	067200					ERRHRD	ERRNO,WRTMSG,SFIMSG			;WRITE CHARACTERISTISC FAILED
	067200	104456							TRAP	C\$ERHRD
	067202	001753							.WORD	1003
	067204	005056							.WORD	WRTMSG
	067206	012034							.WORD	SFIMSG
5965	067210									
5966	067210	112737	000000	071531	110\$:	MOVB	#0,T38BS1			;CLEAR BIT #4
5967	067216	112737	000011	071530		MOVB	#11,T38BS0			;WRITE MISC COMMAND
5968	067224	012704	071520			MOV	#T38PACKET,R4			;SET UP NEW WRT. SUBSYS MEM. COMMAND
5969										
5970										
5971										
5972	067230	010465	000000			MOV	R4,T5DB(R5)			;SET THE PACKET ADDRESS
5973	067234	004737	016336			JSR	PC,CHKTSSR			;WAIT FOR SSR TO SET
5974	067240	103405				BCS	150\$			;BR IF CARRY SET (GOOD RETURN)
5975	067242	010001				MOV	R0,R1			;SAVE CONTENTS OF TSSR
5979	067244					ERRDF	ERRNO,T38SSR,PKTSSR			;DEVICE FATAL SSR FAILED TO SET
	067244	104455							TRAP	C\$ERDF
	067246	001754							.WORD	1004
	067250	072716							.WORD	T38SSR
	067252	012046							.WORD	PKTSSR
5980	067254				150\$:	CKLOOP				;LOOP ON ERROR, IF FLAG SET
	067254	104406							TRAP	C\$CLP1
5981	067256					SETPRI	#PRI06			;RAISE THE PRIORITY
	067256	012700	000300						MOV	#PRI06,R0
	067262	104441							TRAP	C\$SPRI
5982	067264	005037	071504			CLR	TTION2			;ASSUME INTERRUPTS ARE ENABLED
5983	067270	032737	000100	177560		BIT	#100,#TTICSR			;ARE TTI INTERRUPTS ON ?
5984	067276	001005				BNE	701\$			;BRANCH IF YES
5985	067300	005237	071504			INC	TTION2			;FLAG SET IF INTERRUPTS OFF
5986	067304	052737	000100	177560		BIS	#100,#TTICSR			;ENABLE INTERRUPTS
5987	067312	012701	000060		701\$:	MOV	#TTIVEC,R1			;START OF TTI VECTORS
5988	067316	011137	071506			MOV	(R1),TVSAV2			;SAVE THE CUPRENT TTI VECTOR
5989	067322	012721	071000			MOV	#590\$,(R1)			;SET NEW INTERRUPT ROUTINE
5990	067326	011137	071510			MOV	(R1),TPSAV2			;SAVE THE VECTOR PRIORITY
5991	067332	012711	000300			MOV	#PRI06,(R1)			;USE PRIORITY SIX
5992	067336					SETPRI	#PA100			;LOWER INTERRUPT BR LEVEL
	067336	012700	000000						MOV	#PRI00,R0
	067342	104441							TRAP	C\$SPRI

B2

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 28-80

SEQ 0221

TEST 10: MANUAL INTERVENTION

```

5993 067344 012701 177777      MOV      #-1,R1      ;DATA TO WRITE TO TSDB
5994 067350 000240      12$: NOP      ;ALLOW OPERATOR TO TYPE ^C
5995 067352 012702 001750      MOV      #1000.,R2     ;SET-UP INNER LOOP
5996 067356 110165 000000      14$: MOVB     R1,TSDB(R5) ;WRITE DATA TO TSDB
5997 067362 016500 000002      MOV      TSSR(R5),R0    ;READ TSSR
5998 067366 005302      DEC      R2      ;REDUCE INNER COUNT
5999 067370 001372      BNE      14$     ;LOOP TILL EXPIRES
6000 067372 000766      BR       12$     ;LOOP UNTIL HALTED
6001
6002 067374      15$: PRINTF  #T38MS2      ;TYPE CNTL C TO EXIT
      067374 012746 073311      MOV      #T38MS2,-(SP)
      067400 012746 000001      MOV      #1,-(SP)
      067404 010600      MOV      SP,R0
      067406 104417      TRAP     C:PNTF
      067410 062706 000004      ADD      #4,SP
6003 067414 004737 015774      JSR      PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
6004 067420 103405      BCS      200$     ;BR IF SOFT INIT - OK
6008 067422 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
6009 067424      ERRDF     ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
      067424 104455      TRAP     C:ERDF
      067426 001755      .WORD    1005
      067430 003652      .WORD    SFIERR
      067432 012034      .WORD    SFIMSG
6010 067434
6011 067434 013737 002202 072240 200$: MOV      UNITN,T38DSW      ;SET UNIT NUMBER
6012 067442 012704 072220      MOV      #T38PK2,R4     ;SUBROUTINE NEEDS PACKET ADDRESS
6013 067446 004737 010662      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
6014 067452 103405      BCS      210$     ;BR, IF COMMAND ISSUED OK
6018 067454 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
6019 067456      ERRHRD    ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      067456 104456      TRAP     C:ERHRD
      067460 001756      .WORD    1006
      067462 005056      .WORD    WRTMSG
      067464 012034      .WORD    SFIMSG
6020
6021
6022
6023
6024
6025 067466
6026 067466 112737 000000 071531 210$: MOVB     #0,T38BS1      ;CLEAR BIT #4
6027 067474 112737 000025 071530      MOVB     #25,T38BS0     ;STOP DRIVE TEST 22
6028 067502 012704 071520      MOV      #T38PACKET,R4    ;SET UP NEW WRT. SUBSYS MEM. COMMAND
6029 067506 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS
6030 067512 004737 016336      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
6031 067516 103405      BCS      250$     ;BR IF CARRY SET (GOOD RETURN)
6032 067520 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
6036 067522      ERRDF     ERRNO,T38SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      067522 104455      TRAP     C:ERDF
      067524 001757      .WORD    1007
      067526 072716      .WORD    T38SSR
      067530 012046      .WORD    PKTSSR
6037 067532      250$: CKLOOP
      067532 104406      SETPRI  #PRI06      ;LOOP ON ERROR, IF FLAG SET
6038 067534      SETPRI  #PRI06      ;RAISE THE PRIORITY
      067534 012700 000300      MOV      #PRI06,R0
      067540 104441      TRAP     C:SPRI

```

TEST 10: MANUAL INTERVENTION

Address	Op Code	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	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	Op464	Op4
---------	---------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-----

TEST 10: MANUAL INTERVENTION

6116	070232			380\$:	PRINTF	0T38OFL		"DRIVE IS NOW OFF-LINE"	
	070232	012746	073072					MOV	0T38OFL, -(SP)
	070236	012746	000001					MOV	01, -(SP)
	070242	010600						MOV	SP, R0
	070244	104417						TRAP	C\$PNTF
	070246	062706	000004					ADD	04, SP
6117	070252	005037	002224	390\$:	CLR	INTRECV			
6118	070256	000137	067700		JMP	300\$			
6119	070262			25\$:	GMANTL	T38MSG, T38DAY, -1, NO			
	070262	104443						TRAP	C\$GMAN
	070264	000404						BR	10000\$
	070266	074044						.WORD	T38DAT
	070270	000120						.WORD	T\$CODE
	070272	073355						.WORD	T38MSG
	070274	177777						.WORD	-1
	070276								
6120	070276				BNCOMPLETE	25\$		10000\$:	
	070276	103371							
6121	070300	005737	074044		TST	T38DAT			
6122	070304	001002			BNE	27\$		BCC	25\$
6123	070306	000137	066752		JMP	2\$			
6124	070312			27\$:					
6125	070312	004737	015774		JSR	PC, SOFINIT			
6126	070316	103405			BCS	400\$			
6130	070320	010001			MOV	R0, R1			
6131	070322				ERRDF	ERRNO, SFIERR, SFIMSG			
	070322	104455						TRAP	C\$ERDF
	070324	001763						.WORD	1011
	070326	003652						.WORD	SFIERR
	070330	012034						.WORD	SFIMSG
6132	070332			400\$:	CKLOOP				
	070332	104406						TRAP	C\$CLP1
6133	070334	013737	002202		MOV	UNITN, T38DSW			
6134	070342	012704	072220		MOV	0T38PK2, R4			
6135	070346	004737	010662		JSR	PC, WRTCHR			
6136	070352	103405			BCS	410\$			
6140	070354	010001			MOV	R0, R1			
6141	070356				ERRHRD	ERRNO, WRTMSG, SFIMSG			
	070356	104456						TRAP	C\$ERHRD
	070360	001764						.WORD	1012
	070362	005056						.WORD	WRTMSG
	070364	012034						.WORD	SFIMSG
6142	070366			410\$:	CKLOOP				
	070366	104406						TRAP	C\$CLP1
6143	070370	013701	071544		MOV	T38BFR+6, R1			
6144	070374	010102			MOV	R1, R2			
6145	070376	052702	000004		BIS	0BIT2, R2			
6146	070402	020102			CMP	R1, R2			
6147	070404	001406			BEQ	430\$			
6151	070406				ERRHRD	ERRNO, T38WRL, EXPREC			
	070406	104456						TRAP	C\$ERHRD
	070410	001765						.WORD	1013
	070412	072534						.WORD	T38WRL
	070414	015474						.WORD	EXPREC
6152	070416	005237	002222		INC	FATFLG			
6153	070422			430\$:	CKLOOP				
	070422	104406						TRAP	C\$CLP1

TEST 10: MANUAL INTERVENTION

```

6154 070424 005737 002222          TST    FATFLG          ;WAS THE DRIVE NOT WRITE LOCKED
6155 070430 001402                BEQ    435$             ;BR, IF FLAG NOT SET
6156 070432 000137 066752          JMP    2$             ;RE-WRITE MENU
6157 070436 017737 112462 072272 435$: MOV    @FREE,T38WR      ;SET UP WRITE BUFFER ADDRESS
6158 070444 012704 072270          MOV    @T38PK4,R4        ;GET PACKET ADDRESS
6159 070450 010465 000000          MOV    R4,T38B(R5)        ;SET THE PACKET ADDRESS
6160 070454 004737 016250          JSR    PC,WAITF        ;WAIT FOR SSR TO SET
6161 070460 016501 000002          MOV    TSSR(R5),R1        ;GET TSSR
6162 070464 012702 100206          MOV    @SC!SSR!BIT1!BIT2,R2    ;SET UP EXPECTED
6163 070470 020102                CMP    R1,R2            ;ARE THEY EQUAL (CORRECT)
6164 070472 001404                BEQ    440$             ;BR, IF CORRECT STATUS
6168 070474                ERRHRD   ERRNO,T38WRT,PKTSSR      ;"TSSR INCORRECT AFTER WRITE COMMAND
                                TRAP    C$ERHRD
                                .WORD   1014
                                .WORD   T38WRT
                                .WORD   PKTSSR
                                070474 104456
                                070476 001766
                                070500 072450
                                070502 012046
6169 070504                440$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP    C$CLP1
                                070504 104406
6170 070506 013701 071544          MOV    T38BFR+6,R1        ;READ XSTO CONTENTS
6171 070512 010102                MOV    R1,R2            ;SET UPR EXPECTED
6172 070514 052702 004000          BIS    @BIT11,R2        ;SET THE WRITE LOCK ERROR BIT (XSTO)
6173 070520 020102                CMP    R1,R2            ;WAS THE BIT SET
6174 070522 001404                BEQ    450$             ;BR, IF IT WAS (GOOD)
6178 070524                ERRHRD   ERRNO,T38WLE,EXPREC      ;"WRITE LOCK ERROR BIT NOT SET"
                                TRAP    C$ERHRD
                                .WORD   1015
                                .WORD   T38WLE
                                .WORD   EXPREC
                                070524 104456
                                070526 001767
                                070530 072575
                                070532 015474
6179 070534                450$: CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
                                070534 104406
6180 070536 000137 066752          JMP    2$             ;GO BACK TO MENU
6181
6182          ;*****
6183          ;      SERVO EXERCISER NO LONGER USED
6184          ;*****
6185 070542          30$: PRINTB @T38MS3          ;"EXE ANY OTHER MENU SELECTION TO STOP
6186 070542                PRINTB @T38MS3                MOV    @T38MS3,-(SP)
                                MOV    @1,-(SP)
                                MOV    SP,R0
                                TRAP    C$PNTB
                                ADD     @4,SP
                                070542 012746 072355
                                070546 012746 000001
                                070552 010600
                                070554 104414
                                070556 062706 000004
                                JSR     PC,T38REST          ;SET PACKET TO INITIAL VALUES
                                JSR     PC,SOFINIT          ;DO SOFT INIT OF CONTROLLER
                                BCS     500$              ;BR IF SOFT INIT = OK
                                MOV     R0,R1              ;SAVE CONTENTS OF TSSR
                                ERRDF   ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
                                TRAP    C$ERDF
                                .WORD   1016
                                .WORD   SFIERR
                                .WORD   SFIMSG
                                070576 104455
                                070600 001770
                                070602 003652
                                070604 012034
6195 070606 013737 002202 072240 500$: MOV    UNITN,T38DSW    ;SET UNIT NUMBER
6196 070614 012704 072220          MOV    @T38PK2,R4        ;SUBROUTINE NEEDS PACKET ADDRESS
6197 070620 004737 010662          JSR    PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
6198 070624 103405                BCS     510$             ;BR, IF COMMAND ISSUED OK
6202 070626 010001                MOV     R0,R1              ;SAVE CONTENTS OF TSSR
6203 070630                ERRHRD   ERRNO,WRTMSG,SFIMSG      ;WRITE CHARACTERISTICS FAILED

```

TEST 10: MANUAL INTERVENTION

070630	104456					TRAP	C\$ERHRD
070632	001771					.WORD	1017
070634	005056					.WORD	WRTMSG
070636	012034					.WORD	SFIMSG
6204 070640				510\$:	MOVB	#0,T38BS1	;CLEAR BIT #4
6205 070640	112737	000000	071531		MOVB	#20,T38BS0	;EXECUTE DRIVE TEST 22
6206 070646	112737	000020	071530		MOV	#T38PACKET,R4	;SET UP NEW WRT. SUBSYS MEM. COMMAND
6207 070654	012704	071520			MOV	R4,TSD8(R5)	;SET THE PACKET ADDRESS
6208 070660	010465	000000			JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
6209 070664	004737	016336			BCS	550\$	;BR IF CARRY SET (GOOD RETURN)
6210 070670	103405				MOV	R0,R1	;SAVE CONTENTS OF TSSR
6211 070672	010001				ERRDF	ERRNO,T38SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
070674	104455						TRAP
070676	001772						.WORD
070700	072716						.WORD
070702	012046						.WORD
6216 070704				550\$:	CKLOOP		;LOOP ON ERROR, IF FLAG SET
070704	104406						TRAP
6217 070706					SETPRI	#PRI06	;RAISE THE PRIORITY
070706	012700	000300					MOV
070712	104441						TRAP
6218 070714	005037	071504			CLR	TTION2	;ASSUME INTERRUPTS ARE ENABLED
6219 070720	032737	000100	177560		BIT	#100,#TTICSR	;ARE TTI INTERRUPTS ON ?
6220 070726	001005				BNE	555\$	;BRANCH IF YES
6221 070730	005237	071504			INC	TTION2	;FLAG SET IF INTERRUPTS OFF
6222 070734	052737	000100	177560		BIS	#100,#TTICSR	;ENABLE INTERRUPTS
6223 070742	012701	000060		555\$:	MOV	#TTIVEC,R1	;START OF TTI VECTORS
6224 070746	011137	071506			MOV	(R1),TVSAV2	;SAVE THE CURRENT TTI VECTOR
6225 070752	012721	071000			MOV	#590\$,(R1)	;SET NEW INTERRUPT ROUTINE
6226 070756	011137	071510			MOV	(R1),TPSAV2	;SAVE THE VECTOR PRIORITY
6227 070762	012711	000300			MOV	#PRIC\$,(R1)	;USE PRIORITY SIX
6228 070766					SETPRI	#PRI00	;LOWER INTERRUPT BR LEVEL
070766	012700	000000					MOV
070772	104441						TRAP
6229 070774	000240			560\$:	NOP		;LOOP AWHILE
6230 070776	000776				BR	560\$	;STAY IN "TIGHT" LOOP
6231							
6232							
6233							
6234							
6235 071000	010046			590\$:	MOV	R0,-(SP)	;SAVE WORK REGISTER
6236 071002	113700	177562			MOVB	#TTIBFR,R0	;GET THE OPERATOR INPUT
6237 071006	042700	000200			BIC	#200,R0	;STRIP OFF PARITY BIT
6238 071012	122700	000015			CMPB	#15,R0	;IS IT A CARRIAGE RETURN ?
6239 071016	001075				BNE	591\$	;JUST EXIT IF NOT
6240 071020	012766	066752	000002		MOV	#2\$,2(SP)	;RETURN TO MASTER MENU
6241 071026	005066	000004			CLR	4(SP)	;FORCE PRIORITY 0
6242 071032	013737	071506	000060		MOV	TVSAV2,#TTIVEC ;RESTORE VECTOR	
6243 071040	013737	071510	000062		MOV	TPSAV2,#TTIVEC+2	;RESTORE SUPER PRIORITY
6244 071046	112737	000025	071530		MOVB	#25,T38BS0	;STOP DRIVE TEST 22
6245 071054	112737	000000	071531		MOVB	#0,T38BS1	;CLEAR BS1
6246 071062	012704	071520			MOV	#T38PACKET,R4	;SET UP NEW WRT. SUBSYS MEM. COMMAND
6247 071066	010465	000000			MOV	R4,TSD8(R5)	;SET THE PACKET ADDRESS
6248 071072	012737	176750	071512		MOV	#65000,,T38DLY	;SET UP DELAY COUNTER
6249 071100	004737	016250		592\$:	JSR	PC,WAITF	;DO A WAIT FOR SSR
6250 071104	016501	000002			MOV	TSSR(R5),R1	;CONTENTS OF TSSR REGISTER

TEST 10: MANUAL INTERVENTION

6251	071110	032701	000200		BIT	0SSR,R1		;CHECK FOR TSSR SET	
6252	071114	001017			BNE	595:		;KEEP GOING IF NOT SET	
6253	071116				DELAY	250		;CALL DELAY ROUTINE	
	071116	012727	000250					MOV	0250,(PC)+
	071122	000000						.WORD	0
	071124	013727	002116					MOV	L'DLY,(PC)+
	071130	000000						.WORD	0
	071132	005367	177772					DEC	-6(PC)
	071136	001375						BNE	.-4
	071140	005367	177756					DEC	-22(PC)
	071144	001367						BNE	.-20
6254	071146	005337	071512		DEC	T38DLY		;BUMP COUNTER DOWN	
6255	071152	001352			BNE	592:		;BR, IF MORE TIME LEFT	
6256	071154	004737	016336	595:	JSR	PC,CHKTSSR		;WAIT FOR SSR TO SET	
6257	071160	103405			BCS	580:		;BR IF CARRY SET (GOOD RETURN)	
6258	071162	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR	
6262	071164				ERRDF	ERRNO,T38SSR,PKTSSR		;DEVICE FATAL SSR FAILED TO SET	
	071164	104455						TRAP	C#ERDF
	071166	001773						.WORD	1019
	071170	072716						.WORD	T38SSR
	071172	012046						.WORD	PKTSSR
6263	071174			580:	CKLOOP			;LOOP ON ERROR, IF FLAG SET	
	071174	104406						TRAP	C#CLP1
6264	071176	005737	071504		TST	TTION2		;ARE SUPER INTERRUPTS ENABLED	
6265	071202	001403			BEQ	591:		;BR, IF YES	
6266	071204	042737	000100	177560	BIC	0100,00TTICSR		;RESTORE REGISTER	
6267	071212	012600			MOV	(SP)+,R0		;RESTORE REGISTER	
6268	071214	000002			RTI			;RETURN	
6269	071216			35:					
6270	071216	004737	074046		JSR	PC,T38REST		;SET PACKET TO INITIAL VALUES	
6271	071222	004737	015774		JSR	PC,SOFINIT		;DO SOFT INIT OF CONTROLLER	
6272	071226	103405			BCS	600:		;BR IF SOFT INIT = OK	
6276	071230	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR	
6277	071232				ERRDF	ERRNO,SFIERR,SFIMSG		;DEVICE FATAL ERROR DURING INIT	
	071232	104455						TRAP	C#ERDF
	071234	001774						.WORD	1020
	071236	003652						.WORD	SFIERR
	071240	012034						.WORD	SFIMSG
6278	071242			600:	CKLOOP			;LOOP IF SELECTED	
	071242	104406						TRAP	C#CLP1
6279	071244	012701	071536		MOV	0T38BFR,R1		;ADDRESS OF MESSAGE BUFFER	
6280	071250	012702	125252		MOV	0125252,R2		;ALTERNATING 1'S AND 0'S	
6281									
6282	071254	010221		601:	MOV	R2,(R1)+		;CLEAR OUT THE MESSAGE BUFFER	
6283	071256	022701	072212		CM?	0T38EB,R1		;END OF BUFFER YET	
6284	071262	001401			BEQ	605:		;BR, IF AT END OF BUFFER	
6285	071264	000773			BR	501:		;NOT AT END KEEP GOING	
6286	071266	013737	002202	072240	MOV	UNITN,T38DSW		;SET UNIT NUMBER	
6287	071274	012704	072220		MOV	0T38W2,R4		;SUBROUTINE NEEDS PACKET ADDRESS	
6288	071300	004737	010662		JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS	
6289	071304	103405			BCS	610:		;BR, IF COMMAND ISSUED OK	
6293	071306	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR	
6294	071310				ERRHRD	ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTICS FAILED	
	071310	104456						TRAP	C#ERHRD
	071312	001775						.WORD	1021
	071314	005056						.WORD	WRTMSG
	071316	012034						.WORD	SFIMSG

TEST 10: MANUAL INTERVENTION

[illegible]

TEST 10: MANUAL INTERVENTION

```

6339
6340 071512 000000      T38DLY: .WORD      0          ;DELAY COUNTER FOR TEST
6342      071520      .=<.,+10>E177770
6344 071520      T38PACKET:      ;COMMAND PACKET FOR TEST
6345 071520 140006      .WORD      140006      ;WRITE SUBSYSTEM MEM. CMD. ACK,CVC=1
6346 071522 071530      .WORD      T38TAD      ;ADDRESS OF CHARACTERISTICS BLOCK
6347 071524 000000      .WORD      0
6348 071526 000012      .WORD      10.          ;STARTING /ALUE OF BLOCK SIZE
6349 071530      T38TAD:      ;CHARACTERISTICS DATA BLOCK
6350 071530      T38BS0: .BYTE      0          ;BSEL0 BYTE
6351 071531      T38BS1: .BYTE      0          ;BSEL1 BYTE
6352 071532 000000      T38BS2: .WORD      0          ;BSEL1 WORD
6353 071534 000000      .WORD      0          ;DATA
6354 071536      T38BFR: .BLKW      150.        ;MESSAGE BUFFER
6355 072212 000000      T38EB: .WORD          ;END OF BUFFER ADDRESS
6356
6357
6359      072220      .=<.,+10>E177770
6361 072220      T38PK2:      ;COMMAND PACKET FOR TEST
6362 072220 140004      .WORD      140004      ;WRITE CHARA. MEM. CMND., ACK,CVC=1
6363 072222 072230      .WORD      T38DTA      ;ADDRESS OF SELECT DATA BLOCK
6364 072224 000000      .WORD      0
6365 072226 000012      .WORD      10.          ;STARTING VALUE OF BLOCK SIZE
6366
6367
6368 072230      T38DTA:      ;SELECT DATA BLOCK
6369 072230 071536      .WORD      T38BFR      ;ADDRESS OF MESSAGE BUFFER
6370 072232 000000      .WORD      0
6371 072234 000400      .WORD      256.        ;LENGTH OF MESSAGE BUFFER
6372 072236 000000      T38EAI: .WORD      0          ;EAI BIT WORD
6373 072240 000000      T38DSW: .WORD      0          ;DRIVE SELECT WORD ETC
6375      072250      .=<.,+10>E177770
6377 072250 140212      T38PK3: .WORD      140212 ;MESSAGE BUFFER RELEASE COMMAND
6378 072252 000000      .WORD      0          ;NOT USED
6379 072254 000000      .WORD      0          ;NOT USED
6380 072256 000000      .WORD      0          ;NOT USED
6381 072260 000000      .WORD      0          ;NOT USED
6382
6383      ;WRITE TAPE PACKET
6384
6386      072270      .=<.,+10>E177770
6388 072270 140005      T38PK4: .WORD      140005 ;WRITE, ACK, CVC=1 COMMAND
6389 072272 000000      T38WR: .WORD      0          ;ADDRESS OF WRITE BUFFER
6390 072274 000000      .WORD      0          ;MORE ADDRESS OF WRITE BUFFER
6391 072276 000400      T38SI2: .WORD      256.    ;SIZE OF RECORD
6392
6393
6394
6395
6396
6397      ;*
6398      ;LOCAL TEXT MESSAGES FOR TEST
6399      ;-
6400
6401
6402
6403

```

TEST 10: MANUAL INTERVENTION

```

6404 072300      123      164      141  T38NE:  .ASCIZ  'Stand-alone Manual Intervention Not Executed'
6405 072355      045      116      045  T38MS3: .ASCIZ  'MNA Type <RETURN> To Stop Servo Exerciser, Return To Menu'
6406 072450      124      123      123  T38WRT: .ASCIZ  'TSSR Not Correct After WRITE, With WRITE PROTECT On'
6407 072534      127      122      111  T38WRL: .ASCIZ  'WRITE LOCKED Bit Not Set In XST0'
6408 072575      127      122      111  T38WLE: .ASCIZ  'WRITE LOCK ERROR Bit Not Set In XST0'
6409 072642      127      122      111  T38NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
6410 072716      103      157      156  T38SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
6411 073006      045      116      045  T38INT: .ASCIZ  'MNA Interrupt Received'
6412 073036      045      116      045  T38ONL: .ASCIZ  'MNA Drive Is Now  ON-LINE'
6413 073072      045      116      045  T38OFL: .ASCIZ  'MNA Drive Is Now  OFF-LINE'
6414 073126      103      157      156  T38SS1: .ASCIZ  'Contents Of TSSR Incorrect After MESSAGE BUFFER RELEASE'
6415 073216      045      116      045  T38MS1: .ASCIZ  'MNA Toggle ON-LINE Switch to Generate ATTENTION Interrupts'
6416 073311      045      116      045  T38MS2: .ASCIZ  'MNA Type RETURN To Return To Menu'
6417 073355      111      163      040  T38MSG: .ASCIZ  'Is Write-Protected Tape Mounted'
6418 073415      115      141      156  T38ID:  .ASCIZ  'Manual Intervention'
6419
6420 073442      073466      073540      073566  MIMENU: .WORD    1,2,3,4,5,6
6421 073456      073735      074000      074043  .WORD    8,9,10,0
6422
6423 073466      012      123      105  1:      .ASCIZ  '<12>' SELECT OPERATION FROM FOLLOWING OPTIONS:'
6424 073540      012      011      060  2:      .ASCIZ  '<12>' 0      Display This Menu'
6425 073566      011      061      011  3:      .ASCIZ  '      1      Turn On All M7196 LED's'
6426 073620      011      062      011  4:      .ASCIZ  '      2      Turn Off All M7196 LED's'
6427 073653      011      063      011  5:      .ASCIZ  '      3      Offline/Online Attention'
6428 073707      011      064      011  6:      .ASCIZ  '      4      Write Protect Test'
6429 073735      011      065      011  8:      .ASCIZ  '      5      Print Extended Transport Status'
6430 074000      011      066      011  9:      .ASCIZ  '      6      Return to Diagnostic Supervisor'
6431 074043      000
6432
6433
6434
6435
6436
6437
6438 074044      000000      T38DAT: .WORD    0      ;LOGICAL RESPONSE TO QUESTION
6439 074046      T38REST:
6440 074046      SAVREG      ;SAVE THE REGISTERS
6441 074052      012701      071520      MOV      #T38PACKET,R1      ;START OF THE PACKET
6442 074056      012721      140206      MOV      #140206,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
6443 074062      012721      071530      MOV      #T38TAD,(R1)+      ;ADDRESS OF DATA BLOCK
6444 074066      005021      CLR      (R1)+      ;EXTENDED ADDRESS
6445 074070      012721      000006      MOV      #6,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
6446 074074      005021      CLR      (R1)+      ;CLEAR BSEL0 AND BSEL1
6447 074076      005021      CLR      (R1)+      ;CLEAR SEL2
6448 074100      005011      CLR      (R1)      ;CLEAR DATA AREA
6449 074102      000207      RTS      PC      ;RETURN
6450
6451
6452
6453
6454
6455
6456
6457
6458
6459
6460
;+
;
;THIS ROUTINE PRINTS THE CONTENTS OF
;THE 256 BYTE MESSAGE BUFFER RETURNED BY THE
;TSV-05.
;
;INPUT:
;
;      R0      LOW ORDER ADDRESS OF MESSAGE BUFFER

```

TEST 10: MANUAL INTERVENTION

```

6461      ;      R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
6462      ;      NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
6463      ;
6464      ;
6465      ;
6466      ;
6467 074104 T38MBP:
6468 074104 SAVREG      ;SAVE THE REGISTERS
6469 074110 010005 MOV      R0,R5      ;SAVE LOW ORDER ADDRESS
6470 074112 005737 003134 TST      KTENABLE ;ADDRESS ABOVE 28K?
6471 074116 001001 BNE      910$      ;BR IF YES
6472 074120 005001 CLR      R1      ;SET HIGH ORDER ADDRESS TO 0
6473 074122 010103 910$: MOV      R1,R3      ;SAVE HIGH ORDER ADDRESS
6474 074124 006100 ROL      R0      ;SHIFT BIT15 TO C BIT
6475 074126 006101 ROL      R1      ;SHIFT TO HIGH ORDER FOR PRINTOUT
6476 074130 PRINTX    $T38AS0,R1,R5 ;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
        074130 010546
        074132 010146
        074134 012746 074406
        074140 012746 000003
        074144 010600
        074146 104415
        074150 062706 000010
6477 074154 PRINTX    $T38AS1      ;PRINT HEADER FOR CONTENTS
        MOV      $T38AS1,-(SP)
        MOV      $1,-(SP)
        MOV      SP,R0
        TRAP     C$PNTX
        ADD      $4,SP
        074154 012746 074453
        074160 012746 000001
        074164 010600
        074166 104415
        074170 062706 000004
6478 074174 MOV      R5,R1      ;COPY LOW ORDER ADDRESS
6479 074176 MOV      R3,R0      ;COPY HIGH ORDER ADDRESS
6480 074200 BEQ      913$      ;BR IF NOT ABOVE 28K
6481 074202 004737 017316 JSR      PC,SFTMAP ;SETUP PAR ADDRESS IN R0
6482 074206 010005 MOV      R0,R5 ;GET PAR FORMAT ADDRESS ABOVE 28K
6483 074210 010537 074554 913$: MOV      R5,T38CNT ;HOLD ADDRESS
6484 074214 011504 911$: MOV      (R5),R4 ;GET BUFFER ENTRY
6485 074216 022704 125252 CMP      $125252,R4 ;CHECK FOR NO LOAD CONDITION
6486 074222 001417 BEQ      912$      ;BR, IF BUFFER WASN'T LOADED
6487 074224 010403 MOV      R4,R3      ;MAKE COPY
6488 074226 042704 170377 BIC      $170377,R4 ;ONLY BITS 11,10,9 AND 8 ARE SAVED
6489 074232 000241 CLC      ;CLEAR CARRY
6490 074234 006004 ROR      R4      ;11 TO 10 BIT POSITION
6491 074236 006004 ROR      R4      ;10 TO 9 BIT POSITION
6492 074240 006004 ROR      R4      ;9 TO 8 BIT POSITION
6493 074242 006004 ROR      R4      ;8 TO 7 BIT POSITION
6494 074244 042703 177760 BIC      $177760,R3 ;ONLY BITS 3,2,1 AND 0 ARE SAVED
6495 074250 060403 ADD      R4,R3      ;"OR'EM TOGETHER
6496 074252 010325 MOV      R3,(R5)+ ;PUT BACK IN BUFFER
6497 074254 020527 072212 CMP      R5,$T38EB ;END OF BUFFER YET
6498 074260 001355 BNE      911$      ;BR, IF NOT AT END YET
6499 074262 013705 074554 912$: MOV      T38CNT,R5 ;PUT ADDRESS BACK
6500 074266 012704 000001 MOV      $1,R4      ;START BYTE NUMBER AT ONE
6501 074272 915$: PRINTX    $T38ASN,R4,(R5)+ ;PRT MEM BUFFER W/NEWLINE
        MOV      (R5),-(SP)
        MOV      R4,-(SP)
        MOV      $T38ASN,-(SP)
        MOV      $3,-(SP)
        074272 012546
        074274 010446
        074276 012746 074530
        074302 012746 000003

```

TEST 10: MANUAL INTERVENTION

```

074306 010600
074310 104415
074312 062706 000010
6502 074316 005037 074554
6503 074322 000412
6504 074324
920$: PRINTX 0743ASC,R4,(R5)+
074324 012546
074326 010446
074330 012746 074511
074334 012746 000003
074340 010600
074342 104415
074344 062706 000010
6505 074350 005237 074554
6506 074354 005204
6507 074356 020427 000200
6508 074362 003010
6509 074364 023727 074554 000004
6510 074372 001401
6511 074374 000753
6512 074376 005037 074554
6513 074402 000733
6514 074404 000207
6515
6516 074406 045 116 045 T38AS0: .ASCIZ 'NNA Message Buffer Address * 0105'
6517 074453 045 116 045 T38AS1: .ASCIZ 'NNA Message Buffer Contents:'
6518 074511 045 101 040 T38ASC: .ASCIZ 'A D4A: 03'
6519 074530 045 116 045 T38ASN: .ASCIZ 'NNA ByteD4A: 03'
6520 .EVEN
6521 074554 000000 T38CNT: .WORD ;COUNTER FOR PRINT
6522 074556
074556
074556 104401
L10075: TRAP C$ETST

.SBTTL TEST 11: CONFIGURATION TYPEOUT

;THIS IS A STANDALONE ROUTINE THAT PRINTS OUT ON THE CONSOLE TERMINAL
;THE CONFIGURATION OF THE M7196 MODULE AND TSV5 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

```

TEST 11: CONFIGURATION TYPEOUT

```

6547 ;THE ACTUAL CONFIGURATION AT HAND. IF, FOR EXAMPLE, THE PROGRAM INDICATES
6548 ;THAT IT "SEES" TWO TRANSPORTS CONNECTED WHEN IN FACT ONLY ONE IS PRESENT,
6549 ;THE OPERATOR MUST INTERPRET THIS AS AN ERROR AND ATTEMPT TO FIND THE
6550 ;CAUSE (BAD CABLE, FAULTY UNIT-SELECT DECODING IN THE TRANSPORT, ETC.).
6551 ;[SINCE THE CONTROLLER CAN ONLY ACCESS UNIT 0 IF IT IS IN "STANDARD"
6552 ;MODE, THE PROGRAM WILL FORCE THE MODULE INTO EXTENDED MODE VIA THE
6553 ;WRITE SUBSYSTEM MEMORY COMMAND IN ORDER TO SCAN FOR CONNECTED TRANSPORTS.]
6554 ;
6555 ;
6556 ;THIS ROUTINE, WHEN ITS ACTIONS ARE COMPLETED, WILL EXIT BACK TO THE
6557 ;DIAGNOSTIC SUPERVISOR SO THAT IF ADDITIONAL UNITS (CONTROLLERS) ARE
6558 ;SELECTED (E.G., FROM THE INITIAL STARTUP DIALOG), THE ROUTINE WILL BE
6559 ;REENTERED SO THAT THEIR CONFIGURATIONS CAN BE PRINTED.
6560 ;
6561 BGNTST
6562 ;
6563 ;T11::
6564 RFLAGS RO ;GET OPERATOR FLAGS
6565 ;
6566 ;BR, IF OK TO RUN
6567 ;"TEST NOT EXECUTED"
6568 ;JUMP OUT OF TEST IF NOT
6569 ;TEST ID MESSAGE
6570 ;DO THE COMMCI) SETUP
6571 ;IS MANUAL INTERVENTION ALLOWED?
6572 ;BR, IF MANUAL INTERVENTION ALLOWED
6573 ;JUMP TO OUT IF NOT
6574 ;
6575 ;DO SOFT INIT OF CONTROLLER
6576 ;BR IF SOFT INIT = OK
6577 ;SAVE CONTENTS OF TSSR
6578 ;DEVICE FATAL ERROR DURING INIT
6579 ;
6580 ;LOOP IF SELECTED
6581 ;SET UNIT NUMBER
6582 ;SUBROUTINE NEEDS PACKET ADDRESS
6583 ;ISSUE WRITE CHARACTERISTICS
6584 ;BR, IF COMMAND ISSUED OK
6585 ;SAVE CONTENTS OF TSSR
6586 ;WRITE CHARACTERISTICS FAILED
6587 ;
6588 ;LOOP IF SELECTED
6589 ;GET XST2 STATUS FROM MESSAGE BUFFER
6590 ;"STATE OF EXTENDED FEATURES SW = "
6591 ;
6592 ;CHECK STATE OF E.F.S.

```

TEST 11: CONFIGURATION TYPEOUT

Address	Op-Code	Op-Code Hex	Op-Code Dec	Op-Code Name	Op-Code Comment
6597	074722	001011			
6598	074724			BNE PRINTX	100: 0T39OFF
	074724	012746	077545		
	074730	012746	000001		
	074734	010600			
	074736	104415			
	074740	062706	000004		
6599	074744	000410			
6600	074746			100: BR PRINTX	110: 0T39ON
	074746	012746	077554		
	074752	012746	000001		
	074756	010600			
	074760	104415			
	074762	062706	000004		
6601	074766			110: PRINTX	0T39SBS
	074766	012746	077473		
	074772	012746	000001		
	074776	010600			
	075000	104415			
	075002	062706	000004		
6602	075006	032701	000100		
6603	075012	001011		BIT BNE PRINTX	0BIT6,R1 120: 0T39OFF
6604	075014				
	075014	012746	077545		
	075020	012746	000001		
	075024	010600			
	075026	104415			
	075030	062706	000004		
6605	075034	000410			
6606	075036			120: BR PRINTX	130: 0T39ON
	075036	012746	077554		
	075042	012746	000001		
	075046	010600			
	075050	104415			
	075052	062706	000004		
6607	075056	042701	177700		
6608	075062	010137	077640	130: BIC MOV PRINTX	0177700,R1 R1,T39RL 0T39MCL,T39RL
6609	075066				
	075066	013746	077640		
	075072	012746	077563		
	075076	012746	000002		
	075102	010600			
	075104	104415			
	075106	062706	000006		
6610	075112	004737	015774		
6611	075116	103405			
6615	075120	010001			
6616	075122			JSR BCS MOV ERRDF	PC,SOFINIT 140: R0,R1 ERRNO,SFIERR,SFIMSG
	075122	104455			
	075124	002117			
	075126	003652			
	075130	012034			
6617	075132			140: CKLOOP	
	075132	104406			
6618	075134	013737	002202	MOV	UNITN,T39OSW
6619	075142	012704	076500	MOV	0T39PK2,R4
6620	075146	004737	010662	JSR	PC,WRTCHR

```

;BR, IF EXT. FEA. SW. IS ON
;" OFF"
                                MOV      #T39OFF, -(SP)
                                MOV      #1, -(SP)
                                MOV      SP, R0
                                TRAP     C:PNTX
                                ADD      #4, SP
;SKIP OTHER PRINT STATEMENT
;" ON "
                                MOV      #T39ON, -(SP)
                                MOV      #1, -(SP)
                                MOV      SP, R0
                                TRAP     C:PNTX
                                ADD      #4, SP
;"STATE OF BUFFERING SWITCH ="
                                MOV      #T39SBS, -(SP)
                                MOV      #1, -(SP)
                                MOV      SP, R0
                                TRAP     C:PNTX
                                ADD      #4, SP
;CHECK STATE OF BUFFERING SW
;BR. IF BUFFERING IS ON
;" OFF"
                                MOV      #T39OFF, -(SP)
                                MOV      #1, -(SP)
                                MOV      SP, R0
                                TRAP     C:PNTX
                                ADD      #4, SP
;SKIP OTHER PRINT STATEMENT
;" ON "
                                MOV      #T39ON, -(SP)
                                MOV      #1, -(SP)
                                MOV      SP, R0
                                TRAP     C:PNTX
                                ADD      #4, SP
;ONLY LEAVE MICROCODE REV LEVEL
;LOAD UP REV LEVEL
;"MICROCODE REVISION LEVEL =000XXX"
                                MOV      T39RL, -(SP)
                                MOV      #T39MCL, -(SP)
                                MOV      #2, -(SP)
                                MOV      SP, R0
                                TRAP     C:PNTX
                                ADD      #6, SP
;DO SOFT INIT OF CONTROLLER
;BR IF SOFT INIT = OK
;SAVE CONTENTS OF TSSR
;DEVICE FATAL ERROR DURING INIT
                                TRAP     C:ERDF
                                .WORD    1103
                                .WORD    SFIERR
                                .WORD    SFIMSG
                                ;LOOP IF SELECTED
                                TRAP     C:CLP1
;SET UNIT NUMBER
;SUBROUTINE NEEDS PACKET ADDRESS
;ISSUE WRITE CHARACTERISTICS

```

C3

TSV5 - HARDWARE TESTS MACRO M1200 20-MAR-84 08:26 PAGE 28-94

SEQ 0235

TEST 11: CONFIGURATION TYPEOUT

```

6621 075152 103405      BCS      150:      ;BR, IF COMMAND ISSUED OK
6625 075154 010001      MOV      RO,R1      ;SAVE CONTENTS OF TSSR
6626 075156      ERRHRD  ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTICS FAILED
        075156 104456      TRAP      C:ERHRD
        075160 002120      .WORD     1104
        075162 005056      .WORD     WRTMSG
        075164 012034      .WORD     SFMSG
6627 075166      150:      CKLOOP      ;LOOP IF SELECTED
        075166 104406      TRAP      C:CLP1
6628 075170 005737 002226      TST      EXTFEA      ;CHECK FOR EXTENDED FEATURES SW SWITCH
6629 075174 001036      BNE      174:      ;BR IF SWITCH IS ON
6630 075176 112737 000200 076011      MOVB     #200,T39BS1 ;WRITE MISCELLANEOUS CONT/READ STATUS
6631 075204 112737 000010 076010      MOVB     #10,T39BS0 ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
6632 075212 012704 076000      MOV      #T39PKT,R4 ;WRITE SUBSYS MEM PACKET
6633 075216 010465 000000      MOV      R4,TSD8(R5) ;ISSUE COMMAND
6634 075222 004737 016336      JSR      PC,CHKTSSR ;WAIT FOR SSR
6635 075226 103405      BCS      160:      ;BR, IF NO ERROR
6636 075230 010001      MOV      RO,R1      ;ERROR, SAVE TSSR
6640 075232      ERRHRD  ERRNO,T39NBA,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
        075232 104456      TRAP      C:ERHRD
        075234 002121      .WORD     1105
        075236 077255      .WORD     T39NBA
        075240 012046      .WORD     PKTSSR
6641 075242      160:      CKLOOP      ;LOOP IF SELECTED
        075242 104406      TRAP      C:CLP1
6642 075244 012704 076500      MOV      #T39PK2,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
6643      ;*****
6644      ;
6645      ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
6646      ;
6647      ;*****
6648      ;
6649 075250 004737 010662      JSR      PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
6650 075254 103405      BCS      170:      ;BR, IF COMMAND ISSUED OK
6654 075256 010001      MOV      RO,R1      ;SAVE CONTENTS OF TSSR
6655 075260      ERRHRD  ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTICS FAILED
        075260 104456      TRAP      C:ERHRD
        075262 002122      .WORD     1106
        075264 005056      .WORD     WRTMSG
        075266 012034      .WORD     SFMSG
6656 075270      170:      CKLOOP      ;SCOPE LOOP
        075270 104406      TRAP      C:CLP1
6657 075272 005037 002202      174:      CLR      UNITN ;SET TO DRIVE 0
6658 075276 013737 002202 076520 175:      MOV      UNITN,T39DSW ;SET UNIT NUMBER
6659 075304 012704 076500      MOV      #T39PK2,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
6660 075310 004737 010662      JSR      PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
6661 075314 103405      BCS      180:      ;BR, IF COMMAND ISSUED OK
6665 075316 010001      MOV      RO,R1      ;SAVE CONTENTS OF TSSR
6666 075320      ERRHRD  ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTICS FAILED
        075320 104456      TRAP      C:ERHRD
        075322 002123      .WORD     1107
        075324 005056      .WORD     WRTMSG
        075326 012034      .WORD     SFMSG
6667 075330      180:      CKLOOP      ;LOOP IF SELECTED
        075330 104406      TRAP      C:CLP1
6668      ;
6669 075332 016501 000002      190:      MOV      TSSR(R5),R1 ;GET TSSR STATUS

```

TEST 11: CONFIGURATION TYPEOUT

Address	Op Code	Op Name	Comments
6670	075336	032701	000100
6671	075342	001414	
6672	075344		
	075344	013746	002202
	075350	012746	077014
	075354	012746	000002
	075360	010600	
	075362	104415	
	075364	062706	000006
6673	075370	000137	075724
6674	075374		
	075374	013746	002202
	075400	012746	077060
	075404	012746	000002
	075410	010600	
	075412	104415	
	075414	062706	000006
6675	075420	013701	076024
6676	075424	032701	000004
6677	075430	001013	
6678	075432		
	075432	013746	002202
	075436	012746	077176
	075442	012746	000002
	075446	010600	
	075450	104415	
	075452	062706	000006
6679	075456	000412	
6680	075460		
	075460	013746	002202
	075464	012746	077123
	075470	012746	000002
	075474	010600	
	075476	104415	
	075500	062706	000006
6681	075504	012737	125252 076116
6682	075512	112737	000000 076011
6683	075520	112737	000024 076010
6684	075526	012704	076000
6685	075532	010465	000000
6686	075536	012737	000144 075774
6687	075544	004737	016250
6688	075550	016501	000002
6689	075554	032701	000200
6690	075560	001017	
6691	075562		
	075562	012727	000250
	075566	000000	
	075570	013727	002116
	075574	000000	
	075576	005367	177772
	075602	001375	
	075604	005367	177756
	075610	001367	
6692	075612	005337	075774
6693	075616	001352	
6694	075620	004737	016336

Label	Op Code	Op Name	Comments
	BIT	00FL,R1	
	BEQ	2004	
	PRINTX	0T390F2,UNITN	
2004:	JMP	2504	
	PRINTX	0T390N2,UNITN	
	MOV	T398FR+6,R1	
	BIT	08IT2,R1	
	BNE	2104	
	PRINTX	0T39WPN,UNITN	
2104:	BR	2204	
	PRINTX	0T39WRT,UNITN	
2204:	MOV	0125252,T398FR+100	
	MOVB	00,T398S1	
	MOVB	024,T398S0	
	MOV	0T39PACKET,R4	
	MOV	R4,T39DB(R5)	
	MOV	0100.,T39DLY	
2224:	JSR	PC,WAITF	
	MOV	T39SR(R5),R1	
	BIT	0SSR,R1	
	BNE	2254	
	DELAY	250	
2254:	DEC	T39DLY	
	BNE	2224	
	JSR	PC,CHKTSR	

```

;CHECK FOR OFF-LINE
;BR, IF DRIVE IS ON-LINE
;"DRIVE NUMBER XX IS OFF-LINE"
                                MOV        UNITN,-(SP)
                                MOV        0T39OF2,-(SP)
                                MOV        02,-(SP)
                                MOV        SP,R0
                                TRAP       C$PNTX
                                ADD        06,SP

;DO NOT TRY TO GET ANYMORE INFO.
;"DRIVE NUMBER XX IS ON-LINE"
                                MOV        UNITN,-(SP)
                                MOV        0T39ON2,-(SP)
                                MOV        02,-(SP)
                                MOV        SP,R0
                                TRAP       C$PNTX
                                ADD        06,SP

;READ EXTENDED STATUS (XST0)
;IS DRIVE WRITE PROTECTED
;BR, IF WRITE PROTECTED
;"DRIVE NUMBER IS NOT WRT PRO"
                                MOV        UNITN,-(SP)
                                MOV        0T39WPN,-(SP)
                                MOV        02,-(SP)
                                MOV        SP,R0
                                TRAP       C$PNTX
                                ADD        06,SP

;SKIP OVER
;"DRIVE NUMBER XX IS WRT PRO"
                                MOV        UNITN,-(SP)
                                MOV        0T39WRT,-(SP)
                                MOV        02,-(SP)
                                MOV        SP,R0
                                TRAP       C$PNTX
                                ADD        06,SP

;SET 1 LOC TO KNOWN VALUE
;EXTENDED TAPE STATUS
;EXTENDED TAPE STATUS
;WRITE SUBSYS MEM PACKET
;ISSUE COMMAND
;SET UP DELAY ROUTINE
;WAIT AWHILE FOR SSR TO SET
;SEE IF IT REALLY DID
;JUST CHECK THAT BIT
;BR, IF SSR IS SET
;DELAY ABOUT .25 SEC
                                MOV        0250,(PC)+
                                .WORD      0
                                MOV        L$DLY,(PC)+
                                .WORD      0
                                DEC        -6(PC)
                                BNE        -4
                                DEC        -22(PC)
                                BNE        -20

;START DELAY COUNT DOWN
;BR, IF COUNTER IS NOT AT DONE
;WAIT FOR SSR

```

TEST 11: CONFIGURATION TYPEOUT

[illegible]

TEST 11: CONFIGURATION TYPEOUT

6733	076012	000000		T39BS2:	.WORD	0		;BSEL1 WORD
6734	076014	000000			.WORD	0		;DATA
6735	076016			T39BFR:	.BLKW	150,		;MESSAGE BUFFER
6736								
6737								
6739		076500						
6741	076500			T39PK2:		.=<.,+10>E177770		
6742	076500	140004			.WORD	140004		;COMMAND PACKET FOR TEST
6743	076502	076510			.WORD	T39DTA		;WRITE CHARA. MEM. CMND., ACK,CVC=1
6744	076504	000000			.WORD	0		;ADDRESS OF SELECT DATA ELOCK
6745	076506	000012			.WORD	10.		;STARTING VALUE OF BLOCK SIZE
6746								
6747								
6748	076510			T39DTA:				;SELECT DATA BLOCK
6749	076510	076016			.WORD	T39BFR		;ADDRESS OF MESSAGE BUFFER
6750	076512	000000			.WORD	0		
6751	076514	000400			.WORD	256.		;LENGTH OF MESSAGE BUFFER
6752	076516	000000		T39EAT:	.WORD	0		;EAI BIT WORD
6753	076520	000000		T39DSW:	.WORD	0		;DRIVE SELECT WORD ETC
6755		076530				.=<.,+10>E177770		
6757	076530	140012		T39PK3:	.WORD	140012		;MESSAGE BUFFER RELEASE COMMAND
6758	076532	000000			.WORD	0		;NOT USED
6759								
6760								
6761								
6763		076540				.=<.,+10>E177770		
6765	076540	140005		T39PK4:	.WORD	140005		;WRITE, ACK, CVC=1 COMMAND
6766	076542	000000		T39WR:	.WORD	0		;ADDRESS OF WRITE BUFFER
6767	076544	000000			.WORD	0		;MORE ADDRESS OF WRITE BUFFER
6768	076546	000400		T39SIZ:	.WORD	256.		;SIZE OF RECORD
6769								
6770								
6771								
6772								
6773								
6774								
5775								
6776								
6777								
6778								
6779								
6780	076550	045	116	000	T39NFL:	.ASCIZ	'#N'	
6781	076553	123	164	141	T39NE:	.ASCIZ	'Stand-alone Configuration Typeout Not Executed'	
6782	076632	045	116	045	T39ETS:	.ASCIZ	'#N#A Extended Tape Status Available, Drive Number #D2'	
6783	076721	045	116	045	T39ETN:	.ASCIZ	'#N#A Extended Tape Status NOT Available, Drive Number #D2'	
6784	077014	045	116	045	T39OF2:	.ASCIZ	'#N#A Drive Number #D2#A Is Off-Line'	
6785	077060	045	116	045	T39ON2:	.ASCIZ	'#N#A Drive Number #D2#A Is On-Line'	
6786	077123	045	116	045	T39WRT:	.ASCIZ	'#N#A Drive Number #D2#A Is Write Protected'	
6787	077176	045	116	045	T39WPN:	.ASCIZ	'#N#A Drive Number #D2#A Is NOT Write Protected'	
6788	077255	127	122	111	T39NBA:	.ASCIZ	'WRITE SUBSYSTEM MEMORY Command Not Accepted'	
6789	077331	103	157	156	T39SSR:	.ASCIZ	'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'	
6790								
6791	077421	045	116	045	T39SFS:	.ASCIZ	'#N#A State Of Extended Features Switch ='	
6792	077473	045	116	045	T39SBS:	.ASCIZ	'#N#A State Of Buffering Switch ='	
6793	077545	045	101	040	T39OFF:	.ASCIZ	'#A OFF'	
6794	077554	045	101	040	T39ON:	.ASCIZ	'#A ON'	
6795	077563	045	116	045	T39MCL:	.ASCIZ	'#N#A M7196 Microcode Revision Level =#D2'	

```

;+
;LOCAL TEXT MESSAGES FOR TEST
;-

```

TEST 11: CONFIGURATION TYPEOUT

```

6796
6797 077640 000000      T39RL: .EVEN      0
6798                    .WORD
6799                    .EVEN
6800                    .EVEN
6801
6802      ;*
6803      ;LOCAL STORAGE FOR THIS TEST
6804      ;*
6805 077642 000000      T39DAT: .WORD      0      ;LOGICAL RESPONSE TO QUESTION
6806 077644      T39REST:
6807 077644      SAVREG      ;SAVE THE REGISTERS
6808 077650 012701 076000      MOV      #T39PACKET,R1      ;START OF THE PACKET
6809 077654 012721 140006      MOV      #140006,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
6810 077660 012721 076010      MOV      #T39TAD,(R1)+      ;ADDRESS OF DATA BLOCK
6811 077664 005021      CLR      (R1)+      ;EXTENDED ADDRESS
6812 077666 012721 000006      MOV      #6,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
6813 077672 005021      CLR      (R1)+      ;CLEAR BSEL0 AND BSEL1
6814 077674 005021      CLR      (R1)+      ;CLEAR SEL2
6815 077676 005011      CLR      (R1)      ;CLEAR DATA AREA
6816 077700 000207      RTS      PC      ;RETURN
6817
6818      ;*
6819      ;LOCAL TEXT MESSAGES FOR TEST
6820      ;*
6821
6822 077702      103      157      156      TST39ID:      .ASCIZ 'Configuration Typeout'
6823
6824 077730      .EVEN
6825 077730      ENDTST
6826
6827
6828
6829
6830
6831
6832
6833
6834
6835
6836
6837
6838
6839
6840
6841
6842
6843
6844
6845
6846
6847
6848
6849
6850

```

L10076: TRAP C\$ETST

```

.SBTTL TEST 12: SCOPE LOOPS

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

```

TEST 12: SCOPE LOOPS

```

6851      ;
6852      ;
6853      ;
6854      ;
6855      ;
6856      ;
6857      ;
6858      ;
6859      ;
6860      ;
6861      ;
6862      ;
6863      ;
6864      ;
6865      ;
6866      ;
6867      ;
6868      ;
6869      ;
6870      ;
6871      ;
6872      ;
6873      ;
6874      ;
6875      ;
6876      ;
6877      ;
6878      ;
6879      ;
6880      ;
6881      ;
6882      ;
6883      ;
6884      ;
6885      ;
6886      ;
6887      ;
6888      ;
6889      ;
6890      ;
6891      ;
6892      ;
6893      ;
6894      ;
6895      ;
6896      ;
6897      ;
6898      ;
6899      ;
6900      ;
6901      ;

```

5 TSDB LOW BYTE WRITE ACCESS
6 TSDB MAINTENANCE-MODE WORD WRITE ACCESS
7 TSDBX (TSSR HIGH BYTE) WRITE ACCESS
(EXTENDED FEATURES SWITCH MUST BE ON
TO USE SELECTION CODE 7)
8 EXIT (RETURN TO SUPERVISOR)

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

BGNTST

T12::

RFLAGS RO ;GET OPERATOR FLAGS TRAP C\$RFLA

BEQ 1\$;BR. IF OK TO RUN
MOV #T4ONE,RO ;"TEST NOT EXECUTED"
BR 100\$;JUST EXIT IF NOT
MOV #TST40ID,RO ;TEST ID MESSAGE
JSR PC,TSTSETUP ;DO THE COMMON SETUP
JSR PC,CHKMAN ;SEE IF MANUAL INTERVENTION ALLOWED
BCS 2\$;CARRY SET IF INTERVENTION ALLOWED
JMP 64\$;EXIT IF NO MANUAL INTERVENTION
JSR PC,SOFINIT ;DO A SOFT INIT
BCS 5\$;BRANCH IF OK
MOV RO,R1 ;CONTENTS OF TSSR REGISTER
ERRDF ERRNO,\$FIERR,\$FIMSG ;REPORT FATAL ERROR TRAP C\$ERDF
WORD 1201
WORD \$FIERR
WORD \$FIMSG

5\$: MOV #SCMENU,RO ;MENU OF SCOPE LOOP SELECTIONS
MOV #8,R1 ;MAXIMUM ALLOWED SELECTION
JSR PC,GETSEL ;GO GET THE OPERATORS SELECTION
TST RO ;WAS ZERO SPECIFIED ?
BEQ 2\$;REPEAT MENU IF YES.
CMP RO,#7 ;EXTENDED TSSR ?
BNE 3\$;BRANCH IF NOT
TST EXTFEA ;CHECK FOR EXTENDED FEATURES SET
BNE 3\$;BR. IF IT IS ON
PRINTF #EXFMSG ;WARN OPERATOR EXTENDED FEATURES CLEAR
MOV #EXFMSG,-(SP)
MOV #1,-(SP)
MOV SP,RO
TRAP C\$PNTF
ADD #4,SP

3\$: JMP 2\$;GO BACK TO BASIC MENU
MOV RO,R4 ;SAVE THE MENU SELECTION
SETPRI #PRI06 ;RAISE THE PRIORITY
MOV #PRI06,RO
TRAP C\$SPRI

CLR TTION ;ASSUME INTERRUPTS ARE ENABLED
BIT #100,\$TTICSR ;ARE TTI INTERRUPTS ON ?
BNE 4\$;BRANCH IF YES

TEST 12: SCOPE LOOPS

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

TEST 12: SCOPE LOOPS

```

6957
6958 100320 004737 020174      35$: JSR    PC,GETPAT      ;READ THE DATA PATTERN
6959 100324 010001              MOV    R0,R1      ;DATA PATTERN FOR LOOP
6960 100326 012703 000000      MOV    @TSDB,R3    ;SELECT TSDB
6961 100332 060503              ADD    R5,R3      ;POINT TO TSV05'S REGISTERS
6962 100334 010113      37$: MOV    R1,(R3)      ;WRITE THE DATA PATTERN
6963
6964 100336 000776              BR     37$          ;LOOP UNTIL HALTED
6965
6966 100340 004737 020174      40$: JSR    PC,GETPAT      ;READ THE DATA PATTERN
6967 100344 010001              MOV    R0,R1      ;SAVE THE DATA PATTERN
6968 100346 012703 000003      MOV    @TSSRH,R3    ;BYTE ADDRESS OF TSSR, HIGH BYTE
6969 100352 060503              ADD    R5,R3      ;POINT TO TSV03'S REGISTERS
6970 100354 110113      42$: MOV    R1,(R3)      ;WRITE THE DATA TO REGISTER
6971 100356 000776              BR     42$          ;LOOP UNTIL HALTED
6972
6973
6974
6975      ;*
6976      ;PROCESS CONSOLE INTERRUPTS
6977      ;-
6978 100360 010046              60$: MOV    R0,-(SP)      ;SAVE WORK REGISTER
6979 100362 113700 177562      MOV    @TTIBFR,R0    ;GET THE OPERATOR INPUT
6980 100366 042700 000200      BIC    @200,R0      ;STRIP OFF PARITY BIT
6981 100372 122700 000015      CMPB   @15,R0      ;IS IT A CARRIAGE RETURN ?
6982 100376 001021              BNE    61$          ;JUST EXIT IF NOT
6983 100400 012766 077766 000002      MOV    @2$,2(SP)      ;RETURN TO MASTER MENU
6984 100406 005066 000004      CLR     4(SP)      ;FORCE PRIORITY ZERO
6985 100412 013737 100460 000060      MOV    TVECSAV,@TTIVEC    ;RESTORE SUPERVISOR VECTOR
6986 100420 013737 100462 000062      MOV    TPRISAV,@TTIVEC+2  ;RESTORE SUPERVISOR PRIORITY
6987 100426 005737 100456      TST     TTION      ;ARE SUPERVISOR INTERRUPTS ENABLED ?
6988 100432 001403              BEQ    61$          ;BRANCH IF YES
6989 100434 042737 000100 177560      BIC    @100,@TTICSR    ;TURN OFF TTI INTERRUPTS
6990 100442 012600              61$: MOV    (SP)+,R0    ;RESTORE REGISTER
6991 100444 000002              RTI              ;RETURN FROM INTERRUPT
6992
6993 100446              64$:
6994 100446              63$: EXIT    TST          ;EXIT THE TEST
100446 104432
100450 000736
6995 100452 000137 000200      65$: JMP     200          ;RETURN TO SUPERVISOR
6996
6997
6998      ;*
6999      ;LOCAL STORAGE FOR THIS TEST
7000      ;-
7001 100456 000000      TTION:      .WORD    0      ;WORD SET IF SUPERVISOR TTI INTER OFF
7002 100460 000000      TVECSAV:   .WORD    0      ;SAVE TTI VECTOR
7003 100462 000000      TPRISAV:   .WORD    0      ;SAVE TTI PRIORITY
7004
7005
7006
7007      ;*
7008      ;MENU FOR OPERATOR INPUT FOR SCOPE LOOPS
7009      ;-
7010
7011 100464 100516 100571 100617      .EVEN
SCMENU: .WORD    1$,2$,3$,4$,5$,6$

```

TEST 12: SCOPE LOOPS

```

7012 100500 100770 101026 101074      .WORD  7#,8#,9#,10#,11#,12#,0
7013
7014
7015 100516      012      123      105  1#:  .ASCIZ  <12>'SELECT SCOPE LOOP FROM FOLLOWING OPTIONS:'
7016 100571      012      011      060  2#:  .ASCIZ  <12>' 0      Display This Menu'
7017 100617      011      061      011  3#:  .ASCIZ  '      1      TSBA Read Access'
7018 100643      011      062      011  4#:  .ASCIZ  '      2      TSSR Read Access'
7019 100667      011      063      011  5#:  .ASCIZ  '      3      Initialize (TSSR Write Access)'
7020 100731      011      064      011  6#:  .ASCIZ  '      4      TSDB High Byte Write Access'
7021 100770      011      065      011  7#:  .ASCIZ  '      5      TSDB Low Byte Write Access'
7022 101026      011      066      011  8#:  .ASCIZ  '      6      TSDB Maintenance Mode Write Access'
7023 101074      011      067      011  9#:  .ASCIZ  '      7      TSDBX (TSSR High Byte) Write Access'
7024 101143      011      070      011 10#:  .ASCIZ  '      8      Return to Diagnostic Supervisor'
7025 101206      000      11#:  .ASCIZ  ''
7026 101207      124      171      160 12#:  .ASCIZ  'Type RETURN To Stop Scope Loops'
7027 101247      045      116      045 EXFMSG: .ASCIZ  '***A *** Extended Features Switch Not On *** '
7028 101325      123      164      141 T4ONE: .ASCIZ  'Stand-alone Scope Loops Not Executed'
7029 101372      123      143      157 TST40ID: .ASCIZ  'Scope Loops'
7030
7031 101406      .EVEN
      101406      .ENDTST
      101406      L10077:
      104401      TRAP      C$ETST
7032 101410      .ENDMOD

```

TEST 12: SCOPE LOOPS

```

1          .TITLE   TSV6 - PARAMETER CODING
7
12
18
19 101410      BGNMOD   TSV6
101410      TSV6::
20
21          .SBTTL   HARDWARE PARAMETER CODING SECTION
22
23          ;**
24          ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
25          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
26          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
27          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
28          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
29          ; WITH THE OPERATOR.
30          ;--
31 101410      BGNHRD
101410      .WORD L10100-L$HARD/2
101412      L$HARD::
32
33 101412      GPRMA    HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
101412      .WORD     T$CODE
101414      .WORD     HPM1
101416      .WORD     T$LOLIM
101420      .WORD     T$HILIM
34 101422      GPRMA    HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
101422      .WORD     T$CODE
101424      .WORD     HPM2
101426      .WORD     T$LOLIM
101430      .WORD     T$HILIM
35          GPRMD     HPM3,4,0,340,0,7,YES              ;GET INTERRUPT PRIORITY.
36 101432      ENDRD
          .EVEN
          101432      L10100:
37 101432      104      105      126      HPM1:  .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
38 101466      111      116      124      HPM2:  .ASCIZ  'INTERRUPT VECTOR '
39 101512      111      116      124      HPM3:  .ASCIZ  'INTERRUPT PRIORITY '
40          .EVEN

```

SOFTWARE PARAMETER CODING SECTION

```

42                                     .SBTTL  SOFTWARE PARAMETER CODING SECTION
43
44                                     ;**
45                                     ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
46                                     ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
47                                     ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
48                                     ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
49                                     ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
50                                     ; WITH THE OPERATOR.
51                                     ;--
52 101542                                BGNSFT
53 101542 000003                        .WORD L10101-L$SOFT/2
54 101544                                L$SOFT::
55                                     ; GPRML  SPM1,0,-1,YES           ; GET TRANSPORT TEST FLAG.
56                                     ; GPRML  SPM4,2,-1,YES           ; GET ITERATION CONTROL.
57 101552                                .WORD  T$CODE
58                                     .WORD  SPM4
59                                     .WORD  -1
60                                     ; GPRMD  SPM6,4,D,7777,0,7777,YES ; GET LOCAL ERROR LIMIT
61                                     ; GPRMD  SPM7,6,D,7777,0,7777,YES ; GET GLOBAL ERROR LIMIT
62 101552                                ENDSFT
63                                     .EVEN
64
65                                     L10101:
66 101552 105 116 101 SPM1:  .ASCIZ  'ENABLE TRANSPORT TESTS '
67 101602 111 116 110 SPM4:  .ASCIZ  'INHIBIT ITERATIONS '
68 101632 120 105 122 SPM6:  .ASCIZ  'PER TEST ERROR LIMIT '
69 101662 120 105 122 SPM7:  .ASCIZ  'PER UNIT ERROR LIMIT '
70                                     .SBTTL  PATCH AREA
71
72                                     ;
73                                     ; FINALLY A GENEROUS PATCH AREA.
74                                     ;
75                                     ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
76                                     ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
77                                     ;
78
79 101712                                PATCH::
80                                     .BLKW  32.
81                                     ., 1377+1
82 102400 102400                        LASTAD  ;SET LAST USED ADDRESS.
83 102402 000000                        .EVEN
84 102404 000000                        .WORD  0
85                                     .WORD  0
86                                     LAST::
87                                     END 100
88                                     .E

```


SYMBOL TABLE

LOE	040000	G	L#UNIT	002012	G	L10071	056560	OFL	000100	PRMNO	002320	G
LOOPCN	002216	G	L10000	002164	L10072	060024	ONEFIL	000000	PRMSG	014552	G	
LOOPCO	013126	L10001	002176	L10073	066714	O#APTS	000000	PRMSG0	014732			
LOOPFL	003162	G	L10002	005766	L10074	065054	O#AU	000001	PRMSG1	014777		
L#OT	000010	G	L10003	012044	L10075	074556	O#BGNR	000001	PRMSG2	015035		
L#ACP	002110	G	L10004	012062	L10076	077730	O#BGNS	000001	PROASC	014420		
L#APT	002036	G	L10005	012100	L10077	101406	O#DU	000001	PRIASC	014465		
L#AU	022400	G	L10006	012106	L10100	101432	O#ERRT	000000	PST32W	003152	G	
L#AUT	002070	G	L10007	012124	L10101	101552	O#GNSW	000001	PUNIT	022332		
L#AUTO	022604	G	L10010	012142	MEMADD	013754	O#POIN	000001	PW.D11	000021		
L#CCP	002106	G	L10011	012166	MEMCK	021276	O#SETU	000000	PW.D13	000022		
L#CLEA	022664	G	L10012	012240	MENASC	020447	PASRPT	022102	PW.D22	000020		
L#CO	002032	G	L10013	012410	MENERR	020374	PATCH	101712	PW.NOP	000000		
L#DEPO	002011	G	L10014	013124	MENRES	020476	PATDAT	020230	PW.NO1	000023		
L#DESC	003410	G	L10015	013752	MIMENU	073442	PC.ERA	002400	PW.RDE	000024		
L#DESP	002076	G	L10016	013774	MMVEC	000250	PC.IER	002000	PW.RDR	000001		
L#DEVP	002060	G	L10017	015500	MSA.FR	000006	PC.NOO	001000	PW.RDS	000005		
L#DISP	002124	G	L10020	015506	MSA.NO	000000	PC.REL	000000	PW.RFI	000003		
L#DLY	002116	G	L10021	015514	MSA.NR	000004	PC.REW	000400	PW.WCT	000006		
L#DTP	002040	G	L10022	015526	MSA.VO	000002	PKBCNT	000006	PW.WFI	000004		
L#DTYP	002034	G	L10023	015530	MSGEXP	012144	PKHI	000004	PW.WFM	000007		
L#DU	022476	G	L10024	015576	MSGLOO	013064	PKLOW	000002	PW.WHI	000010		
L#DUT	002072	G	L10025	015736	MSGSTA	012350	PKTADD	007554	PW.WNP	000011		
L#DVTY	003402	G	L10026	016246	MSGSUB	013742	PKTFRM	007516	PW.WTR	000002		
L#EF	002052	G	L10030	022330	MS.ATT	000006	PKTGET	012064	P.ACK	100000		
L#ENVI	002044	G	L10031	022474	MS.EXT	000200	PKTMES	012110	P.CMD	000037		
L#ETP	002102	G	L10032	022602	MS.RSD	000001	PKTRAM	004745	P.CONT	000012		
L#EXP1	002046	G	L10033	022662	MS.RSF	000020	PKTSSR	012046	P.CVC	040000		
L#EXP4	002064	G	L10034	022710	MS.RST	000010	PNT	001000	P.FMT	000140		
L#EXP5	002066	G	L10035	023152	M8186	005554	PRAMPK	013776	P.FORM	000011		
L#HARD	101412	G	L10036	024454	M8189	005645	PRASC	014523	P.GETS	000017		
L#HIME	002120	G	L10037	026446	NBA	002000	PRBEXP	015470	P.IE	000200		
L#HPCP	002016	G	L10040	024730	NEWPAS	022036	PRBMSG	015336	P.INIT	000013		
L#HPTP	002022	G	L10041	025174	NODEV	003114	PRBREC	015472	P.MODE	007400		
L#HW	002156	G	L10042	032042	NOEXTF	030236	PRBTOT	015423	P.OPP	020000		
L#ICP	002104	G	L10043	027022	NOINIT	004335	PRBYTE	015122	P.POSI	000010		
L#INIT	021556	G	L10044	027312	NOINTR	004221	PRI	002000	P.READ	000001		
L#LADP	002026	G	L10045	027610	NOITS	002170	PRIADD	010160	P.SWB	010000		
L#LAST	102404	G	L10046	030242	NOMAN	020534	PRIAO	010230	P.WRIT	000005		
L#LOAD	002100	G	L10047	034632	NOMEM	005460	PRIBXO	007612	P.WATC	000004		
L#LUN	002074	G	L10050	032376	NP.IR	000200	PRIEQU	010060	P.WRTS	000006		
L#PREV	002050	G	L10051	032770	NP.LOO	000040	PRIPKT	007370	QVP	002204	G	
L#NAME	002000	G	L10052	033376	NP.OUT	000100	PRIRAM	010066	RAMASC	014156		
L#PRIO	002042	G	L10053	040424	NP.WRP	000020	PRITAD	010274	RAMDAT	002242	G	
L#PROT	021546	G	L10054	035716	NSI	004152	PRITSS	006024	RAMERR	015510	G	
L#PRT	002112	G	L10055	036650	NSINIT	004407	PRITO	010356	RAMEXP	015530	G	
L#REPP	002062	G	L10056	050536	NUL	004527	PRITI	010421	RAMFOR	010116		
L#REV	002010	G	L10057	040730	NULCR	004530	PRIXOR	007742	RAMSIZ	002302	G	
L#RPT	022712	G	L10060	042140	NXM	004000	PRIOO	000000	RAMTAD	015516	G	
L#SOFT	101544	G	L10061	043500	NXMFLG	003136	PRI01	000040	RCVHIA	002304	G	
L#SPC	002056	G	L10062	044056	NXMHI	003142	PRI02	000100	RCVLOA	002306	G	
L#SPCP	002020	G	L10063	045330	NXMLO	003140	PRI03	000140	RDERR	005206		
L#SPTP	002024	G	L10064	046374	NXMTST	021452	PRI04	000200	RECMG	002466	G	
L#STA	002030	G	L10065	052016	NXR	003740	PRI05	000240	RECV	002234	G	
L#SW	002166	G	L10066	062644	NXRERR	005736	PRI06	000300	REGSAV	020140		
L#TEST	002114	G	L10067	054022	NXRX	003777	PRI07	000340	RETEHR	005372		
L#TIML	002014	G	L10070	055310	NXTU	022050	PRMESS	014242	REWIND	011014	G	

C4

SEQ 0248

SYMBOL TABLE

RMCHBE=	000167	S1.IID=	004000	TST40I	101372	T10	066716	G	T15BFR	033442		
RMCHEN=	000200	S1.IIR=	020000	TSV2	002000	G	T11	074560	G	T15BF2	034130	
RMMSGB=	000215	S1.I2R=	040000	TSV3	002176	G	T12	077732	G	T15BS0	034130	
RMMSGC=	000234	S1.PAR=	100000	TSV4	021546	G	T12BFR	030332		T15BS1	034131	
RMPKTB=	000201	S2.ATI=	000010	TSV5	023474	G	T12BKG	031007		T15DAT	033430	
RMPKTE=	000210	S2.BTI=	000004	TSV6	101410	G	T12BLK	030364		T15L00	032074	
RMR	010000	S2.DIM=	000200	TTIBFR=	177562	G	T12CHA	032000		T15PAC	033420	
RWPACK	011110	S2.IIW=	000100	TTICSR=	177560	G	T12CKR	031560	G	T15PK2	034120	
SC	100000	S2.INR=	000020	TTION	100456		T12CON	031366		T15RES	034522	
SCE	020000	S2.OUT=	000040	TTION2	071504		T12DAT	030320		T15RT2	034574	
SCHEER	005300	S2.UND=	000003	TTIVEC=	000060	G	T12DPR	031166		T15SSR	034136	
SCME	005013	TBLEND=	003062	G	TVCSA	100460	T12GET	030546		T15S2	034132	
SCMENU	100464	TCOASC	006476	TVSAV2	071506		T12HIA	030352		T15S3	034134	
SDELAY	010660	TCOCOD	006676	TARGC=	000001		T12KT	030360		T16BEN	040310	
SELASC	020442	TEMP1	003116	G	TICDCE=	001130	T12LOA	030354		T16BFR	040262	
SELDAT=	000004	TEMP2	003120	G	TIERRN=	002261	T12LOO	026476		T16BFS	040302	
SEL2	000002	TERCLS=	000016		TLEXCP=	000000	T12MSG	030711		T16CLR	040126	
SETMAP	017316	TESTNO=	000014		TFLAG=	000040	T12NIN	031075		T16DAT	040250	
SETU	022134	TEXASC	006435		TIGMAN=	000000	T12NXH	031257		T16DT2	040320	
SFFMSG	012102	TFCASC	006537		TIHILI=	000776	T12PAC	030310		T16D01	036702	
SFHERR	003705	TIMEXP	015552	G	TILAST=	000001	T12PAR	030356		T16D28	036704	
SFIERR	003652	TIMSGO	015600		TILOLI=	000000	T12SET	031736		T16D53	036706	
SFIMSG	012034	TINERR	012021		TILSYM=	010000	T12SWR	031670		T16D78	036710	
SFPTBL	002166	TMPBFR	002632	G	TILTNO=	000014	T12TBE	030516		T16LEN	037232	
SIFLAG	003154	TNAM	016704		TINEST=	177777	T12WRT	030622		T16L00	034652	
SIMSG	011766	TPRISA	100462		TINSO =	000000	T12L0	026610		T16PAC	040240	
SKIPT	003400	TPSAV2	071510		TINS1 =	000005	T12L0	027144		T16PK2	040310	
SOFINI	015774	TRANST	002166	G	TINS2 =	000002	T123L0	027364		T16REJ	037344	
SPACE	010466	TSBA	000000	G	TINS3 =	000003	T124L0	030006		T16RES	040060	
SPM1	101552	TSBAH =	000001	G	TPTNU=	000000	T124TS	030362		T16SEX	040220	
SPM4	101602	TSDB =	000000	G	TSAVL =	177777	T13BFR	024112		T16SRD	040152	
SPM6	101632	TSDBH =	000001	G	TSEGL =	177777	T13DAT	024100		T16SSR	036762	
SPM7	101662	TSFCOD	007236		TSEKO =	010000	T13LOO	023512		T16TSB	037130	
SRO	177572	TSREJ =	000006		TSUBN=	000000	T13MEM	024205		T16T01	037461	
SR1	177574	TSSDEF	006606		TTAGL =	177777	T13NBA	024244		T16T28	037560	
SR2	177576	TSSR =	000002	G	TTAGN =	010102	T13PAC	024070		T16T53	037660	
SR3	172516	TSSRBI	003502	G	TTEMP =	000000	T13RES	024406		T16T78	037760	
SSR	000200	TSSRFO	006415		TTEST =	000014	T13SSR	024317		T16WMI	040172	
STATCO	012412	TSSRH =	000003	G	TTSTM =	177777	T14BFR	025226		T162SS	037017	
SVCGBL=	000000	TSSX	004020		TITSTS=	000001	T14BS0	025220		T163SS	037063	
SVCINS=	000000	TSTBLK	002752	G	TIAU =	010031	T14BS1	025221		T17BEN	050412	
SVCSUB=	000001	TSTCNT	002214	G	TIAUT =	010033	T14BS2	025222		T17BFR	050272	
SVCTAG=	000000	TSTEND	016720		TICLE =	010034	T14DAT	025220		T17BFS	050312	
SVCTST=	000001	TSTFLA	002314	G	TIDU =	010032	T14DTA	025640		T17CLE	050176	
SILSYM=	010000	TSTL00	016456	G	TIHAR =	010100	T14L00	024474		T17CLR	050010	
SO.IDB=	000010	TSTPTR	002316	G	TIHW =	010000	T14NBA	025652		T17DAT	050260	
SO.IFB=	000002	TSTSET	016510	G	TIINI =	010030	T14NIN	026113		T17DT2	050430	
SO.IFP=	000001	TST12I	030520		TIIMSG=	010025	T14PAC	025210		T17EXE	046554	
SO.ILD=	000020	TST13I	024132		TIPRO =	010027	T14PK2	025630		T17EXP	046432	
SO.ION=	000040	TST14I	026271		TIRPT =	010035	T14RES	026336		T17EXS	046452	
SO.IRD=	000100	TST15I	034503		TISEG =	010000	T14RST	026374		T17L00	040454	
SO.IRW=	000004	TST16I	036712		TISOF =	010101	T14SSR	026023		T17MSK	046426	
SO.ISP=	000200	TST17I	046656		TISRV =	010026	T14TSB	026205		T17PAC	050250	
S1.ICE=	002000	TST18I	051246		TIISUB=	010074	T142RE	025726		T17PK2	050420	
S1.IE0=	010000	TST19I	060232		TIISW =	010001	T15AM2	034212		T17RFI	050156	
S1.IFM=	001000	TST20I	065232		TIITES=	010077	T15AM3	034313		T17RSF	050054	
S1.IHE=	000400	TST39I	077702		T1	023474	G	T15AM4	034415		T17SET	050216

SYMBOL TABLE

T17SNP	050076	T19SSR	060273	T3	026450 G	T39NBA	077255	WF.IED	000010
T17SRD	050034	T19WCT	062172	T3BFLG	003150 G	T39NE	076553	WF.IER	000004
T17SSR	046675	T19WFI	062136	T3.1	026476	T39NFL	076550	WF.IHI	000200
T17WFD	046554	T19WFH	062212	T3.2	027036	T39OFF	077545	WF.IRE	000040
T17WFI	050122	T19WST	061573	T3.3	027326	T39OF2	077014	WF.IWF	000020
T171CM	047215	T191CM	060730	T3.4	027624	T39ON	077554	WF.IWR	000100
T172CM	047277	T192CM	061012	T38ASC	074511	T39ON2	077060	WF.I3R	000002
T172SS	046732	T192SS	060330	T38ASN	074530	T39PAC	076000	WF.I4R	000001
T173CM	047373	T193CM	061106	T38AS0	074406	T39PK2	076500	WRTCHR	010662 G
T173SS	046776	T193SS	060374	T38AS1	074453	T39PK3	076530	WRTERR	005113
T174CM	047457	T194SS	060441	T38BFR	071536	T39PK4	076540	WRTMSG	005056
T174SS	047043	T195CM	061174	T38BS0	071530	T39RES	077644	WSMBK	021270 G
T175CM	047542	T195SS	060504	T38BS1	071531	T39RL	077640	XFERAS	015740
T175SS	047106	T196CM	061250	T38BS2	071532	T39SBS	077473	XNXM	016376
T176CM	047616	T196SS	060550	T38CNT	074554	T39SFS	077421	XORBF0	007674
T176SS	047152	T197CM	061333	T38DAT	074044	T39SI2	076546	XORFOR	010012
T177CI	047724	T197SS	060613	T38DLY	071512	T39SSR	077331	XST0	000006 G
T18BFR	051752	T198CM	061421	T38DSW	072240	T39TAD	076010	XST1	000010 G
T18CMP	051453	T198SS	060662	T38DTA	072230	T39WPN	077176	XST2	000012 G
T18DAT	051740	T199CM	061510	T38EAI	072236	T39WR	076542	XST3	000014 G
T18DT2	052010	T2	024456 G	T38EB	072212	T39WRT	077123	XST4	000015 G
T18EXP	051216	T2.1	024474	T38ID	073415	T4	032044 G	XSOBOT	000002
T18LON	050556	T2.2	024744	T38INT	073006	T4.1	032074	XSOEOT	000001
T18MSK	051212	T20BEN	066572	T38MBP	074104	T4.2	032400	XSOIE	000040
T18PAC	051730	T20BFR	066452	T38MSG	073355	T4.3	032772	XSOILA	000400
T18PK2	052000	T20BFS	066472	T38MS1	073216	T4ONE	101325	XSOILC	001000
T18SET	051620	T20CLE	066364	T38MS2	073311	T5	034634 G	XSOLET	020000
T18SMI	051576	T20CLR	066152	T38MS3	072355	T5.1	034652	XSOMOT	000200
T18SRD	051556	T20CLP	066057	T38NBA	072642	T5.2	035732	XSONEF	002000
T18SSR	051305	T20DAT	066440	T38NE	072300	T6	040426 G	XSOONL	000100
T18XS	051242	T20DT2	066610	T38OFL	073072	T6.1	040454	XSOPED	000010
T182SS	051342	T20EXE	065232	T38ONL	073036	T6.2	040744	XSORLL	010000
T183SS	051406	T20EXP	065112	T38PAC	071520	T6.3	042154	XSORLS	040000
T18BEN	062522	T20EXS	065132	T38PK2	072220	T6.4	043514	XSOYMK	100000
T18BFC	060046	T20L00	062664	T38PK3	072250	T6.5	044072	XSOVCK	000020
T18BFR	062402	T20MSK	065106	T38PK4	072270	T6.6	045344	XSOVLE	004000
T18BFS	062422	T20PAC	066430	T38RES	074046	T7	050540 G	XSOVWK	000004
T19CLE	062250	T20PK2	066600	T38SI2	072276	T8	052020 G	XXCOMM	003122 G
T19CLH	062004	T20RFI	066236	T38SSR	072716	T8.1	052036	X4ALWA	000000
T19CNV	062314	T20RSF	065756	T38SST	073126	T8.2	054036	X4FALS	000040
T19DAT	062370	T20SET	066404	T38TAD	071530	T8.3	055324	X4OFFS	000400
T19DT2	062540	T20SEX	066346	T38WLE	072575	T8.4	056574	X4TRUE	000020
T19EXE	060232	T20SRD	066176	T38WR	072272	T9	062646 G	X1.COR	020000
T19EXP	060112	T20SSR	065261	T38WRL	072534	T9.1	062664	X1.DLT	100000
T19XS	060132	T20SWP	065647	T38WRT	072450	UAM	000200 G	X1.MBZ	017375
T19L00	052036	T20WFI	066256	T38WFR	076016	UNJTN	002202 G	X1.RBP	000400
T19MSK	060106	T20WFH	066312	T38BS0	076010	UNREC	000006	X1.SPA	040000
T19PAC	062360	T20WHI	066332	T38BS1	076011	USI	004123	X1.UNC	000002
T19PK2	062530	T20WNP	066216	T38BS2	076012	WAITF	016250 G	X2.BUF	000100
T19PRE	060044	T202SS	065316	T39DAT	077642	WC.IFA	000200	X2.EXT	000200
T19RFI	062116	T203SS	065362	T39DLY	075774	WC.IFE	000002	X2.OPM	100000
T19RSF	062050	T204SS	065427	T39DSW	076520	WC.IGO	000001	X2.RCE	040000
T19RST	061700	T205SS	065472	T39DTA	076510	WC.IRE	000010	X2.REV	000077
T19SET	062270	T206SS	065536	T39EAI	076516	WC.IRW	000004	X2.SPA	035400
T19SEX	062232	T208SS	065601	T39ETN	076721	WC.IOT	000100	X2.UNI	000007
T19SNP	062072	T23A	003144 G	T39ETS	076632	WC.IIT	000040	X2.WCF	002000
T19SRD	062030	T23B	003146 G	T39MCL	077563	WC.I5R	000020	X3.DCK	000010

....B1
....C1
....D1
....E1
....F1
....G1
....H1
....I1
....J1
....K1
....L1
....M1
....N1

....B2
....C2
....D2
....E2
....F2
....G2
....H2
....I2
....J2
....K2
....L2
....M2
....N2

....B3
....C3
....D3
....E3
....F3
....G3
....H3
....I3
....J3
....K3
....L3
....M3
....N3

....B4
....C4
....D4
....E4