

.REM_
IDENTIFICATION

PRODUCT ID: AC-T718A-MC
PRODUCT TITLE: CZTSBA0 TSU05 DIAG PART 2
DEPARTMENT: COMPUTER SPECIAL SYSTEMS/PPG
DATE: JUNE 06, 1983

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) 1983 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	MANUALS
2.2	SWITCHES
2.3	PLUGS
2.4	HARDWARE CONNECT
2.5	SOFTWARE
2.6	EXTENDED TABLE DIALOGUE
2.7	QUICK SETUP PROCEDURE
3.0	ERROR INFORMATION
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES
7.0	MAINTENANCE HISTORY

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS IS A PDP-11 RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF A TSU05 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A PDP-11 SYSTEM (UNIBUS). THE PROGRAM PROVIDES ERROR MESSAGES WHICH IDENTIFY FAILING FUNCTIONS THAT AID IN THE REPAIR OF THE DEVICE. THIS DIAGNOSTIC CONSIST OF TWELVE TEST. TEST 1-9 ARE EXECUTED IN SEQUENCE. TEST 10-12 ARE STAND ALONE TEST WHICH ALLOW THE OPERATOR TO PERFORM SPECIFIC FUNCTIONAL TEST ON SCOPE LOOPS ON CERTAIN FUNCTIONS.

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

1.2 SYSTEM REQUIREMENTS

PDP-11 PROCESSOR AND MEMORY
CAUTION: DIAGNOSTIC REQUIRES 32K WORDS OF MEMORY
(28K USEABLE AND 4K RESERVED FOR I/O PAGE)
TSU05 MAGTAPE SUBSYSTEM (DRIVE AND CONTROLLER)
CONSOLE TERMINAL
PDP-11 DIAGNOSTIC SUPERVISOR (HSAAA.SYS VERSION 34 OR LATER)
PDP-11 DIAGNOSTIC LOADER/MONITOR (XXDP+)

1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

1. CHQUS XXDP+ USERS GUIDE; DOCUMENT NUMBER AC-F348E-MC
DATE: 14 JULY 1980.
2. TSU05 TRANSPORT SUBSYSTEM USER'S GUIDE; DOCUMENT NUMBER EK-TSU05-UG-001
DATE: AUGUST 1982
3. TSU05 TRANSPORT SUBSYSTEM TECHNICAL MANUAL; DOCUMENT NUMBER EK-TSU05-TM-001
DATE: AUGUST 1982
4. TSU05 TRANSPORT SUBSYSTEM INSTALLATION MANUAL; DOCUMENT NUMBER EK-TSU05-IN-001
DATE: AUGUST 1982

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

FUNCTIONAL PDP-11 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.
CZTSAA HAS RUN SUCCESSFULLY.

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

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

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

2.1.1 OPERATOR COMMANDS

THE TS005 DIAGNOSTIC IS A PDP-11 DIAGNOSTIC SUPERVISOR COMPATIBLE PROGRAM. ALL LOADING AND RUNTIME INSTRUCTIONS CAN BE REFERENCED IN THE CHQUS XXDP+ USERS GUIDE, DOCUMENT NUMBER AC-F348E-MC. THE USER ENTRY IS IN QUOTES.

BOOT THE DIAGNOSTIC MEDIA

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

*****TSU05 LOGIC DIAGNOSTIC*****
 UNIT IS TSU05
 >DR

2.2 SWITCHES

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

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

EXAMPLE OF SWITCH USAGE:

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

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

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

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

2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
LOE	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)
FRI	DIRECT MESSAGES TO LINE PRINTER
INT	PRINT TEST NUMBER AS TEST EXECUTES
BOE	"BELL" ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE
LOT	LOOP ON TEST

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

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

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

2.4 HARDWARE QUESTIONS

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

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

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

TSBA/TSDB = 172520, VECTOR = 224

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

UNITS (D) ? <ENTER THE NUMBER OF M7455 CONTROLLERS
PRESENT TO BE TESTED>

UNIT 0

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

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

THE ADDRESS AND VECTOR QUESTIONS WILL BE ASKED FOR EACH OF THE NUMBER OF UNITS (CONTROLLERS) SPECIFIED IN THE "# UNITS?" QUESTION. LOGICAL UNIT NUMBERS ARE ASSIGNED IN ORDER, BEGINNING AT 0. UP TO FOUR UNITS CAN BE SELECTED FOR TESTING AS FOLLOWS:
UP TO 4 TSU05 CONTROLLERS PER 11 AND UP TO 2 DRIVES PER CONTROLLER

2.5 SOFTWARE QUESTIONS

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

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

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

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

2.6 EXTENDED P-TABLE DIALOGUE

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

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

UNITS (0) ? 8<CR>

UNIT 1

CSR ADDRESS (0) ? 160000<CR>

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

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

UNIT 2

CSR ADDRESS (0) ? 160000<CR>

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

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

UNIT 3

CSR ADDRESS (0) ? 160000<CR>

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

Q-FACTOR (0) 0 ? <CR>

UNIT 4

CSR ADDRESS (0) ? 160000<CR>

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

Q-FACTOR (0) 0 ? <CR>

UNIT 5

CSR ADDRESS (0) ? 160000<CR>

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

Q-FACTOR (0) 0 ? <CR>

UNIT 6

CSR ADDRESS (0) ? 160000<CR>

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

Q-FACTOR (0) 0 ? <CR>

UNIT 7

CSR ADDRESS (0) ? 160000<CR>

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

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

UNIT 8

CSR ADDRESS (0) 160000<CR>

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

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

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

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

```
# UNITS (0) ? 8<CR>

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

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

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

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

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

```
# UNITS (0) ? 8<CR>

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

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

2.7 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

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

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

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

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

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

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

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

EXTENDED ERROR MESSAGES CONTAIN SUPPLEMENTARY ERROR INFORMATION SUCH AS REGISTER CONTENTS OR GOOD/BAJ 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
CZTSB 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> IOML<5> IFBY<1>
IRE3V1<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.

CZTSB 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 (PDP-11)

DR>STA/FLA:PNT:H0E:UAM

UNITS (0) ? 1

UNIT 0

DEVICE ADDRESS (0) 172520 ? <CR>

VECTOR (0) 224 ? <CR>

CHANGE SW (L) ? N<CR>

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

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

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

0 ERRORS

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

PROGRAM RUN TIMES

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

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

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

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

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

5.0 DEVICE INFORMATION TABLES

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

CHANGE HW (L) ?

* UNITS (D) ? <ENTER THE NUMBER OF M7455 CONTROLLERS
PRESENT TO BE TESTED>

UNIT 0

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

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

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

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

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

CHANGE SW (L) ?

6.0 TEST SUMMARIES

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

TEST DESCRIPTION:

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

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

THIS TEST VERIFIES THAT THE WRITE SUBSYSTEM MEMORY COMMAND WITH A 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 (DATC). 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 11B SYSTEM WITH MORE THAN 128K WORDS OF MEMORY!

TEST 4: RAM EXERCISER TEST

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

TEST 5: EXTENDED FEATURES SWITCH AND TIMERS A,B

TEST DESCRIPTION:

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

TEST 6: FIFO EXERCISER

TEST DESCRIPTION:

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

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

TEST DESCRIPTION:

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

TEST 8: TRANSPORT BUS INTERFACE LOOPBACK TEST

TEST DESCRIPTION:

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

TEST 9: READ/WRITE DATA PARITY TEST

TEST DESCRIPTION:

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

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

TEST 10: MANUAL INTERVENTION

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

CODE	ROUTINE
0	HELP. PRINTS THIS MENU.
1	TURN ON ALL M7455 LED INDICATORS
2	TURN OFF ALL M7455 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 M7455 MODULE AND TSU05 SUBSYSTEM. SPECIFICALLY, THE FOLLOWING INFORMATION IS PRESENTED:

- 1.0 STATE OF THE EXTENDED FEATURES SWITCH ON THE M7455: ON (EXTENDED FEATURES ENABLED) OR OFF (EXTENDED FEATURES DISABLED).
- 2.0 STATE OF THE BUFFERING ENABLE SWITCH ON THE M7455: ON (BUFFERING ENABLED) OR OFF (BUFFERING DISABLED).
- 3.0 MICROCODE REVISION LEVEL OF THE M7455.
- 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 M7455 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 0-277, TYPING <RETURN> CAUSES AN EXIT FROM THE SCOPE LOOP BACK TO MENU LEVEL.

7.0 MAINTENANCE HISTORY

REVISION A - JUNE 1983

```

1          .TITLE   TSV2 - PROGRAM HEADER
2          .SBTTL   PROGRAM HEADER
3 000000    .PSECT   ABS
4
10         .MCALL   SVC
11 000000    SVC                      ; INITIALIZE SUPERVISOR MACROS
12         .ENABLE  LC
13         .NLIST   BEX,CND
19         .ENABL   AMA
20         .+2000
21 002000    002000'
22         BGNMOD   TSV2
23
24         ;**
25         ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
26         ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
27         ;--
28
29 002000    POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
30 002000    HEADER  CZTSB,A,0,655..0
          L$NAME:: ;DIAGNOSTIC NAME
          .ASCII /C/
          .ASCII /Z/
          .ASCII /T/
          .ASCII /S/
          .ASCII /B/
          .BYTE  0
          .BYTE  0
          .BYTE  0
          L$REV:: ;REVISION LEVEL
          .ASCII /A/
          L$DEPO:: ;0
          .ASCII /0/
          L$UNIT:: ;NUMBER OF UNITS
          .WORD  0
          L$TIML:: ;LONGEST TEST TIME
          .WORD  655.
          L$HPCP:: ;POINTER TO H.W. QUES.
          .WORD  L$HARD
          L$SPCP:: ;POINTER TO S.W. QUES.
          .WORD  L$SOFT
          L$HPTP:: ;PTR. TO DEF. H.W. PTABLE
          .WORD  L$HW
          L$SPTP:: ;PTR. TO S.W. PTABLE
          .WORD  L$SW
          L$LADP:: ;DIAG. END ADDRESS
          .WORD  L$LAST
          L$STA:: ;RESERVED FOR APT STATS
          .WORD  0
          L$CO::
          .WORD  0
          L$DTYP:: ;DIAGNOSTIC TYPE
          .WORD  0
          L$APT:: ;APT EXPANSION
          .WORD  0
          L$DTP:: ;PTR. TO DISPATCH TABLE

```

TSV2 - PROGRAM HEADER MACRO M1113 01-FEB-84 17:02

SEQ 019

```

002040 002124'      L$PRIO:: .WORD L$DISPATCH
002042              ;DIAGNOSTIC RUN PRIORITY
002042 000000      L$ENVI:: .WORD 0
002044              ;FLAGS DESCRIBE HOW IT WAS SETUP
002044 000000      L$EXP1:: .WORD 0
002046              ;EXPANSION WORD
002046 000000      L$MREV:: .WORD 0
002050              ;SVC REV AND EDIT #
002050 003          C$REVISION
002051 003          C$EDIT
002052              L$EF:: .WORD 0
002052 000000      ;DIAG. EVENT FLAGS
002054 000000      .WORD 0
002056              L$SPC:: .WORD 0
002056 000000      .WORD 0
002060              L$DEVP:: .WORD 0
002060 003402'      ; PTR. TO DEVICE TYPE LIST
002062              L$REPP:: .WORD L$DVTYP
002062 022434'      ;PTR. TO REPORT CODE
002064              L$EXP4:: .WORD L$RPT
002064 000000      .WORD 0
002066              L$EXP5:: .WORD 0
002066 000000      .WORD 0
002070              L$AUT:: .WORD 0
002070 022122'      ;PTR. TO ADD UNIT CODE
002072              L$DUT:: .WORD L$AU
002072 022220'      ;PTR. TO DROP UNIT CODE
002074              L$LUN:: .WORD L$DU
002074 000000      ;LUN FOR EXERCISERS TO FILL
002076              L$DESP:: .WORD 0
002076 003410'      ;PTR. TO DIAG. DESCRIPTION
002100              L$LOAD:: .WORD L$DESC
002100 104035      ;GENERATE SPECIAL AUTOLOAD EMT
002102              EMT E$LOAD
002102 000000      L$ETP:: .WORD 0
002102              ;PTR. TO ERR TBL
002104              L$ICP:: .WORD 0
002104 021326'      ;PTR. TO INIT CODE
002106              L$CCP:: .WORD L$INIT
002106 022406'      ;PTR. TO CLEAN-UP CODE
002110              L$ACP:: .WORD L$CLEAN
002110 022326'      ;PTR. TO AUTO CODE
002112              L$PRT:: .WORD L$AUTO
002112 021316'      ;PTR. TO PROTECT TABLE
002114              L$TEST:: .WORD L$PROT
002114 000000      ;TEST NUMBER
002116              L$DLY:: .WORD 0
002116 000000      ;DELAY COUNT
002120              L$HIME:: .WORD 0
002120 000000      ;PTR. TO HIGH MEM
002120 000000      .WORD 0
31              .SBTTL DISPATCH TABLE
32
33
34
35              ;**
36              ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
37              ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
38              ;**

```

TSV2 - PROGRAM HEADER MACRO M1113 01-FEB-84 17:02

SEQ 020

```

DISPATCH TABLE
39 002122          DISPATCH 12
    002122 000014      .WORD 12
    002124          L$DISPATCH::
    002124 023216'      .WORD T1
    002126 024206'      .WORD T2
    002130 026200'      .WORD T3
    002132 031524'      .WORD T4
    002134 034314'      .WORD T5
    002136 040116'      .WORD T6
    002140 070230'      .WORD T7
    002142 051510'      .WORD T8
    002144 062336'      .WORD T9
    002146 066416'      .WORD T10
    002150 074260'      .WORD T11
    002152 077432'      .WORD T12
40
41          .SBTTL  DEFAULT HARDWARE P-TABLE
42
43          ;**
44          ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
45          ; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
46          ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
47          ;--
48 002154          BGNHW  DFPTBL          ;DEFAULT HARD-P-TABLE
    002154 000003      .WORD  L10000-L$HW/2
    002156          L$HW::
    002156          DFPTBL::
49
50 002156 172520      .WORD 172520          ; 1ST (OF 2) REGISTERS.
51 002160 000224      .WORD 224            ; INTERRUPT VECTOR
52 002162 000200      .WORD PRI04          ; INTERRUPT PRIORITY.
53 002164          ENDPHW
    002164          L10000:
54
55          .SBTTL  SOFTWARE P-TABLE
56
57          ;**
58          ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
59          ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
60          ;--
61 002164          BGNSW  SFPTBL
    002164 000004      .WORD  L10001-L$SW/2
    002166          L$SW::
    002166          SFPTBL::
62
63 002166 000000      TRANSTST::          .WORD 0          ; ENABLE TEST OF TRANSPORT(S) IF *1
64 002170 000000      NOITS::          .WORD 0            ; INHIBIT ITERATION OPTION.
65                                     ; ... 0 = ITERATE.
66                                     ; ... NZ = INHIBIT ITERATE.
67 002172 000017      LERRMAX::          .WORD 15.          ; LOCAL (PER TEST) ERROR LIMIT
68 002174 000310      GERRMAX::          .WORD 200.         ; GLOBAL (PER UNIT) ERROR LIMIT
69 002176          ENDSW
    002176          L10001:
70
71 002176          ENDMOD
72

```

TSV3 - GLOBAL AREAS
SOFTWARE P-TABLE

MACRO M1113 01-FEB-84 17:02

SEQ 021

```

7
8
13
19
20 002176
    002176
21
22
23
24
25
26
27
28
29
33 002176

        .TITLE  TSV3 - GLOBAL AREAS
        .SBTTL  GLOBAL EQUATES SECTION

        BGNMOD  TSV3
TSV3::

        .SBTTL  GLOBAL EQUATES SECTION

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

        EQUALS          ; GET STANDARD EQUATES.

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

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

        ; EVEN FLAG DEFINITIONS
        ; EF02:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
        ;
        000040          EF.START== 32.          ; START COMMAND WAS ISSUED
        000037          EF.RESTART== 31.        ; RESTART COMMAND WAS ISSUED
        000036          EF.CONTINUE== 30.       ; CONTINUE COMMAND WAS ISSUED
        000035          EF.NEW== 29.           ; A NEW PASS HAS BEEN STARTED
        000034          EF.PWR== 28.           ; A POWER-FAIL/POWER-UP OCCURRED
        ;
        ;

```

TSV3 - GLOBAL AREAS
GLOBAL EQUATES SECTION

MACRO M1113 01-FEB-84 17:02

J2

SEQ 022

```

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

```

```

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

```

54
35 002176

```

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

```

```
.ENDC
; *USER "I" PAGE ADDRESS REGISTER
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
; 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
```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
MEMORY MANAGEMENT DEFINITIONS

SEQ 024

```

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

; SBTTL TSU05 REGISTER AND PACKET DEFINITIONS
;
; SOME GENERAL EQUATES.
;
39
40
41
42
43
44
45
46
47 000004 ERRVEC= 4 ; POINTER TO ERROR VECTOR FOR BUS TIME OUT.
```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
TSU05 REGISTER AND PACKET DEFINITIONS

SEQ 025

```

48      000060      TTIVEC==      60      ; INTERRUPT VECTOR FOR CONSOLE INPUT
49      177560      TTICSR==      177560    ; BUS ADDRESS OF CONSOLE INPUT
50      177562      TTIBFR==      177562    ; CONSOLE INPUT DATA BUFFER
51      177520      BDVPCR==      177520    ; BDV2.1 PAGE CONTROL REGISTER
52
53      ;+
54      ;BIT DEFINITIONS FOR TSSR REGISTER
55      ;-
56
57      100000      SC=      BIT15      ;SPECIAL CONDITION
58      040000      BIE=      BIT14      ;BUS INTERFACE ERROR
59      020000      SCE=      BIT13      ;SANITY CHECK ERROR
60      010000      RMR=      BIT12      ;MODIFICATION REFUSED
61      004000      NXM=      BIT11      ;NONEXISTANT MEMORY ERROR
62      002000      NBA=      BIT10      ;NEED BUFFER ADDRESS
63      001400      HIADDR= BIT9:BIT8    ;EXTENDED ADDRESS BITS
64      000200      SSR=      BIT7      ;SUB SYSTEM READY
65      000100      OFL=      BIT6      ;OFF LINE BIT
66      000060      FATERR= BIT4:BIT5    ;FATAL TERMINATION ERROR CODES
67      000016      TERCLS= BIT3:BIT2:BIT1 ;TERMINATION CODES
68
69      ;+
70      ;
71      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
72      ;(XST0)
73      ;
74      ;-
75
76      100000      XSOTMK= BIT15      ;TAPE MARK DETECTED
77      040000      XSORLS= BIT14      ;RECORD LENGTH SHORT
78      020000      XSOLET= BIT13      ;LOGICAL END OF TAPE
79      010000      XSORLL= BIT12      ;RECORD LENGTH LONG
80      004000      XSOWLE= BIT11      ;WRITE LOCK ERROR
81      002000      XSONEF= BIT10      ;NON EXECUTABLE FUNCTION
82      001000      XSOILC= BIT9      ;ILLEGAL COMMAND
83      000400      XSOILA= BIT8      ;ILLEGAL ADDRESS
84      000200      XSOMOT= BIT7      ;TAPE IN MOTION
85      000100      XSOONL= BIT6      ;TRANSPORT ON LINE
86      000040      XSOIE= BIT5      ;INTERRUPT ENABLE
87      000020      XSOVCK= BIT4      ;VOLUME CHECK BIT
88      000010      XSOPED= BIT3      ;PHASE ENCODED DRIVE
89      000004      XSOWLK= BIT2      ;WRITE LOCKED
90      000002      XSOTOT= BIT1      ;BEGINNING OF TAPE
91      000001      XSOTET= BIT0      ;END OF TAPE
92
93      ;+
94      ;BIT DEFINITION, FOR EXTENDED STATUS REGISTER 1
95      ;(XST1)
96      ;-
97
98      100000      X1.DLT = BIT15      ;DATA LATE
99      040000      X1.SPARE= BIT14      ;NOT USED
100     020000      X1.COR = BIT13      ;CORRECTABLE DATA ERROR
101     017375      X1.MBZ = BIT12:BIT11:BIT10:BIT9:BIT7:BIT6:BIT5:BIT4:BIT3:BIT2:BIT0 ;ALWAYS 0
102     000400      X1.RBP = BIT8      ;READ BUS PARITY ERROR
103     000002      X1.UNC = BIT1      ;UNCORRECTABLE DATA OR HARD ERROR
104

```

N2

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
TSU05 REGISTER AND PACKET DEFINITIONS

SEQ 026

```

105
106
107      ;+
108      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
109      ;(XST2)
110      ;-
110      100000      X2.OPM = BIT15      ;OPERATION IN PROGRESS (TAPE MOVING)
111      040000      X2.RCE = BIT14      ;RAM CHECKSUM ERROR
112      035400      X2.SPARE= BIT13+BIT12+BIT11+BIT9+BIT8      ;NOT USED BY TSU05 (ALWAYS=0)
113      002000      X2.WCF = BIT10      ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
114      000200      X2.EXTF = BIT7      ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
115      000100      X2.BUFE = BIT6      ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
116      000077      X2.REV = 000077      ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
117      000007      X2.UNIT = BIT2+BIT1+BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.
118
119      ;+
120      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
121      ;(XST3)
122      ;-
123      177400      X3.MDE = 177400      ;MICRO-DIAGNOSTIC ERROR CODE
124      000200      X3.SPARE= BIT7      ;NOT USED BY TSU05
125      000100      X3.OPI = BIT6      ;OPERATION INCOMPLETE
126      000040      X3.REV = BIT5      ;REVERSE
127      000020      X3.TRF = BIT4      ;TRANSPORT RESPONSE FAILURE
128      000010      X3.DCK = BIT3      ;DENSITY CHECK
129      000006      X3.MBZ =BIT2+BIT1      ;NOT USED ALWAYS 0
130      000001      X3.RIB = BIT0      ;REVERSE INTO BOT
131
132      ;+
133      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
134      ;(XST4)
135      ;-
136      100000      X4.HSP = BIT15      ;HIGH SPEED
137      040000      X4.RCE = BIT14      ;RETRY COUNT EXCEEDED
138      020000      X4.TSM = BIT13      ;TRANSPORT SPECIAL MODE
139      017400      X4.MBZ = BIT12+BIT11+BIT10+BIT9+BIT8      ;NOT USED ALWAYS 0
140      000377      X4.WRC = 000377      ;WRITE RETRY COUNT FIELD
141
142
143      ;+
144      ;
145      ;TSSR TERMINATION CODES (BIT 0-2)
146      ;
147      ;-
148
149      000006      TSREJ= 3*2      ;COMMAND REJECTED
150      000006      UNREC= 6      ;UNRECOVERABLE ERROR
151
152      ;+
153      ;
154      ;DEVICE REGISTER OFFSETS
155      ;
156      ;-
157
158      000000      TSBA== 0
159      000000      TSDB== 0      ;TSDB/TSBA REGISTER
160      000001      TSBAH== 1
161      000001      TSDBH== 1      ;TSDB/TSBA REGISTER HIGH BYTE

```

TCV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
TS005 REGISTER AND PACKET DEFINITIONS

SEQ 027

```

162      000002      TSSR** 2      ;TSSR REGISTER
163      000003      TSSRH** 3      ;TSSR REGISTER HIGH BYTE
164
165      ;
166      ; TSOB ADDRESS BIT DEFINITIONS
167      ;
168      000003      A1716 = BIT1:BIT0      ;ADDRESS BITS 17:16 ARE IN 1:0
169
170      ;
171      ; COMMAND DEFINITIONS
172      ;
173      000017      P.GETSTAT = 17      ;GET STATUS
174      000013      P.INIT = 13      ;INITIALIZE
175      000012      P.CONTROL = 12      ;CONTROL COMMANDS
176      000011      P.FORMAT = 11      ;FORMAT
177      000010      P.POSITION = 10      ;POSITION
178      000006      P.WRTSUB = 6      ;SUBSYSTEM WRITE
179      000005      P.WRITE = 5      ;WRITE
180      000004      P.WRTCHAR = 4      ;WRITE CHARACTERISTICS
181      000001      P.READ = 1      ;READ
182
183      ;
184      ; COMMAND PACKET HEADER WORD BIT DEFINITIONS
185      ;
186      100000      P.ACK = BIT15      ;BUFFER AVAIL FOR CONTROLLER
187      040000      P.CVC = BIT14      ;CLEAR VOLUME CHECK
188      020000      P.OPP = BIT13      ;REVERSE SEQUENCE OF DATA BITS
189      010000      P.SWB = BIT12      ;SWAP BYTES IN MEMORY
190      007400      P.MODE = BIT11:BIT10:BIT9:BIT8      ;EXTENDED COMMAND MODE FIELD
191      000200      P.IE = BIT7      ;INTERRUPT ENABLE
192      000140      P.FMT = BIT6:BIT5      ;PACKET HEADER TYPE (ALWAYS=0)
193      000037      P.CMD = 37      ;MAJOR COMMAND FIELD
194
195      ;
196      ; CONTROL COMMAND MODE CODES
197      ;
198      000000      PC.RELEASE = 0*256.      ;RELEASE BUFFER
199      000400      PC.REWIND = 1*256.      ;REWIND
200      001000      PC.NOOP = 2*256.      ;NO-OP
201      002000      PC.IEREW = 4*256.      ;REWIND IMMEDIATE INTERRUPT
202      002400      PC.ERASE = 5*256.      ;SECURITY ERASE
203
204      ;
205      ; CONTROLLER RAM DEFINITIONS
206      ;
207      000167      RMCHBEG = 167      ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
208      000200      RMCHEND = 200      ;CHARACTERISTICS IO DATA END RAM ADDRESS
209      000201      RMPKTBEG = 201      ;COMMAND PACKET BEGIN RAM ADDRESS
210      000210      RMPKTEND = 210      ;COMMAND PACKET END RAM ADDRESS
211      000215      RMMSGBEG = 215      ;MESSAGE BUFFER BEGIN RAM ADDRESS
212      000234      RMMSGEND = 234      ;MESSAGE BUFFER END RAM ADDRESS
213
214      ;
215      ; REGISTER DEFINITIONS IN THE MESSAGE BUFFER
216      ;
217
218      000006      XSIQ** 6      ;EXTENDED STATUS REGISTER 0 (WORD 4)

```

C3

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
TSU05 REGISTER AND PACKET DEFINITIONS

SEQ 028

219	000010	XST1== 8.	;EXTENDED STATUS REGISTER 1 (WORD 5)
220	000012	XST2== 10.	;EXTENDED STATUS REGISTER 2 (WORD 6)
221	000014	XST3== 12.	;EXTENDED STATUS REGISTER 3 (WORD 7)
222	000016	XST4== 14.	;EXTENDED STATUS REGISTER 4 (WORD 8)
223			
224			
225			
226			
227			
228			
229			
230			
231	000002	PKLOW = 2	;LOW ORDER CHARACTERISTIC DATA POINTER
232	000004	PKHI = 4	;HIGH ORDER CHARACTERISTIC DATA POINTER
233	000006	PKBCNT = 6	;NUMBER OF BYTES IN DATA PACKET
234			
235	000010	EXBCNT=10	;NUMBER OF BYTES IN EXTENDED DATA PACKET
236			
237			
238			
239			
240	000000	BSELO = 0	;BYTE 0
241	000001	BSEL1 = 1	;BYTE 1
242	000002	SEL2 = 2	;WORD 2
243	000004	SELDATA = 4	;WORD 3
244			
245			
246			
247			
248	000000	PW.NOP = 0	;NO-OP
249	000001	PW.RDRAM = 1	;READ RAM
250	000002	PW.WTRAM = 2	;WRITE RAM
251	000003	PW.RFIFO = 3	;READ FIFO
252	000004	PW.WFIFO = 4	;WRITE FIFO
253	000005	PW.RDSTAT = 5	;READ STATUS
254	000006	PW.WCTL = 6	;WRITE TAPE CONTROL
255	000007	PW.WFMT = 7	;WRITE TAPE FORMAT
256	000010	PW.WMISC = 10	;WRITE MISCELLANEOUS
257	000011	PW.WNPR = 11	;WRITE NPR CONTROL
258	000020	PW.D22 = 20	;DO MICROTEST 22
259	000021	PW.D11 = 21	;DO MICROTEST 11
260	000022	PW.D13 = 22	;DO MICROTEST 13
261	000023	PW.NO1311 = 23	;DISABLE MICROTEST 11 AND 13
262	000024	PW.RDXT = 24	;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS)
263			
264			
265			
266			
267	000200	WC.IFAD = BIT7	;IFAD - FORMATTER ADDRESS
268	000100	WC.IOTAD = BIT6	;ITADO - TRANSPORT ADDRESS BIT 0
269	000040	WC.IITAD = BIT5	;ITAO1 - TRANSPORT ADDRESS BIT 1
270	000020	WC.ISRESV = BIT4	;IRESV5 - RESERVED #5
271	000010	WC.IREW = BIT3	;IREW - REWIND
272	000004	WC.IRWU = BIT2	;IRWU - REWIND AND UNLOAD
273	000002	WC.IFEN = BIT1	;IFEN - FORMATTER ENABLE
274	000001	WC.IGQ = BIT0	;IGQ
275			

```

276
277      ; BSEL1 CODES FOR WRITE FORMAT
278      ;
279      000200      WF.IHISP      = BIT7      ; IHISP - HIGH SPEED
280      000100      WF.IWRT      = BIT6      ; IWRT  - WRITE
281      000040      WF.IREV      = BIT5      ; IREV  - REVERSE
282      000020      WF.IWFM      = BIT4      ; IWFM  - WRITE FILE MARK
283      000010      WF.IEDIT     = BIT3      ; IEDIT - EDIT
284      000004      WF.IERASE    = BIT2      ; IERASE - ERASE
285      000002      WF.IRESV3    = BIT1      ; IRESV3 - RESERVED #3
286      000001      WF.IRESV4    = BIT0      ; IRESV4 - RESERVED #4
287
288
289      ;
290      ; BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
291      ;
292      000200      MS.EXT        = BIT7      ; INVERT SENSE OF EXTENDED FEATURES SWITCH
293      000020      MS.RSFIFO     = BIT4      ; RESET FIFO AND INPUT PARITY ERRORR
294      000010      MS.RSTAPE     = BIT3      ; RESET TAPE STATUS IN 2 FLIP-FLOPS
295      000006      MS.ATTN       = BIT2:BIT1 ; ATTENTION TRIGGER FIELD
296      000001      MS.RSD        = BIT0      ; RESET TIMER A,B THEN DELAY TIMES IN SEL2
297
298      ; MS.ATTN SUBCODES
299      ;
300      000000      MSA.NOP       = 0*2      ; NO-OP (NOTHING TRIGGERED)
301      000002      MSA.VOL       = 1*2      ; SIMULATE ON-LINE/OFF-LINE TRANSITION
302      000004      MSA.NRAM      = 2*2      ; FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
303      000006      MSA.FRAME     = 3*2      ; FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
304
305      ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
306      ;
307      000200      NP.IR         = BIT7      ; INTERRUPT REQUEST (0-1 TRANSITION)
308      000100      NP.OUT        = BIT6      ; TAPE DATA DIRECTION OUT (0= IN)
309      000040      NP.LOOP       = BIT5      ; ENABLE TRANSPORT LOOPBACK
310      000020      NP.WRP        = BIT4      ; WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
311
312      ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
313      ;
314
315      000200      S2.DIM         = BIT7      ; WORD #9 BYTE 2 DATA IN MISS
316      000100      S2.ILW        = BIT6      ; ILW H
317      000040      S2.OUTRDY      = BIT5      ; OUT RDY H
318      000020      S2.INRDY       = BIT4      ; IN RDY H
319      000010      S2.ATIMR       = BIT3      ; TIMER A FLAG H
320      000004      S2.BTIMR       = BIT2      ; TIMER B FLAG H
321      000003      S2.UNDEF       = BIT1:BIT0 ; (UNDEFINED)
322      100000      S1.PARIN       = BIT15     ; WORD #8 BYTE 1 PARIN H
323      040000      S1.I2RESV      = BIT14     ; IRESV2
324      020000      S1.I1RESV      = BIT13     ; IRESV1
325      010000      S1.IEOT        = BIT12     ; IEOT L
326      004000      S1.IIDENT      = BIT11     ; IIDENT H
327      002000      S1.ICER        = BIT10     ; ICER H
328      001000      S1.IFMK        = BIT9      ; IFMK H
329      000400      S1.IHER        = BIT8      ; IHER H
330      000200      S0.ISPEED      = BIT7      ; WORD #8 BYTE 0 ISPEED H
331      000100      S0.IRDY        = BIT6      ; IRDY L
332      000040      S0.IONL        = BIT5      ; IONL L

```

E3

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 TSU05 REGISTER AND PACKET DEFINITIONS

SEQ 030

```

333      000020      SO.ILDP      = BIT4      ;          ILDP L
334      000010      SO.IDBY      = BIT3      ;          IDBY L
335      000004      SO.IRWD      = BIT2      ;          IRWD L
336      000002      SO.IFBY      = BIT1      ;          IFBY L
337      000001      SO.IFPT      = BIT0      ;          IFPT L
338
339      ;*
340      ;UNIBUS MAP DEFINATIONS
341      ;
342      MMRO= 170200
343
344      .SBTTL  SPECIAL MACROS AND OPDEFS.
345
346
347      ;*
348      ;SAVE GENERAL REGS 1 TO 5
349      ;
350
351      .MACRO  SAVREG
352      JSR     R5,REGSAV
353      .ENDM
354
355      ;*
356      ; MACRO TO FORCE AN ERROR
357      ;
358      .MACRO  FORCERROR      TAG,NOTSSR
359      .NLIST
360      .IIF NDF LISTALL, .NLIST
361      .LIST
362      .IF B NOTSSR
363      MOV     TSSR(R5),R1      ;READ TSSR
364      .ENDC
365      MOV     FORCER,FORCER    ;IS FORCER SET? (LEAVE C BIT ALONE)
366      BNE     TAG              ;BR IF YES
367      .NLIST
368      .IIF NDF LISTALL, .LIST
369      .LIST
370      .ENDM
371
372      ;*
373      ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
374      ; WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
375      ; SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
376      ; FORCER TO 177777
377      ; TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
378      ;
379      .MACRO  FORCEEXIT      TAG
380      .NLIST
381      .IIF NDF LISTALL, .NLIST
382      .LIST
383      MOV     FORCER,FORCER    ;IS FORCER NEGATIVE?
384      BMI     TAG              ;BR IF YES
385      .NLIST
386      .IIF NDF LISTALL, .LIST
387      .LIST
388      .ENDM
389      ;*

```

```

390      ; MACRO TO INCREMENT ERROR COUNTS
391      ;-
392      .MACRO NEXT.ERRNO
393      .NLIST
394      ;;;.IIF NDF LISTALL, .NLIST
395      ERRNO=ERRNO+1
396      ;;;.IIF NDF LISTALL, .LIST
397      .LIST
398      .ENDM
399
400      ;+
401      ;MACRO TO PERFORM XOR
402      ;-
403
404      .MACRO XOR      A,B
405      MOV      A,-(SP)
406      BIC      B,(SP)
407      BIC      A,B
408      BIS      (SP)+,B
409      .ENDM
410
411      000000      EN=0      ; INITIALIZE ERROR NUMBER
412      .SBTTL  FORCER  - FORCE ERROR FLAG
413
414      ;
415      ; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
416      ; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
417      ;
418
419      002176  000000  FORCER::      0      ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED -
420      ; - BY THE MACRO "IFERROR"). AN ERROR NEED NOT -
421      ; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.
422
423
424
425      .SBTTL  GLOBAL DATA SECTION
426
427      ;++
428      ;THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
429      ;IN MORE THAN ONE TEST.
430      ;--
431
432      ;
433      ;THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
434      ;SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.
435      ;
436      002200  000000  EPRTSW::      .WORD      0      ;PRINT SWITCH
437      002202  000000  UNITN::      .WORD      0      ;UNIT # UNDER TEST.
438      002204  000000  QVP::      .WORD      0      ;QUICK VERIFY FLAG.
439      002206  000000  CSRADDR::      .WORD      0      ;ADDRESS OF CSR FOR CURRENT DEVICE
440      002210  000224  IVEC::      .WORD      224      ;INTERRUPT VECTOR
441      002212  000200  IPRI::      .WORD      PRI04      ;INTERRUPT PRIORITY.
442      002214  000000  TSTCNT::      .WORD      0      ;NUMBER OF TESTS RUN IN THIS PASS
443      002216  000000  LOOPCNT::      .WORD      0      ;REMAINING ITERATION COUNT FOR TEST
444      002220  000000  DEVCNT::      .WORD      0      ;NUMBER OF DEVICE UNDER TEST
445      002222  000000  FATFLG::      .WORD      0      ;SET IF FATAL ERROR IS DETECTED IN TEST
446      002224  000000  INTRECV::      .WORD      0      ;SET IF TAPE INTERRUPT WAS RECEIVED

```

TSV3 - GLOBAL AREAS
GLOBAL DATA SECTION

MACRO M1113 01-FEB-84 17:02

SEQ 032

447	002226	000000	EXTFEA::	.WORD	0	;EXTENDED FEATURES SOFTWARE SW 0-OFF;1-ON
448	002230	000000	BENBSW::	.WORD	0	;BUFFER ENABLE SWITCH SW 0-OFF;1-ON
449	002232	000000	EXPD::	.WORD	0	;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
450	002234	000000	RECV::	.WORD	0	;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
451	002236	000000	ERRHI::	.WORD	0	;HIGH ADDRESS MEMORY ERROR
452	002240	000000	ERRLO::	.WORD	0	;LOW ADDRESS MEMORY ERROR
453	002242		RAMDATA::	.BLKW	16.	;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
454	002302	000000	RAMSIZ::	.WORD	0	;RAM DATA SIZE FOR PRAMPKT ROUTINE
455	002304	000000	RCVHIADD::	.WORD	0	;RECEIVED BUFFER HIGH ADDRESS
456	002306	000000	RCVLOADD::	.WORD	0	;RECEIVED BUFFER LOW ADDRESS
457	002310	000000	COUNT::	.WORD	0	;TEST COUNT PATTERN
458	002312	000000	DATA::	.WORD	0	;TEST DATA
459	002314	000000	TSTFLAG::	.WORD	0	;TEST FLAG WORD
460	002316	000000	TSTPTR::	.WORD	0	;TSTBLK POINTER
461	002320	000000	PRMNO::	.WORD	0	;PRINT ROUTINE TEMP
462	002322		EXPMMSG::	.BLKB	100.	;EXPECTED MESSAGE BUFFER DATA
463	002466		RECMMSG::	.BLKB	100.	;RECEIVED MESSAGE BUFFER DATA
464	002632		TMPBFR::	.BLKB	80.	;TEMPORARY STORAGE FOR PRINT

.SBTTL TSTBLK - TEST DATA TABLE

```

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

```

483	002752		TSTBLK::			
484	002752	000000		.WORD	0	;ALL ZEROS
485	002754	177777		.WORD	177777	;ALL ONES
486	002756	000001		.WORD	BIT0	;DATA FOR WALKING ONES
487	002760	000002		.WORD	BIT1	
488	002762	000004		.WORD	BIT2	
489	002764	000010		.WORD	BIT3	
490	002766	000020		.WORD	BIT4	
491	002770	000040		.WORD	BIT5	
492	002772	000100		.WORD	BIT6	
493	002774	000200		.WORD	BIT7	
494	002776	000400		.WORD	BIT8	
495	003000	001000		.WORD	BIT9	
496	003002	002000		.WORD	BIT10	
497	003004	004000		.WORD	BIT11	
498	003006	010000		.WORD	BIT12	
499	003010	020000		.WORD	BIT13	
500	003012	040000		.WORD	BIT14	
501	003014	100000		.WORD	BIT15	
502	003016	177776		.WORD	*CBIT0	;DATA FOR WALKING ZEROS
503	003020	177775		.WORD	*CBIT1	

H3

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
TSTBLK - TEST DATA TABLE

SEQ 033

```

504 003022 177773      .WORD      +CBIT2
505 003024 177767      .WORD      +CBIT3
506 003026 177757      .WORD      +CBIT4
507 003030 177737      .WORD      +CBIT5
508 003032 177677      .WORD      +CBIT6
509 003034 177577      .WORD      +CBIT7
510 003036 177377      .WORD      +CBIT8
511 003040 176777      .WORD      +CBIT9
512 003042 175777      .WORD      +CBIT10
513 003044 173777      .WORD      +CBIT11
514 003046 167777      .WORD      +CBIT12
515 003050 157777      .WORD      +CBIT13
516 003052 137777      .WORD      +CBIT14
517 003054 077777      .WORD      +CBIT15
518 003056 125252      .WORD      125252      ;ALTERNATING ONES, ZEROS
519 003060 052525      .WORD      052525      ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
520                                TBLEND==.
521
522
523      .SBTTL  GLOBAL ENVIRONMENT STORAGE
524      ;
525      ;STORAGE FOR DEVICE REGISTERS
526      ;
527 003062 000000 100000 000000 DUMMY: 0,100000,0,0      ;DUMMY DEVICE REGISTERS...
528 003072 000000 000000 000000      0,0,0,0,0,0,0,0
529                                ;...FOR MULTI-UNIT CHECKOUT.
530
531
532 003112 000000      DUFLG::      .WORD      0      ;"DROPPED UNIT" FLAG.
533                                ;INHIBITS CODE IN "CLEAN-UP".
534 003114 000000      NODEV::      .WORD      0      ;FLAG TO SAY NO DEVICE.
535
536 003116 000000      TEMP1::      .WORD      0      ;SOME TEMP LOCATIONS.
537 003120 000000      TEMP2::      .WORD      0
538 003122 000000      XXCOMM::      .WORD      0      ;XXDP+ COMM BLOCK POINTER.
539 003124 000000      FREE::      .WORD      0      ;1ST FREE MEMORY ADDRESS...
540 003126 000000      FRESIZ::      .WORD      0      ;...AND SIZE (IN WORDS).
541 003130 000000      FREEHI: .WORD      0      ;LAST WORD IN FREE SPACE
542 003132 000000      KTFLG::      .WORD      0      ;KT11, MEM AVAIL FLAG -
543                                ;- .WORD      0 = <24K OR NO KT -
544                                ;- NZ = >24K AND KT.
545 003134 000000      KTENABLE::      .WORD      0      ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
546 003136 000000      NXMFLG::      .WORD      0      ;SET IF WE CAN TEST CLEARED OTHERWISE
547 003140 000000      NXML0::      .WORD      0      ;NXM LO ADDRESS BITS
548 003142 000000      NXMHI::      .WORD      0      ;NXM HI ADDRESS BITS FOR DAL'S 16-21
549 003144 000000      T23A::      .WORD      0      ;PROCESSOR TYPE FLAG
550 003146 000000      T23B::      .WORD      0      ;PROCESSOR TYPE FLAG B
551 003150 000000      T3BFLG::      .WORD      0      ;TEST 3B FLAG +0
552 003152 002000      PST32W::      .WORD      2000      ;32W BLOCK ADDRESS FOR 32K START
553 003154 000000      SIFLAG::      .WORD      0
554 003156 000000      BADDAT::      .WORD      0      ;ACTUAL DATA
555 003160 000000      GODAT::      .WORD      0      ;EXPECTED DATA
556 003162 000000      LOOPFL::      .WORD      0
557 003164      CTAB::      .WORD      0      ;CONFIGURATION TABLES.
558 003164 000000      CTABM::      .WORD      0      ;CONFIG WORK.
559 003166 000000      .WORD      0
560 003170 000000      .WORD      0

```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
GLOBAL ENVIRONMENT STORAGE

SEQ 034

```

561 003172 000000 .WORD 0
562 003174 177777 .WORD -1 ;END OF MEM TABLE.
563 003176
564 CTABE::
565 ;ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
566 ;
567 ; 0 = UNIT NOT TESTED
568 ; 100000 = UNIT ONLINE, NO ERRORS
569 ; 10XXXX = UNIT ONLINE, ENCOUNTERED XXXX ERRORS
570 ; 160000 = UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
571 ; 160001 = UNIT DROPPED, NOT IDLE AT START
572 ; 14XXXX = UNIT DROPPED, ENCOUNTERED XXXX ERRORS
573 003176
574 003376 000000
575
576 003400 000000
577
578 SKIPT: .WORD 0 ;1=SKIP SUBTEST 0=NO SKIP OF SUBTEST
579
580 ;SBTTL GLOBAL TEXT MESSAGES
581 ;++
582 ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
583 ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
584 ; MORE THAN ONE TEST.
585 ;--
586
587 ;+
588 ;NAMES OF DEVICES SUPPORTED
589 ;-
590
591 003402 DEVTYP <TSU05>
592 003402
593 003402 124 123 125 L$DVTYP::
594 ;.ASCIZ /TSU05/
595 ;.EVEN
596
597
598
599
600 ;+
601 ;TEST DESCRIPTION
602 ;-
603 003410
604 003410
605 003410 052 052 052 L$DESC::
606 ;.ASCIZ /**** TSU05 DIAG PART 2 - REPLACE M7455 IF ERROR ****/
607 ;.EVEN
608
609
610
611
612 ;+
613 ;BIT TO ASCII CONVERSION FOR TSSR REGISTER
614 ;-
615
616 003476 003536 003541 003545 TSSRBIT:: .WORD 1$,2$,3$,4$,5$,6$,7$,8$
617 003516 003577 003603 003607 .WORD 9$,10$,11$,12$,13$,14$,15$,16$
618 003536 123 103 000 1$: .ASCIZ 'SC'
619 003541 102 111 105 2$: .ASCIZ 'BIE'
620 003545 123 103 105 3$: .ASCIZ 'SCE'
621 003551 122 115 122 4$: .ASCIZ 'RMR'
622 003555 116 130 115 5$: .ASCIZ 'NXM'

```

TSV3 - GLOBAL AREAS
GLOBAL TEXT MESSAGES

MACRO M1113 01-FEB-84 17:02

SEQ 035

```

632 003561      116      102      101  64:      .ASCIZ  'NBA'
633 003565      102      111      124  74:      .ASCIZ  'BIT9'
634 003572      102      111      124  84:      .ASCIZ  'BIT8'
635 003577      123      123      122  94:      .ASCIZ  'SSR'
636 003603      117      106      114  104:     .ASCIZ  'OFL'
637 003607      102      111      124  114:     .ASCIZ  'BIT5'
638 003614      102      111      124  124:     .ASCIZ  'BIT4'
639 003621      102      111      124  134:     .ASCIZ  'BIT3'
640 003626      102      111      124  144:     .ASCIZ  'BIT2'
641 003633      102      111      124  154:     .ASCIZ  'BIT1'
642 003640      102      111      124  164:     .ASCIZ  'BIT0'
643              .EVEN
644 003646      124      123      123  SFIERR: .ASCIZ  'TSSR ERROR AFTER SOFT INIT'
645 003701      124      123      123  SFHERR: .ASCIZ  'TSSR ERROR AFTER BUS RESET'
646 003734      040      040      116  NXR:    .ASCIZ  / NON-EXISTANT DEVICE REGISTER/
647 003773      045      101      040  NXR:    .ASCIZ  /#A ADDRESS: #06/
648 004014      045      101      040  TSSX:   .ASCII  /#A TSBA,TSSR EXP'D: #06#A,#06#N/
649 004054      045      101      040  TSSX:   .ASCIZ  /#A TSBA,TSSR REC'D: #06#A,#06/
650 004113      045      116      045  FUSI:   .ASCII  /#N#A/
651 004117      040      040      125  USI:    .ASCIZ  / UNEXPECTED INTERRUPT/
652 004146      040      040      111  NSI:    .ASCIZ  / INTERRUPT EXPECTED, NOT RECEIVED/
653 004211      045      116      045  FNOINTR: .ASCII  /#N#A/
654 004215      040      040      116  NOINTR: .ASCIZ  / NO INTERRUPT WAS GENERATED/
655 004252      040      040      111  IFAULT: .ASCIZ  / INTERRUPT FAULT/
656 004274      045      101      040  INTX:   .ASCIZ  /#A CPU PC: #06#A TSBA: #06/
657 004331      040      040      042  NOINIT: .ASCIZ  / "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
658 004403      040      040      042  NSINIT: .ASCIZ  / "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
659 004453      040      040      042  BRINIT: .ASCIZ  / "BUS-RESET" DIDN'T INITIALIZE THE DPU/
660
661 004523      000              NUL:    .ASCIZ  //
662 004524      045      116      000  NULCR: .ASCIZ  /#N/
663 004527      045      101      040  EXPGOT: .ASCIZ  /#A EXP'D: #06#A, REC'D: #06/
664 004563      045      116      045  EXPGT2: .ASCIZ  /#N#A EXP'D: #06#A, #06#N#A REC'D: #0#A, #06/
665 004637      045      101      040  DUAD12: .ASCIZ  /#A REG(W) WRITTEN TO: #06#A REG(R) READ: EXP'D: #06#A, REC'D: #06/
666 004741      122      101      115  PKTRAM: .ASCIZ  'RAM Contents Do Not Match Packet Sent'
667 005007      040      040      103  SCME:   .ASCIZ  / CONFIG DOESN'T MATCH MFG. MASTER/
668 005052      127      122      111  WRTMSG: .ASCIZ  'WRITE CHARACTERISTICS Failed'
669 005107      124      123      123  WRTERR: .ASCIZ  'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
670 005202      124      123      123  RDERR:  .ASCIZ  'TSSR Incorrect After READ Command, More Bits Set Than SSR'
671 005274      106      101      124  SCHERR: .ASCIZ  'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
672 005366      105      122      122  RETERR: .ASCIZ  'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
673 005454      045      116      045  NOMEM:  .ASCIZ  '#N#A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****N'
674              .EVEN
675
676              .SBTTL  GLOBAL ERROR REPORT SECTION
677
678
679      ;**
680      ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
681      ; CALLS THAT ARE USED IN MORE THAN ONE TEST.
682      ; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
683      ;**
684 005550      BGNMSG  NXRERR              ;NON-EXISTANT DEVICE REGISTER.
685 005550      NXRERR: PRINTX  #NXRX,NODEV      ;NODEV = NEXM ADDRESS.
686 005550      MOV      NODEV,-(SP)
687 005554      MOV      #NXRX,-(SP)

```

K3

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
GLOBAL ERROR REPORT SECTION

SEQ 036

005560	012746	000002	MOV	#2,-(SP)	
005564	010600		MOV	SP,R0	
005566	104415		TRAP	C#PNTX	
005570	062706	000006	ADD	#6,SP	
686 005574	004737	005602'	JSR	PC,EXTEND	; PRINT EXTENSION IF REQUIRED.
687 005600			ENDMSG		
005600			L10002:		
005600	104423		TRAP	C#MSG	
688					
689					
690					
691					
692					
693					
694 005602	005727		EXTEND: TST	(PC)+	
695 005604	000000		EXTA:	0	; 0 = NO EXTENSION.
696 005606	001402		BEQ	1\$	
697 005610	004777	177770	JSR	PC,EXTA	; APPEND EXTENSION TEXT.
698 005614			PRINTX	#NULCR	; PRINT A BLANK LINE
005614	012746	004524'	MOV	#NULCR,-(SP)	
005620	012746	000001	MOV	#2,-(SP)	
005624	010600		MOV	SP,R0	
005626	104415		TRAP	C#PNTX	
005630	062706	000004	ADD	#4,SP	
699 005634	000207		RTS	PC	
700					
701					
702					
703					
704					
705					
706					
707					
708					
709					
710					
711					
712					
713					
714					
715					
716					
717					
718					
719 005636					
720 005636					
721 005642	010104				
722 005644					
005644	010446				
005646	012746	006227'			
005652	012746	000002			
005656	010600				
005660	104414				
005662	062706	000006			
723 005666	010400				
724 005670	004737	015654'			
725 005674	103410				


```

; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
; TO ANY OF THE ABOVE ERROR SIGNATURES.
;
EXTEND: TST      (PC)+
EXTA:  0                ; 0 = NO EXTENSION.
      BEQ      1$
      JSR      PC,EXTA    ; APPEND EXTENSION TEXT.
1$:     PRINTX  #NULCR    ; PRINT A BLANK LINE
      MOV      #NULCR,-(SP)
      MOV      #2,-(SP)
      MOV      SP,R0
      TRAP     C#PNTX
      ADD      #4,SP
      RTS      PC

.SBTTL  PRITSSR - PRINT TSSR CONTENTS

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

PRITSSR:
      SAVREG                ; SAVE GENERAL REGISTERS
      MOV      R1,R4        ; SAVE THE TSSR CONTENTS
      PRINTB   #TSSRFOR,R4  ; PRINT THE CONTENTS OF TSSR
      MOV      R4,-(SP)
      MOV      #TSSRFOR,-(SP)
      MOV      #2,-(SP)
      MOV      SP,R0
      TRAP     C#PNTB
      ADC      #6,SP
      MOV      R4,R0        ; GET TSSR BACK FOR CHKAMB
      JSR      PC,CHKAMB    ; ARE CONTENTS AMBIGUOUS ?
      BCS      5$          ; BRANCH IF NOT

```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 PRITSSR - PRINT TSSR CONTENTS

SEQ 037

```

726 005676          PRINTX  #AMBTSSR          ;SHOW CONTENTS ARE AMBIGUOUS
      005676 012746 006447'      MOV          #AMBTSSR,-(SP)
      005702 012746 000001      MOV          #1,-(SP)
      005706 010600          MOV          SP,R0
      005710 104415          TRAP         C#PNTX
      005712 062706 000004      ADD          #4,SP
727 005716 010403          5$:  MOV          R4,R3          ;CONTENTS OF TSSR
728 005720 042703 001476      BIC          #HIADDR!FATERR!TERCLS,R3 ;CLEAR ALL MULTIPLE BIT FIELDS
729 005721 001434          BEQ          20$          ;NO BITS ARE SET
730 005726 012702 002632'      MOV          #TMPBFR,R2          ;TEMPORARY ASCII BUFFER
731 005732 012701 003476'      MOV          #TSSRBIT,R1          ;ASCII EQUIVALENT OF BITS
732 005736 005703          10$:  TST          R3          ;REMAINING BITS TO CONVERT
733 005740 001413          BEQ          15$          ;BRANCH WHEN ALL ARE DONE
734 005742 000241          CLC          ;CLEAR CARRY FOR SHIFT
735 005744 006103          ROL          R3          ;SHIFT NEXT BIT TO CARRY
736 005746 103006          BCC          13$          ;BRANCH IF BIT NOT SET
737 005750 011100          MOV          (R1),R0          ;POINTER TO BIT DEFINITION
738 005752 112022          11$:  MOVB       (R0)+,(R2)+      ;MOVE ASCII TO BUFFER
739 005754 001376          BNE          11$          ;MOVE ALL BITS
740 005756 112762 000054 177777  MOVB       #'',(R2)      ;INSERT A COMMA TO TERMINATE
741 005764 005721          13$:  TST          (R1)+          ;POINT TO NEXT DESCRIPTION
742 005766 000763          BR          10$          ;GET THE REMAINING BITS
743 005770 105042          15$:  CLRB       -(R2)          ;TERMINATE THE LINE
744 005772          PRINTX  #TSSDEF,#TMPBFR      ;PRINT THE BIT DEFINITIONS
      005772 012746 002632'      MOV          #TMPBFR,-(SP)
      005776 012746 006420'      MOV          #TSSDEF,-(SP)
      006002 012746 000002      MOV          #2,-(SP)
      006006 010600          MOV          SP,R0
      006010 104415          TRAP         C#PNTX
      006012 062706 000006      ADD          #6,SP

745
746 006016 010403          20$:  MOV          R4,R3          ;GET THE TSSR CONTENTS
747 006020 042703 177761      BIC          #CTERCLS,R3          ;CLEAR ALL BUT TERMINATION
748 006024 016303 006510'      MOV          TCOCOD(R3),R3          ;GET THE TERMINATION CODE MEANING
749 006030          PRINTX  #TCOASC,R3          ;PRINT THE TERMINATION CODE
      006030 010346          MOV          R3,-(SP)
      006032 012746 006310'      MOV          #TCOASC,-(SP)
      006036 012746 000002      MOV          #2,-(SP)
      006042 010600          MOV          SP,R0
      006044 104415          TRAP         C#PNTX
      006046 062706 000006      ADD          #6,SP

750 006052 010403          MOV          R4,R3          ;TSSR CONTENTS AGAIN
751 006054 042703 177717      BIC          #CFATERR,R3          ;CLEAR ALL BUT FATAL TERMINATION
752 006060 001416          BEQ          25$          ;DON'T PRINT IF ZERO
753 006062 006203          ASR          R3
754 006064 006203          ASR          R3
755 006066 006203          ASR          R3          ;ALINE TERMINATION CODE FOR INDEX
756 006070 016303 007050'      MOV          TSFCOD(R3),R3          ;GET THE FATAL TERMINATION CODE
757 006074          PRINTX  #TFCASC,R3          ;PRINT THE FATAL TERMINATION CODE
      006074 010346          MOV          R3,-(SP)
      006076 012746 006351'      MOV          #TFCASC,-(SP)
      006102 012746 000002      MOV          #2,-(SP)
      006106 010600          MOV          SP,R0
      006110 104415          TRAP         C#PNTX
      006112 062706 000006      ADD          #6,SP
758 006116 042704 176377          25$:  BIC          #CHIADDR,R4          ;CLEAR ALL BUT EXTENDED ADDRESS
759 006122 001411          BEQ          30$          ;DON'T PRINT IF ZERO

```

M3

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
PRITSSR - PRINT TSSR CONTENTS

SEQ 038

```

750 006124          PRINIX  #TEXASC,R4          ;PRINT THE EXTENDED ADDRESS BITS
      006124 010446  MOV      R4,-(SP)
      006126 012746 006247' MOV      #TEXASC,-(SP)
      006132 012746 000002  MOV      #2,-(SP)
      006136 010600  MOV      SP,R0
      006140 104415  TRAP     C$PNTX
      006142 062706 000006  ADD      #6,SP
761 006146 013703 002200' 30$: MOV      EPRTSW,R3          ;PRINT MEASGE BUFFER ADDRESS
762 006152          PRINTX  R3          ;PRINT PROPER MESSAGE
      006152 010346  MOV      R3,-(SP)
      006154 012746 000001  MOV      #1,-(SP)
      006160 010600  MOV      SP,R0
      006162 104415  TRAP     C$PNTX
      006164 062706 000004  ADD      #4,SP
763 006170 000207  RTS      PC          ;RETURN TO CALLER
764
770 006172          EPRT2:
771 006172          045      116      045  EPRT1: .ASCIZ  '***REPLACE M7455***'
772
782 006227          045      116      045  TSSRFOR: .ASCIZ  '***TSSR = #06'
783 006247          045      116      045  TEXASC: .ASCIZ  '***Extended Address Bits = #06'
784 006310          045      116      045  TCOASC: .ASCIZ  '***Termination Class Code = #T'
785 006351          045      116      045  TFCASC: .ASCIZ  '***Fatal Termination Class Code = #T'
786 006420          045      116      045  TSSDEF: .ASCIZ  '***TSSR Bits Set: #T'
787 006447          045      116      045  AMBTSSR: .ASCIZ  '***TSSR Contents Are Ambiguous'
788
789 006510 006530' 006553' 006601' TCOCOD: .EVEN
790 006530          116      157      162  1$: .ASCIZ  'Normal Termination'
791 006553          124      145      162  2$: .ASCIZ  'Termination Condition'
792 006601          124      141      160  3$: .ASCIZ  'Tape Status Alert'
793 006623          106      165      156  4$: .ASCIZ  'Function Reject'
794 006643          122      145      143  5$: .ASCIZ  'Recoverable Error - Tape Position One Record Down'
795 006725          122      145      143  6$: .ASCIZ  'Recoverable Error - Tape Was Not Moved'
796 006774          125      156      162  7$: .ASCIZ  'Unrecoverable Error'
797 007020          106      141      164  8$: .ASCIZ  'Fatal Controller Error'
798
799
800 007050 007060' 007114' 007125' TSFCOD: .WORD  1$,2$,3$,4$
801 007060          111      156      164  1$: .ASCIZ  'Internal Diagnostic Failure'
802 007114          122      145      163  2$: .ASCIZ  'Reserved'
803 007125          102      165      163  3$: .ASCIZ  'Bus Interface or Sanity Check Error'
804 007171          122      145      163  4$: .ASCIZ  'Reserved'
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
      .SBTTL  PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

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

```

N3

TSV3 - GLOBAL AREA: MACRO M1113 01-FEB-84 17:02
 PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

SEQ 039

```

820      ;
821      ;
822      PRIPKT:
823      SAVREG      ;SAVE THE REGISTERS
824      MOV R0,R5    ;SAVE NO. OF WORDS IN PACKET
825      TST KENABLE ;ABOVE 28K UNDER TEST?
826      BNE 10$      ;BR IF YES
827      CLR R3       ;SET HIGH ORDER ADDRESS TO 0
828      MOV R3,R1    ;COPY HIGH ORDER ADDRESS
829      MOV R4,R0    ;GET LOWER ADDRESS
830      ROL R0       ;SHIFT BIT 15 INTO C BIT
831      ROL R1       ;AND INTO HIGH ORDER.
832      PRINTB @PKTADD,R1,R4 ;PRINT PACKET ADDRESS
      MOV R4,-(SP)
      MOV R1,-(SP)
      MOV @PKTADD,-(SP)
      MOV @3,-(SP)
      MOV SP,R0
      TRAP C$PNTB
      ADD @10,SP
833      MOV R3,R0    ;GET HIGH ORDER ADDRESS
834      BEQ 20$      ;BR IF NOT ABOVE 28K.
835      MOV R4,R1    ;GET LOW ORDER ADDRESS
836      JSR PC,SETMAP ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
837      MOV R0,R4    ;GET RETURNED PAR6 ADDRESS BIAS
838      CLR R1       ;SAVE WORD NUMBER
839      MOV (R4)+,R2 ;GET PACKET CONTENTS
840      PRINTB @PKTFRM,R1,R2 ;PRINT THE DATA
      MOV R2,-(SP)
      MOV R1,-(SP)
      MOV @PKTFRM,-(SP)
      MOV @3,-(SP)
      MOV SP,R0
      TRAP C$PNTB
      ADD @10,SP
841      INC R1       ;NEXT WORD NUMBER
842      CMP R1,R5    ;DONE ALL PACKET WORDS?
843      BLT 25$      ;LOOP TILL ALL DONE
844      RTS PC       ;RETURN
845
846      045 116 045 PKTFRM: .ASCIZ 'NNA Packet Word #D1NA = #05'
847      045 116 045 PKTADD: .ASCIZ 'NNA Packet Address = #01#05'
848      .EVEN
849
850
851      .SBTTL PRIBXOR - PRINT EXPD, RECV AND XOR BYTE
852
853      ;
854      ;
855      ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE.
856      ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
857      ;
858      ;INPUTS:
859      ;
860      R1      RECEIVED DATA
861      R2      EXPECTED DATA
862      ;

```

B4

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 PRIEXOR - PRINT EXPD, RECV AND XOR BYTE

SEQ 040

```

863      ;OUTPUT:
864      ;
865      ;      RO      XOR OF EXPECTED/RECEIVED DATA
866      ;
867      ;
868
869 007424      PRIEXOR::
870 007424      SAVREG      ;SAVE THE REGISTERS
871 007430 010203      MOV      R2,R3      ;EXPECTED DATA
872 007432      XOR      R1,R3      ;FORM THE EXCLUSIVE OR
873 007442 012700 177400      MOV      #C<377>,R0      ;BYTE MASK
874 007446 040001      BIC      R0,R1      ;SAVE LOW BYTE RECV
875 007450 040002      BIC      R0,R2      ;SAVE LOW BYTE EXPD
876 007452 040003      BIC      R0,R3      ;SAVE LOW BYTE XOR
877 007454      PRINTB      #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
      007454 010346      MOV      R3,-(SP)
      007456 010146      MOV      R1,-(SP)
      007460 010246      MOV      R2,-(SP)
      007462 012746 007506      MOV      #XORFOR,-(SP)
      007466 012746 000004      MOV      #4,-(SP)
      007472 010600      MOV      SP,R0
      007474 104414      TRAP      C1PNTB
      007476 062706 000012      ADD      #12,SP
878 007502 010300      MOV      R3,R0      ;RO HAS XOR ON RETURN
879 007504 000207      RTS      PC      ;RETURN TO CALLER
880
881 007506 045 116 045 XORBFOR: .ASCIZ 'NNA EXPD; #03NA RECV; #03NA XOR; #03'
882      .EVEN
883
884      .SBTTL PRIEXOR - PRINT EXPD, RECV AND XOR
885
886
887
888
889 ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
890 ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
891
892 ;INPUTS:
893 ;
894 ;      R1      RECEIVED DATA
895 ;      R2      EXPECTED DATA
896
897 ;OUTPUT:
898 ;
899 ;      RO      XOR OF EXPECTED/RECEIVED DATA
900 ;
901 ;
902
903 007554      PRIEXOR::
904 007554      SAVREG      ;SAVE THE REGISTERS
905 007560 010203      MOV      R2,R3      ;EXPECTED DATA
906 007562      XOR      R1,R3      ;FORM THE EXCLUSIVE OR
907 007572      PRINTB      #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
      007572 010346      MOV      R3,-(SP)
      007574 010146      MOV      R1,-(SP)
      007576 010246      MOV      R2,-(SP)
      007600 012746 007624      MOV      #XORFOR,-(SP)

```

C4

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
PRI XOR - PRINT EXPD, RECV AND XOR

SEQ 041

```

007604 012746 000004      MOV    #4,-(SP)
007610 010600      MOV    SP,R0
007612 104414      TRAP   C#PNTB
007614 062706 000012      ADD    #12,SP
908 007620 010300      MOV    R3,R0      ;R0 HAS XOR ON RETURN
909 007622 000207      RTS     PC      ;RETURN TO CALLER
910
911 007624      045      116      045 XORFOR: .ASCIZ 'NNNA EXPD; #06NA RECV; #06NA XOR; #06'
912                      .EVEN
913
914                      .SBTTL PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT
915
916
917
918      ;
919      ;ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
920      ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
921      ;
922      ;INPUTS:
923      ;
924      ;      R0      OCTAL VALUE TO CONVERT
925      ;      R1      TABLE OF POINTERS TO ASCII EQUIVALENT
926      ;
927      ;
928 007672      PRIEQU:
929 007672      SAVREG
930 007676 000207      RTS     PC      ;SAVE THE REGISTERS
931                      ;RETURN TO CALLER
932
933
934
935                      .SBTTL PRIRAM - PRINT RAM ADDRESS
936
937      ;
938      ;PRINT CONTROLLER RAM ADDRESS.
939      ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
940      ;
941      ;INPUTS:
942      ;
943      ;      R4      RAM ADDRESS
944      ;
945      ;
946 007700      PRIRAM:
947 007700      SAVREG
948 007704      PRINTB  #RAMFOR,R4      ;SAVE R1-R5 UNTIL NEXT RETURN
949                      ;PRINT RAM ADDRESS IN ERROR
950                      MOV    R4,-(SP)
951                      MOV    #RAMFOR,-(SP)
952                      MOV    #2,-(SP)
953                      MOV    SP,R0
954                      TRAP   C#PNTB
955                      ADD    #6,SP
956                      RTS     PC      ;RETURN
957
958 007730      045      116      045 RAMFOR: .ASCIZ 'NNNA CONTROLLER RAM ADDRESS = #06'
959                      .EVEN
960
961
962

```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 PRIADD - PRINT MEMORY ERROR ADDRESS

SEQ 042

```

955                                     .SBTTL PRIADD - PRINT MEMORY ERROR ADDRESS
956                                     **
957                                     ;
958                                     ;PRINT MEMORY ADDRESS
959                                     ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
960                                     ;
961                                     ; IMPLICIT INPUTS
962                                     ;
963                                     ;     ERRHI - HIGH ORDER ADDRESS
964                                     ;     ERRLO - LOW ORDER ADDRESS
965                                     ;
966                                     ;
967 007772 PRIADD:
968 007772 SAVREG                                     ;SAVE R1-P5 UNTIL NEXT RETURN
969 007776 013700 002236' MOV ERRHI,R0               ;GET HIGH ADDRESS
970 010002 013701 002240' MOV ERRLO,R1             ;GET LOW ADDRESS
971 010006 010102 MOV R1,R2                         ;COPY LOW ADDRESS
972 010010 006101 ROL R1                             ;SHIFT BIT 15 TO C BIT
973 010012 006100 ROL R0                             ;SHIFT INTO HIGH ORDER
974 010014 PRINTB #PRIA0,R0,R2                       ;PRINT MEMORY ADDRESS IN ERROR
          010014 010246 MOV R2,-(SP)
          010016 010046 MOV R0,-(SP)
          010020 012746 010042' MOV #PRIA0,-(SP)
          010024 012746 000003 MOV #3,-(SP)
          010030 010600 MOV SP,R0
          010032 010414 TRAP C:PNTB
          010034 010706 000010 ADD #10,SP
975 010040 000207 RTS PC                             ;RETURN
976
977 010042 045 116 045 PRIA0: .ASCIZ 'NWA MEMORY ERROR ADDRESS - #U1#05'
978 .EVEN
979
980                                     .SBTTL PRITADD - PRINT MEMORY TEST ADDRESS
981                                     **
982                                     ;
983                                     ;PRINT MEMORY ADDRESS
984                                     ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
985                                     ;
986                                     ; IMPLICIT INPUTS
987                                     ;
988                                     ;     ERRHI - HIGH ORDER ADDRESS
989                                     ;     ERRLO - LOW ORDER ADDRESS
990                                     ;
991                                     ;
992                                     ;
993 010106 PRITADD:
994 010106 SAVREG                                     ;SAVE R1-R5 UNTIL NEXT RETURN
995 010110 013702 002236' MOV ERRHI,R2               ;GET HIGH ADDRESS
996 010116 013701 002240' MOV ERRLO,R1             ;GET LOW ADDRESS
997 MOV R1,R2                         ;COPY LOW ADDRESS
998 ROL R1                             ;SHIFT BIT 15 TO C BIT
999 ROL R0                             ;SHIFT INTO HIGH ORDER
1000 PRINTB #PRITO,R1                       ;PRINT MEMORY ADDRESS LOW IN ERROR
          010122 010146 MOV R1,-(SP)
          010124 012746 010170' MOV #PRITO,-(SP)
          010130 012746 000002 MOV #2,-(SP)
          010134 010600 MOV SP,R0

```

E4

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
PRITADD - PRINT MEMORY TEST ADDRESS

SEQ 043

1001	010136	104414				TRAP	CIPNTB	
	010140	062706	000006			ADD	#6,SP	
	010144					PRINTB	#PRIT1,R2	;PRINT MEMORY ADDRESS HIGH IN ERROR
	010146	010246				MOV	R2,-(SP)	
	010146	012746	010233'			MOV	#PRIT1,-(SP)	
	010152	012746	000002			MOV	#2,-(SP)	
	010156	010600				MOV	SP,R0	
	010160	104414				TRAP	CIPNTB	
	010162	062706	000006			ADD	#6,SP	
1002	010168	000207				RTS	PC	;RETURN
1003								
1004	010170	045	116	045	PRIT0:	.ASCIZ	'NNA MEMORY TEST ADDRESS LOW = #06'	
1005	010233	045	116	045	PRIT1:	.ASCIZ	'NNA MEMORY TEST ADDRESS HIGH = #06'	
1006						.EVEN		
1007								
1008								
1009						.SBTTL	SPACE	- SPACE RECORDS (FORWARD AND REVERSE) COMMAND
1010								
1011								
1012								
1013								
1014								
1015								
1016								
1017								
1018								
1019								
1020								
1021								
1022								
1023								
1024								
1025								
1026								
1027								
1028								
1029								
1030								
1031								
1032								
1033								
1034								
1035								
1036								
1037								
1038								
1039								
1040								
1041								
1042								
1043								
1044	010300							
1045	010300							
1046	010304	012737	000764	010470'		SAVREG		;SAVE THE GENERAL REGISTERS
1047	010312	012737	140010	010460'		MOV	#500,,SDELAY	;SET UP DELAY
1048	010320	005703				MOV	#140010,803	;SET UP COMMAND, SPACE FORWARD
1049	010322	100403				IST	R3	;CHECK FOR DIRECTION
						BMI	53	;BR, IF REVERSE INDICATED

F4

TSV3 - GLOBAL AREAS
SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

SEQ 044

```

1050 010324 010337 010462'      MOV      R3,90$      ;LOAD UP NUMBER OF RECORDS TO SPACE
1051 010330 000407              BR      10$      ;GO DO COMMAND
1052 010332 042703 100000      5$:      BIC      @BIT15,R3      ;CLEAR DIRECTION BIT
1053 010336 010337 010462'      MOV      R3,90$      ;LOAD UP NUMBER OF RECORDS TO SPACE
1054 010342 052737 000400 010460'  BIS      @BIT8,80$      ;SET REVERSE BIT IN COMMAND PACKET
1055 010350 012704 010460'      10$:     MOV      @80$,R4      ;SET UP R4 WITH PACKET ADDRESS
1056 010354 010465 000000      MOV      R4,TSDB(R5)      ;SEND OUT COMMAND
1057 010360 004737 016060'      15$:     JSR      PC,WAITF      ;WAIT FOR SSR
1058 010364 103420              BCS      20$      ;BR, IF SSR IS SET AND OK
1059 010366              DELAY      250      ;DELAY ABOUT .25 SECONDS
      010366 012727 000250      MOV      @250,(PC)+
      010372 000000              .WORD      0
      010374 013727 002116'      MOV      L$DLY,(PC)+
      010400 000000              .WORD      0
      010402 005367 177772      DEC      -6(PC)
      010406 001375              BNE      -4
      010410 005367 177756      DEC      -22(PC)
      010414 001367              BNE      -20
1060 010416 005337 010470'      DEC      SDELAY      ;BUMP DELAY COUNTER DOWN
1061 010422 001356              BNE      15$      ;BR, IF MORE DELAY
1062 010424 000411              BR      60$      ;BR IF TROUBLE CARRY = CLEAR
1063 010426 016501 000002      20$:     MOV      TSSR(R5),R1      ;READ TSSR
1064 010432 012702 000200      MOV      @SSR,R2      ;SET UP EXPECTED
1065 010436 020201      25$:     CMP      R2,R1      ;ARE THEY OK
1066 010440 001401              BEQ      40$      ;BR, IF EQUAL = OK
1067 010442 000402              BR      60$      ;TROUBLE EXIT
1068 010444 000261      40$:     SEC              ;SET CARRY NO TROUBLE
1069 010446 000401              BR      70$      ;EXIT
1070 010450 000241      60$:     CLC              ;CARRY CLEAR = ERROR
1071 010452      70$:              ;
1072 010452 010400      MOV      R4,R0      ;PASS PACKET ADDRESS
1073 010454 000207      RTS      PC      ;RETURN
1074
1075
1076
1077
1078      ;PACKET FOR SPACE COMMAND
1079
1080 010456              .BLKB      10-<.-TSV2&7>
1081
1082      ;
1083      ;COMMAND WORD
1084      80$:      .WORD
1085 010460 000000      ;NUMBER OF RECORDS TO BE SPACED OVER WORD
1086      90$:      .WORD
1087 010462 000000      .WORD
1088 010464 000000      .WORD
1089 010466 000000      .WORD
1090 010470 000000      SDELAY: .WORD      0      ;DELAY COUNTER
1091              .EVEN
1092
1093
1094      .SBTTL  WRCHR - WRITE CHARACTERISTICS COMMAND
1095
1096
1097
1098      ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
1099      ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
1100

```

G4

TSV3 - GLOBAL AREA MACRO M1113 01-FEB-84 17:02
WRTCHR - WRITE CHARACTERISTICS COMMAND

SEQ 045

```

1101      ;INPUT:
1102      ;
1103      R4      ADDRESS OF PACKET FROM TEST
1104      R5      FIRST DEVICE UNIBUS ADDRESS
1105      ;      REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
1106      ;
1107      ;OUTPUT:
1108      ;
1109      R0      TSSR CONTENTS
1110      CARRY   SET - WRITE CHARACTERISTICS COMMAND OK
1111      ;      CLR - WRITE CHARACTERISTICS FAILED
1112      ;
1113      ;IMPLICIT OUTPUT:
1114      ;
1115      ;      MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
1116      ;      SOFTWARE SWITCHES SET AS FOLLOWS:
1117      ;      EXTFEA = EXTENDED FEATURES PRESENT
1118      ;      BENBSW = BUFFER ENABLE SWITCH ON OR OFF
1119      ;
1120      ;
1121      ;SIDE EFFECTS:
1122      ;
1123      ;
1124      ;
1125      ;
1126      WRTCHR:
1127      ;      SAVREG      ;SAVE THE GENERAL REGISTERS
1128      CLR      BENBSW    ;CLEAR BUFFER ENABLE SWITCH
1129      CLR      EXTFEA    ;CLEAR EXTENDED FEATURES SW SWITCH
1130      10$: MOV      R4,TSDB(R5) ;SEND OUT COMMAND
1131      JSR      PC,CHKTSSR ;WAIT FOR SSR
1132      BCS      20$      ;BR, IF SSR IS SET AND OK
1133      BR       60$      ;BR IF TROUBLE CARRY = CLEAR
1134      20$: MOV      TSSR(R5),R1 ;READ TSSR
1135      MOV      %SSR,R2   ;SET UP EXPECTED
1136      BIT      %OFL,R1   ;WAS OFF LINE SET IN TSSR
1137      BEQ      25$      ;BR, IF NO OFL SET
1138      BIS      %OFL,R2   ;MAKE THEM LOOK ALIKE
1139      25$: CMP      R2,R1 ;ARE THEY OK
1140      BEQ      40$      ;BR, IF EQUAL = OK
1141      BR       60$      ;TROUBLE EXIT
1142      40$: ADD      #8,R4  ;POINT TO WRT CHARA DATA PACKET
1143      MOV      (R4),R3   ;GET ADDRESS OF MESSAGE BUFFER
1144      RIT      %X2.EXTF,XST2(R3) ;EXTENDED FEATURES BIT SET?
1145      BEQ      45$      ;BR IF NO
1146      INC      EXTFEA    ;SET EXTENDED FEATURES SW SWITCH
1147      45$:
1148      BIT      %X2.BUFE,XST2(R3) ;BUFFER ENABLE SWITCH SET
1149      BEQ      50$      ;BR, IF SWITCH NOT SET
1150      INC      BENBSW    ;SET SOFTWARE SWITCH FOR ENABLED
1151      50$:
1152      SEC      ;SET CARRY NO TROUBLE
1153      BR       70$      ;EXIT
1154      60$: CLC      ;CARRY CLEAR = ERROR
1155      70$: MOV      TSSR(R5),R0 ;RETURN TSSR CONTENTS
1156      RTS      PC      ;RETURN
1157
1126 010472
1127 010472
1128 010476 005037 002230'
1129 010502 005037 002226'
1130 010506 010465 000000
1131 010512 004737 016146'
1132 010516 003401
1133 010520 000435
1134 010522 016501 000002
1135 010526 012702 000200
1136 010532 032701 000100
1137 010536 001402
1138 010540 052702 000100
1139 010544 020201
1140 010546 001401
1141 010550 000421
1142 010552 062704 000010
1143 010556 011403
1144 010560 032763 000200 000012
1145 010566 001402
1146 010570 005237 002226'
1147 010574
1148 010576 032763 000100 000012
1149 010602 001402
1150 010606 005037 002230'
1151 010610
1152 010616 000261
1153 010622 000401
1154 010628 000241
1155 010634 016503 000002
1156 010640 000207
1157

```

H4

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
WRITCHR - WRITE CHARACTERISTICS COMMAND

SEQ 046

```

1158
1159          .SBTTL  REWIND - POSITION TAPE (REWIND) COMMAND
1160
1161          ;*
1162          ; THIS ROUTINE WILL REWIND THE SELECTED TAPE.
1163          ;
1164          ; CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
1165          ; TO ARRIVE, ALSO THE CALLER MUST CHECK FOR
1166          ; SSR TO SET IN THE TSSR
1167          ;
1168          ;
1169          ; CALLING SEQUENCE:
1170          ;
1171          ; DO A SOFT INIT
1172          ; DO A WRITE CHARACTERISTICS
1173          ; JSR      PC,REWIND
1174          ;
1175          ; INPUT:
1176          ;
1177          ; R5      FIRST DEVICE UNIBUS ADDRESS
1178          ;
1179          ;
1180          ; OUTPUT
1181          ;
1182          ; R0      THE CONTENTS OF R4 IS PASSED TO R0
1183          ;
1184          ;
1185          ;
1186          ; -
1187          REWIND::
1188          SAVREG
1189          MOV     @RWPACK,R4
1190          MOV     R4,TSDB(R5)
1191          MOV     @360.,R3
1192          JSR     PC,WAITF
1193          BCS     20$
1194          DELAY   250.
1195          MOV     @250.,(PC)+
1196          .WORD   0
1197          MOV     L$DLY,(PC)+
1198          .WORD   0
1199          DEC     -6(PC)
1200          BNE     -4
1201          DEC     -22(PC)
1202          BNE     -20
1203          DEC     R3
1204          BNE     10$
1205          CLC
1206          MOV     R4,R0
1207          RTS     PC
1208
1209          ; SAVE R1-R5 UNTIL NEXT RETURN
1210          ; GET PACKET ADDRESS
1211          ; SEND PACKET ADDRESS TO EXECUTE
1212          ; ENOUGH TIME FOR 2400' REEL TO REWIND
1213          ; WAIT FOR SSR TO SET
1214          ; LEAVE WHEN SSR IS SET
1215          ; WAIT FOR .25 SECONDS
1216
1217          ; BUMP COUNTER DOWN
1218          ; KEEP GOING
1219          ; CLEAR CARRY TO SET ERROR
1220          ; PASS THE PACKET ADDRESS
1221          ; RETURN
1222
1223          RWPACK: .BLKB  10-<.-TSV2&7>
1224
1225          .WORD   102010
1226          .WORD   0
1227
1228          ; POSITION COMMAND (REWIND)
1229          ; NOT USED

```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 REWIND - POSITION TAPE (REWIND) COMMAND

SEQ 047

1209
 1210
 1211
 1212
 1213
 1214
 1215
 1216
 1217
 1218
 1219
 1220
 1221
 1222
 1223
 1224
 1225
 1226
 1227
 1228
 1229
 1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239 010724
 1240 010724
 1241 010730 012701 002242'
 1242 010733 012702 000201
 1243 010740 005003
 1244 010742 004737 016146'
 1245 010746 112765 000000 000000
 1246 010753 004737 016146'
 1247 010760 010265 000000
 1248 010764 004737 016146'
 1249 010770 116511 000000
 1250 010774 122124
 1251 010776 001401
 1252 011000 005203
 1253 011000 005202
 1254 011004 020227 000210
 1255 011010 003761
 1256 011012 005703
 1257 011013 001402
 1258 011016 000241
 1259 011020 000401
 1260 011022 000261
 1261 011024 012737 000010 002302'
 1262 011032 000207
 1263
 1264
 1265

.SBTTL CKRAM - COMPARE RAM TO I/O PACKET

```

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

```

CKRAM:.

```

; SAVREG
; ADDRESS TO SAVE THE RAM DATA
; BYTE ADDRESS OF FIRST RAM DATA
; CLEAR THE ERROR FLAG
; WAIT FOR SSR
; SET MAINTENANCE MODE
; WAIT FOR SSR TO SET
; SELECT NEXT RAM ADDRESS
; WAIT FOR SSR TO SET
; READ THE RAM DATA
; COMPARE TO EXPECTED
; BRANCH IF OK
; SET ERROR FLAG
; ADDRESS OF NEXT RAM LOCATION
; REACHED END YET ?
; BRANCH TILL ALL READ
; WAS AN ERROR FOUND ?
; BRANCH IF NOT
; CLEAR CARRY TO SHOW ERROR
; AND EXIT
; SHOW GOOD COMPARE
; SETUP RAMSIZ FOR PRAMPKT ROUTINE
; RETURN

```

.SBTTL CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA

SEQ 048

```

1266      ;*
1267      ;
1268      ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
1269      ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
1270      ;
1271      ;INPUT:
1272      ;
1273      ;      R4      ADDRESS OF THE CHARACTERISTICS DATA
1274      ;      R5      FIRST DEVICE UNIBUS ADDRESS
1275      ;
1276      ;OUTPUT:
1277      ;
1278      ;      CARRY   SET - RAM MATCHES PACKET
1279      ;             CLR - RAM DOES NOT MATCH PACKET
1280      ;
1281      ;IMPLICIT OUTPUT:
1282      ;
1283      ;      THE TABLE RAMDATA IS FILLED WITH THE
1284      ;      DATA HELD IN RAM.
1285      ;      RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
1286      ;
1287      ;SIDE EFFECTS:
1288      ;
1289      ;      THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1290      ;
1291      ;*
1292      ;
1293      CKRAM2::
1294      ;
1295      ;      SAVREG
1296      ;      MOV      #RAMDATA,R1      ;SAVE THE GENERAL REGISTERS
1297      ;      MOV      #RMCHBEG,R2      ;ADDRESS TO SAVE THE RAM DATA
1298      ;      CLR      R3               ;BYTE ADDRESS OF FIRST RAM DATA
1299      ;      JSR      FC,CHKTSSR       ;CLEAR THE ERROR FLAG
1300      ;      MOV      #0,TSDB(R5)      ;WAIT FOR SSR
1301      ;      JSR      PC,CHKTSSR       ;SET MAINTENANCE MODE
1302      ;      MOV      R2,TSDB(R5)      ;WAIT FOR SSR TO SET
1303      ;      JSR      PC,CHKTSSR       ;SELECT NEXT RAM ADDRESS
1304      ;      MOV      TSBA(R5),(R1)    ;WAIT FOR SSR TO SET
1305      ;      CMP      (R1)+,(R4)+      ;READ THE RAM DATA
1306      ;      BEQ      20$              ;COMPARE TO EXPECTED
1307      ;      INC      R3               ;BRANCH IF OK
1308      ;      INC      R2               ;SET ERROR FLAG
1309      ;      MOV      #8,,RAMSIZ       ;ADDRESS OF NEXT RAM LOCATION
1310      ;      TST      EXTFEA           ;ASSUME EXTFEA NOT SET
1311      ;      BEQ      25$              ;IS THE SOFTWARE EXTENDED FEATURES SET
1312      ;      MOV      #10,,RAMSIZ      ;BR, IF NOT SET
1313      ;      CMP      R2,#RMCHEND      ;SET RAMSIZ FOR EXTEND FEATURES
1314      ;      BLE      10$              ;AT END OF EXTENDED BUFFER
1315      ;      BR      27$              ;BR, IF NOT AT END YET
1316      ;      CMP      R2,#RMCHEND-2    ;AT END BRANCH
1317      ;      BLE      10$              ;REACHED END YET ?
1318      ;      TST      R3               ;BRANCH TILL ALL READ
1319      ;      BEQ      30$              ;WAS AN ERROR FOUND ?
1320      ;      CLC                       ;BRANCH IF NOT
1321      ;      BR      50$              ;CLEAR CARRY TO SHOW ERROR
1322      ;      SEC                       ;AND EXIT
1323      ;      RTS      PC               ;SHOW GOOD COMPARE
1324      ;
1325      ;      RETURN

```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA

SEQ 049

1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350 011170
1351 011170
1352 011174 010037 002304'
1353 011200 010137 002306'
1354 011204 005737 003134'
1355 011210 001403
1356 011212 004737 017130'
1357 011216 010001
1358 011220 005004
1359 011222 005003
1360 011224 010205
1361 011226 011264 002322'
1362 011232 011164 002466'
1363 011236 022221
1364 011240 001401
1365 011242 005203
1366 011244 062704 000002
1367 011250 020427 000014
1368 011254 003764
1369 011256 032765 000200 000012
1370 011264 001403
1371 011266 020427 000016
1372 011272 003755
1373 011274 005703
1374 011276 001402
1375 011300 000241
1376 011302 000401
1377 011304 000261
1378 011306 000207
1379

```
.SBTTL CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS
;
; ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
; BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
; ERROR PRINT ROUTINES.
; INPUT:
; R0 RECV MESSAGE BUFFER HIGH ORDER ADDRESS
; R1 RECV MESSAGE BUFFER LOW ORDER ADDRESS
; R2 EXPD MESSAGE BUFFER ADDRESS
; OUTPUT:
; CARRY SET - MESSAGE BUFFERS MATCH
; CLR -MESSAGE BUFFERS DON'T MATCH
; IMPLICIT OUTPUT:
; EXPMSG BUFFER IS SET TO EXPD DATA
; RECMMSG BUFFER IS SET TO RECV DATA
; RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
; RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
;
CKMSG::
    SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
    MOV R0,RCVHIADD                      ;SAVE RECV HIGH ADDRESS
    MOV R1,RCVLOADD                      ;SAVE RECV LOW ADDRESS
    TST KTENARLE                          ;TESTING ABOVE 28K?
    BEQ 10$                               ;BR IF NO
    JSR PC,SETMAP                        ;RETURN ADDRESS BIASED TO PAR6 IN R0
    MOV R0,R1                            ;GET RETURNED ADDRESS BIASED TO PAR6
    CLR R4                               ;WORD IN BUFFER
    CLR R3                               ;CLEAR ERROR SEEN FLAG
    MOV R2,R5                            ;GET EXPD BUFFER ADDRESS
    MOV (R2),EXPMSG(R4)                  ;SAVE EXPD FOR ERROR REPORT
    MOV (R1),RCMMSG(R4)                  ;SAVE RECV FOR ERROR REPORT
    CMP (R2)+,(R1)+                      ;EXPD EQUAL RECV?
    BEQ 25$                               ;BR IF YES
    INC R3                               ;SET ERROR SEEN FLAG
    ADD #2,R4                            ;POINT TO NEXT WORD ADDRESS
    CMP R4,#14                           ;DONE FIRST 7 WORDS?
    BLE 15$                               ;BR IF NO
    BIT #X2,EXTF,XST2(R5)                ;IS EXTENDED FEATURES SET IN EXPD?
    BEQ 50$                               ;BR IF NO
    CMP R4,#16                           ;DONE EXTENDED FEATURES WORD?
    BLE 15$                               ;BR IF NO
    YST R3                               ;ANY ERRORS SEEN?
    BEQ 55$                               ;BR IF NO
    CLC                                  ;SET FAILURE
    BR 60$                               ;
    SEC                                  ;SET SUCCESS
    RTS PC                               ;RETURN
```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS

SEQ 050

1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408 011310
1409 011310
1410 011314 020327 000144
1411 011320 003412
1412 011322 012703 000144
1413 011326
011326 012746 011442
011332 012746 000001
011336 010600
011340 104417
011342 062706 000004
1414 011346 010037 002304
1415 011352 010137 002306
1416 011356 005737 003134
1417 011362 001403
1418 011364 004737 017130
1419 011370 010001
1420 011372 005004
1421 011374 005005
1422 011376 111264 002322
1423 011402 111164 002466
1424 011406 122221
1425 011410 001401
1426 011412 005205
1427 011414 062704 000001
1428 011420 020403
1429 011422 002001
1430 011424 000764
1431 011426 005705

```

.SBTTL CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS
;
; ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
; BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
; ERROR PRINT ROUTINES.
;
; INPUT:
;
;      R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
;      R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
;      R2      EXPD MESSAGE BUFFER ADDRESS
;      R3      NUMBER OF BYTES TO COMPARE
;
; OUTPUT:
;
;      CARRY    SET - MESSAGE BUFFERS MATCH
;               CLR - MESSAGE BUFFERS DON'T MATCH
;
; IMPLICIT OUTPUT:
;
;      EXPMSG    BUFFER IS SET TO EXPD DATA
;      RECVMSG    BUFFER IS SET TO RECV DATA
;      RCVHIADD   SET TO HIGH ORDER ADDRESS OF RECV
;      RCVLOADD   SET TO LOW ORDER ADDRESS OF RECV
;
CKMSG2::
    SAVREG                ;SAVE R1-R5 UNTIL NEXT RETURN
    CMP      R3,00000000  ;R3,000 IS COUNT ABOVE MAX ALLOWED?
    BLE      5$           ;000 BR IF NO
    MOV      R3,00000000  ;R3,000
    PRINTF   00000000     ;000
    MOV      R4,00000000  ;000
    MOV      R4,00000000  ;000
    MOV      R4,00000000  ;000
    MOV      R4,00000000  ;000
    TRAP     C$PNTF
    ADD      R4,SP
    5$: MOV    R0,RCVHIADD  ;SAVE RECV HIGH ADDRESS
        MOV    R1,RCVLOAD  ;SAVE RECV LOW ADDRESS
        TST    R4          ;TESTING ABOVE 200?
        BEQ    10$         ;OR IF NO
        JSR    PC,SETMAP   ;RETURN ADDRESS BIASED TO PAR6 IN R0
        MOV    R0,R1       ;GET RETURNED ADDRESS BIASED TO PAR6
    10$: CLR    R4          ;WORD IN BUFFER
        CLR    R5          ;CLEAR ERROR SEEN FLAG
    15$: MOVB   (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
        MOVB   (R1),RECVMSG(R4) ;SAVE RECV FOR ERROR REPORT
        CMPB   (R2), (R1)    ;EXPD EQUAL RECV?
        BEQ    25$         ;BR IF YES
        INC    R5          ;SET ERROR SEEN FLAG
    25$: ADD    R1,R4       ;POINT TO NEXT BYTE
        CMP    R4,R3       ;DONE ALL BYTES?
        BGE    50$         ;BR IF YES
        BR     15$         ;DO NEXT BYTE
    50$: TST    R5          ;ANY ERRORS SEEN?

```

M4

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

SEQ 051

```

1432 011430 001402      BEQ      55$      ;BR IF NO
1433 011432 000241      CLC              ;SET FAILURE
1434 011434 000401      OR         60$      ;
1435 011436 000261      55$: SEC          ;SET SUCCESS
1436 011440 000207      60$: RTS         PC   ;RETURN
1437
1438 011442      120      122      117  DEBUGMSG: .ASCIZ  PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-' ;@@D
1439 011532      045      116      045  FERCM: .ASCII /NMA ***/
1440 011543      040      040      124  ERCM: .ASCIZ / TSSR ERROR CODE REC'D - /
1441 011576      056      056      056  SIMSG: .ASCIZ /... AFTER DOING SOFT INIT/
1442 011631      124      105      123  TINERR: .ASCIZ /TEST: .../
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460 011644      BGNMSG  SFIMSG
      011644      SFIMSG: JSR      PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1461 011644 004737 005636' JSR      PC,CKDROP      ;DROP UNIT, IF ALLOWED
1462 011650 004737 017014' ENDMMSG
1463 011654
      011654      L10003: TRAP      C$MSG
      011654 104423
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476 011656      BGNMSG  PKTSSR
      011656      PKTSSR: JSR      PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1477 011656 004737 005636' MOV      #4,R0      ;NO. OF WORDS IN PACKET
1478 011662 012700 000004' JSR      PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
1479 011666 004737 007202' ENDMMSG
1480 011672
      011672      L10004: TRAP      C$MSG
      011672 104423
1481
1482

```

N4

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

SEQ 052

```

1483 ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1484 ;TSSR AND A GET STATUS COMMAND PACKET.
1485 ;
1486 ;INPUTS:
1487 ;
1488 ; R1 TSSR CONTENTS
1489 ; R4 ADDRESS OF COMMAND PACKET
1490 ;
1491 ;
1492 ;
1493 011674 BGNMSG PKTGETS
011674 PKTGETS:
1494 011674 004737 005636' JSR PC,PRITSSR ;PRINT THE CONTENTS OF TSSR REGISTER
1495 011700 012700 000002' MOV #2,R0 ;NO. OF WORDS IN GET STATUS PACKET
1496 011704 004737 007202' JSR PC,PRIPKT ;PRINT THE CONTENTS OF COMMAND PACKET
1497 011710 ENDMSG
011710 L10005:
011710 104423 TRAP C$MSG

1498
1499
1500
1501 ;*
1502 ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
1503 ;
1504 ;INPUTS:
1505 ;
1506 ; R1 TSSR CONTENTS
1507 ; R4 ADDRESS OF COMMAND PACKET
1508 ;
1509 ;
1509 011712 BGNMSG SFFMSG
011712 SFFMSG:
1510 011712 004737 005636' JSR PC,PRITSSR ;PRINT CONTENTS OF TSSR REGISTER
1511 011716 ENDMSG
011716 L10006:
011716 104423 TRAP C$MSG

1512
1513
1514 .SBTTL PKTMES - PRINT TSSR AND MESSAGE BUFFER
1515 ;*
1516 ;
1517 ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
1518 ;BUFFER FOR ERROR REPORTS
1519 ;
1520 ;INPUTS:
1521 ;
1522 ; R1 CONTENTS OF TSSR
1523 ; R2 LOW ORDER MESSAGE BUFFER
1524 ; R3 HIGH ORDER MESSAGE BUFFER ADDRESS
1525 ; NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
1526 ;
1527 011720 BGNMSG PKTMES
011720 PKTMES:
1528 011720 004737 005636' JSR PC,PRITSSR ;PRINT CONTENTS OF TSSR
1529 011724 010200 MOV R2,R0 ;LOW ORDER ADDRESS
1530 011726 010301 MOV R3,R1 ;HIGH ORDER ADDRESS
1531 011730 004737 014052' JSR PC,PRMESS ;PRINT THE MESSAGE BUFFER
1532 011734 ENDMSG

```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
PKTMS - PRINT TSSR AND MESSAGE BUFFER

SEQ 053

```

011734      L10007: TRAP      C#MSG
011734      104423

1533
1534
1535          .SBTTL  ADDSSR - PRINT TEST ADDRESS AND TSSR
1536
1537          ;PRIN: ROUTINE TO PRINT THE CONTENTS OF
1538          ;TSSR AND A MEMORY TEST ADDRESS
1539
1540          ;INPUTS:
1541          ;
1542          ;      R5      FIRST DEVICE UNIBUS ADDRESS
1543          ;      ERAMI   HIGH ORDER MEMORY TEST ADDRESS
1544          ;      ERRLO   LOW ORDER MEMORY TEST ADDRESS
1545          ;
1546
1547      011736      BGNMSG  ADDSSR
1548      011736      ADDSSR: JSR      PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
1549      011742      004737 010106'  MOV      TSSR(R5),R1      ;GET CURRENT TSSR
1550      011746      004737 005636'  JSR      PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1551      011752      ENDMMSG
1552      011752      L10010: TRAP      C#MSG
1553      011752      104423

1552
1553
1554          .SBTTL  MSGEXP - PRINT WRITE CHAR. EXPD-RCV MESSAGE BUFFERS
1555
1556          ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
1557
1558          ;IMPLICIT INPUTS:
1559          ;
1560          ;      EXPMSG - EXPECTED MESSAGE BUFFER
1561          ;      RECMMSG - RECEIVED MESSAGE BUFFER
1562          ;      RCVHIADD - RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1563          ;      RCVLOADD - RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1564          ;
1565
1566      011754      BGNMSG  MSGEXP
1567      011754      MSGEXP: MOV      #7,R0      ;ASSUME NO EXT FEATURES
1568      011760      005737 002226'  TST      EXTFEA      ;EXT FEATURES SET?
1569      011764      001402      BEQ      5$      ;BR IF NO
1570      011766      012700 000010  MOV      #8,R0      ;EXT FEATURE BUFFER IS 8 WORDS
1571      011772      004737 014362'  JSR      PC,PRMSGEXP      ;PRINT EXPD/RCV MESSAGE BUFFERS
1572      011776      ENDMMSG
1573      011776      L10011: TRAP      C#MSG
1574      011776      104423

1573
1574
1575          .SBTTL  FIFEXP - PRINT FIFO EXP/RCV DATA
1576
1577          ;PRINT ROUTINE TO PRINT FIFO EXP/RCV DATA
1578
1579          ;      R1      BYTE COUNT
1580
1581          ;IMPLICIT INPUTS:

```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 FIFEXP - PRINT FIFO EXP/RECV DATA

SEQ 054

```

1582
1583
1584
1585
1586 012000
      012000
1587 012000
      012000 010146
      012002 012746 012052'
      012006 012746 000002
      012012 010600
      012014 104415
      012016 062706 000006
1588 012022
      012022 012746 012121'
      012026 012746 000001
      012032 010600
      012034 104415
      012036 062706 000004
1589 012042 010100
1590 012044 004737 014732'
1591 012050
      012050
      012050 104423
1592 012052 045 116 045 FIF1MSG:
1593 012121 045 116 045 FIF2MSG:
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609 012160
      012160
1610 012160 012701 012222'
1611 012164 012100
1612 012166 001410
1613 012170
      012170 010046
      012172 012746 000001
      012176 010600
      012200 104415
      012202 062706 000004
1614 012206 000766
1615 012210 012700 000012
1616 012214 004737 014362'
1617 012220
      012220

```

```

EXPMSG - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
RECMSG - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
BGNMSG FIFEXP:
PRINTX #FIF1MSG,R1 ;PRINT BYTES TRANSFERRED
MOV R1,-(SP)
MOV #FIF1MSG,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #6,SP
PRINTX #FIF2MSG ;PRINT HEADER MSG
MOV #FIF2MSG,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #4,SP
MOV R1,R0 ;GET BYTE COUNT
JSR PC,PRBYTEXP ;PRINT FIFO BYTES IN ERROR
ENDMSG

L10012:
TRAP C#MSG
.ASCIZ 'NWA NUMBER OF BYTES TRANSFERRED - #02'
.ASCIZ 'NWA FIFO DATA BYTES IN ERROR:'
.EVEN

.SBTTL MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
;
;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
;
;IMPLICIT INPUTS:
;
EXPMSG - EXPECTED MESSAGE BUFFER
RECMSG - RECEIVED MESSAGE BUFFER
RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
;
BGNMSG MSGSTAT
MSGSTAT:
MOV #STATCOD,R1 ;ASCII ADDRESS TABLE
104: MOV (R1),R0 ;DONE ALL MSG LINES?
BEQ 204 ;BR IF YES
PRINTX R0 ;PRINT STATUS BIT NAMES
MOV R0,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #4,SP
BR 104 ;DO ANOTHER MSG LINE
204: MOV #10,R0 ;NUMBER OF WORDS IN A READ STATUS BUFFER
JSR PC,PRMSGEXP ;PRINT EXPD/RECV MESSAGE BUFFERS
ENDMSG

L10013:

```

D5

TSV3 GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS

SEQ 055

```

1618      012220 104423      TRAP      CMSG
1619 012222 012240' 012302' 012373' STATCOD:      .WORD 1#,2#,3#,4#,5#,6#,0
1620 012240      045      116      045 1#:.ASCIZ 'NNA Tape Bus Signals in Word #8:'
1621 012302      045      116      045 2#:.ASCIZ 'NNA      PARERR<15> IEOT <12> IFMK <9> IRDY<6> IRWD<2>'
1622 012373      045      116      045 3#:.ASCIZ 'NNA      IRESV2<14> IIDENT<11> IHER <8> IONL<5> IFBY<1>'
1623 012464      045      116      045 4#:.ASCIZ 'NNA      IRESV1<13> ICER <10> ISPEED<7> ILDP<4> IFPT<0>'
1624 012555      045      116      045 5#:.ASCIZ 'NNA Tape Bus Signals in Word #9:'
1625 012617      045      116      045 6#:.ASCIZ 'NNA      DATMIS<7> ILW<6> OUTRDY<5> INRDY<4>'
1626      .EVEN
1627
1628
1629
1630      .SBTTL MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
1631
1632      ;
1633      ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
1634      ;
1635      ;IMPLICIT INPUTS:
1636      ;
1637      ;      EXPMSG - EXPECTED MESSAGE BUFFER
1638      ;      RECHMSG - RECEIVED MESSAGE BUFFER
1639      ;      RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1640      ;      RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1641      ;
1642 012674      BGNMSG MSGLOOP
1643 012674      MSGLOOP::
1644 012700      MOV      #LOOPCOD,R1      ;ASCII ADDRESS TABLE
1645 012702      MOV      (R1),R0      ;DONE ALL MSG LINES?
1646 012704      BEQ      20#      ;BR IF YES
1647      PRINTX      R0      ;PRINT STATUS BIT NAMES
1648      MOV      R0,-(SP)
1649      MOV      #1,-(SP)
1650      MOV      SP,R0
1651      TRAP      C#PNTX
1652      ADD      #4,SP
1653      BR      10#
1654 012722      MOV      #10,R0      ;DO ANOTHER MSG LINE
1655 012724      JSR      PC,PRMSGEXP      ;NUMBER OF WORDS IN A READ STATUS BUFFER
1656 012730      ENDMSG      ;PRINT EXPD/RCV MESSAGE BUFFERS
1657 012734      L10014:
1658 012734      TRAP      CMSG
1659
1660 012736 012756' 013031' 013130' LOOPCOD:      .WORD 1#,2#,3#,4#,5#,6#,7#,0
1661 012756      045      116      045 1#:.ASCIZ 'NNA Tape Bus Loopback Signals in Word #8:'
1662 013031      045      116      045 2#:.ASCIZ 'NNA      PARERR<15> IRESV2<14> IRESV1<13>'
1663 013130      045      116      045 3#:.ASCIZ 'NNA      IHISP->IEOT<12> IWRT->IIDENT<11> IHEV ->ICER <10>'
1664 013227      045      116      045 4#:.ASCIZ 'NNA      IWM ->IFMK<09> IEDIT->IHER <08> IFAD ->ISPEED<07>'
1665 013326      045      116      045 5#:.ASCIZ 'NNA      ITADO->IRDY<06> ITAD1->IONL <05> IERASE->ILDP <04>'
1666 013425      045      116      045 6#:.ASCIZ 'NNA      IREW ->IOBY<03> IRWU ->IRWD <02> IFEN ->IFBY <01>'
1667 013524      045      116      045 7#:.ASCIZ 'NNA      IGO ->IFPT<00>'
1668      .EVEN
1669
1670      .SBTTL MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
1671
1672      ;
1673      ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV

```

E5

TSV3 - GLOBAL AREAS MACRC M1113 01-FEB-84 17:02
 MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER

SEQ 056

```

1666      |
1667      |
1668      |IMPLICIT INPUTS:
1669      |
1670      |      EXPMSG - EXPECTED MESSAGE BUFFER
1671      |      RECMG  - RECEIVED MESSAGE BUFFER
1672      |      RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1673      |      RCVLOAD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1674      |
1675      |      BGNMSG MSGSUB
1676      |MSGSUB::
1677      |      MOV      #10,,R0      ;SIZE OF WRITE SUBSYSTEM BUFFER
1678      |      JSR      PC,PRMSGEXP ;PRINT EXPD/RCV MESSAGE BUFFERS
1679      |      ENDMSG
1680      |
1681      |L10015:
1682      |      TRAP      C#MSG
1683      |
1684      |
1685      |      .SBTTL MEMADD - PRINT MEMORY ADDRESS DATA ERROR
1686      |
1687      |PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
1688      |
1689      |IMPLICIT INPUTS:
1690      |
1691      |      ERRHI - MEMORY ERROR HIGH ORDER ADDRESS
1692      |      ERRLO - MEMORY ERROR LOW ORDER ADDRESS
1693      |      EXP   - EXPECTED DATA
1694      |      RECV  - RECEIVED DATA
1695      |
1696      |      BGNMSG MEMADD
1697      |MEMADD::
1698      |      JSR      PC,PRIADD      ;PRINT MEMORY ADDRESS IN ERROR
1699      |      MOV      EXPD,R1        ;GET EXPD DATA
1700      |      MOV      RECV,R2        ;GET RECEIVED DATA
1701      |      JSR      PC,PRIOR      ;PRINT EXPD/RCV
1702      |      ENDMSG
1703      |
1704      |L10016:
1705      |      TRAP      C#MSG
1706      |
1707      |      .SBTTL PRAMPKT - PRINT RAM AND PACKET DATA
1708      |
1709      |PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1710      |WHEN THE RAM DATA DOES NOT MATCH.
1711      |
1712      |INPUTS:
1713      |
1714      |      R4      POINTER TO COMMAND PACKET
1715      |
1716      |IMPLICIT INPUTS:
1717      |
1718      |      RAMDATA DATA AS READ FROM THE RAM
1719      |      RAMSIZ  NUMBER OF BYTES IN PACKET

```

F5

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
PRAMPKT - PRINT RAM AND PACKET DATA

SEQ 057

```

1717                                     IF RAMSIZ=0 THEN DEFAULT TO 8.
1718                                     ;
1719                                     ;IMPLICIT OUTPUTS:
1720                                     ;
1721                                     ;      RAMSIZ  SET TO 0
1722                                     ;
1723                                     ;
1724 PRAMPKT:
1725                                     SAVREG                                     ;SAVE R1-R5 UNTIL NEXT RETURN
1726 013606                               MOV      #RAMDATA,R1                     ;DATA FROM THE RAM
1727 013612 012701 002242'                CLR      R2                          ;INIT BYTE NUMBER
1728 013616 005002                        CMPB    (R1)+,(R4)+                    ;COMPARE EXPECTED, RECEIVED
1729 013620 122124                        BNE     7$                             ;BR IF NO MATCH
1730 013622 001005                        FORCERROR 7$,NOTSSR
1731 013634 000436                        BR      10$                             ;END
1732 013636 116105 177777                7$:  MOVB    -1(R1),R5                 ;GET RECV RAM DATA
1733 013642 116403 177777                MOVB    -1(R4),R3                 ;GET EXPD PACKET DATA
1734 013646                                XOR     R5,R3                      ;XOR EXPD/RECV
1735 013656 042703 177400                BIC     #177400,R3                 ;LOW BYTE ONLY
1736 013662 116137 177777 002234'        MOVB    -1(R1),RECV                ;GET RECEIVED RAM DATA
1737 013670 116437 177777 002232'        MOVB    -1(R4),EXPD                ;GET EXPECTED RAM DATA
1738 013676                                PRINTB  #RAMASC,R2,RECV,EXPD,R3
1739                                     MOV     R3,-(SP)
1740                                     MOV     EXPD,-(SP)
1741                                     MOV     RECV,-(SP)
1742                                     MOV     R2,-(SP)
1743                                     MOV     #RAMASC,-(SP)
1744                                     MOV     #5,-(SP)
1745                                     MOV     SP,R0
1746                                     TRAP    C#PNTB
1747                                     ADD     #14,SP
1748                                     10$:  INC     R2                          ;UPDATE BYTE COUNT
1749                                     TST     RAMSIZ                          ;DEFAULT TO 8.?
1750                                     BEQ     15$                             ;BR IF YES
1751                                     CMP     R2,RAMSIZ                      ;DONE ALL BYTES?
1752                                     BLE     5$                             ;BR IF NO
1753                                     BR      25$                             ;
1754                                     15$:  CMP     R2,#8.                    ;DONE DEFAULT NUMBER OF BYTES?
1755                                     20$:  BLT     5$                             ;BR IF NO
1756                                     25$:  CLR     RAMSIZ                    ;SET DEFAULT RAMSIZ
1757                                     RTS      PC                          ;RETURN
1758
1759 013706 045 116 045 RAMASC: .ASCIZ 'N#A  BYTE: #D2#A RAM: #03#A Packet: #03#A XOR:#03'
1760                                     .EVEN
1761
1762                                     .SBTTL  PRMESS - PRINT CONTENTS OF MESSAGE BUFFER
1763                                     ;
1764                                     ;
1765                                     ; THIS ROUTINE PRINTS THE CONTENTS OF
1766                                     ; THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE
1767                                     ; TSV-05.
1768                                     ;
1769                                     ; INPUT:
1770                                     ;
1771                                     ;      R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
1772                                     ;      R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
1773                                     ; NOTE: R1 IS IGNORED IF KENABLE FLAG IS CLEAR

```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

SEQ 058

```

1765      ;
1766      ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
1767      ;
1768      ;-
1769
1770 014052      PRMESS:
1771 014052      SAVREG      ;SAVE THE REGISTERS
1772 014056 010005      MOV     R0,R5      ;SAVE LOW ORDER ADDRESS
1773 014060 005737 003134'      TST     KTENABLE      ;ADDRESS ABOVE 28K?
1774 014064 001001      BNE     10$      ;BR IF YES
1775 014066 005001      CLR     R1      ;SET HIGH ORDER ADDRESS TO 0
1776 014070 010103      10$:      MOV     R1,R3      ;SAVE HIGH ORDER ADDRESS
1777 014072 006100      ROL     R0      ;SHIFT BIT15 TO C BIT
1778 014074 006101      ROL     R1      ;SHIFT TO HIGH ORDER FOR PRINTOUT
1779 014076      PRINTX    @PROASC,R1,R5      ;PRINT MESSAGE BUFFER ADDRESS
      014076 010546      MOV     R5,-(SP)
      014100 010146      MOV     R1,-(SP)
      014102 012746 014230'      MOV     @PROASC,-(SP)
      014106 012746 000003      MOV     @3,-(SP)
      014112 010600      MOV     SP,R0
      014114 104415      TRAP    C:PNTX
      014116 062706 000010      ADD     @10,SP
1780 014122      PRINTX    @PRIASC      ;PRINT HEADER FOR CONTENTS
      014122 012746 014275'      MOV     @PRIASC,-(SP)
      014126 012746 000001      MOV     @1,-(SP)
      014132 010600      MOV     SP,R0
      014134 104415      TRAP    C:PNTX
      014136 062706 000004      ADD     @4,SP
1781 014142 005004      CLR     R4      ;NUMBER OF THE NEXT WORD
1782 014144 010501      MOV     R5,R1      ;COPY LOW ORDER ADDRESS
1783 014146 010300      MOV     R3,R0      ;COPY HIGH ORDER ADDRESS
1784 014150 001403      BEQ     20$      ;BR IF NOT ABOVE 28K
1785 014152 004737 017130'      JSR     PC,SETMAP      ;SETUP PAR ADDRESS IN R0
1786 014156 010005      MOV     R0,R5      ;GET PAR FORMAT ADDRESS ABOVE 28K
1787 014160      20$:      PRINTX    @PRASC,R4,(R5)+ ;PRINT THE CONTENTS OF MEMORY BUFFER
      014160 012546      MOV     (R5)+,-(SP)
      014162 010446      MOV     R4,-(SP)
      014164 012746 014333'      MOV     @PRASC,-(SP)
      014170 012746 000003      MOV     @3,-(SP)
      014174 010600      MOV     SP,R0
      014176 104415      TRAP    C:PNTX
      014200 062706 000010      ADD     @10,SP
1788 014204 005204      INC     R4      ;NUMBER OF THE NEXT
1789 014206 020427 000007      CMP     R4,@7      ;DONE ALL YET ?
1790 014212 003005      BGT     50$      ;BRANCH IF ALL DONE
1791 014214 002761      BLT     20$      ;PRINT FIRST 7 WORDS
1792 014216 032763 000200 000012      BIT     @X2,EXTF,XST2(R3);EXTENDED FEATURES ON ?
1793 014224 001355      BNE     20$      ;PRINT EXTENDED STATUS WORD
1794 014226 000207      50$:      RTS     PC      ;RETURN
1795
1796 014230      045      116      045      PROASC: .ASCIZ '##A Message Buffer Address = #01#05'
1797 014275      045      116      045      PRIASC: .ASCIZ '##A Message Buffer Contents:'
1798 014333      045      116      045      PRASC:  .ASCIZ '##A Word#D1#A: #0'
1799      .EVEN
1800
1801      .SBTTL PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
1802

```

H5

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
PRMSGEXP - PRINT EXPD/RCV MESSAGE BUFFERS

SEQ 059

```

1803
1804 ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
1805 ;
1806 ;      RO      - NUMBER OF WORDS IN BUFFER
1807 ;
1808 ;IMPLICIT INPUTS:
1809 ;
1810 ;      EXPMSG   - EXPECTED MESSAGE BUFFER
1811 ;      RECMSG   - RECEIVED MESSAGE BUFFER
1812 ;      RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1813 ;      RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1814 ;
1815 PRMSGEXP::
1816 SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
1817 MOV      RO,R5                      ;SAVE NUMBER OF WORDS
1818 MOV      RCVLOADD,RO                ;GET RECV LOW ADDRESS
1819 MOV      RO,R4                      ;COPY LOW ADDRESS
1820 MOV      RCVHIADD,R1               ;GET RECV HIGH ADDRESS
1821 ROL      RO                        ;SHIFT BIT15 TO C BIT
1822 ROL      R1                        ;SHIFT TO HIGH ORDER FOR PRINTOUT
1823 PRINTX   @PRMSG0,R1,R4             ;PRINT MESSAGE BUFFER ADDRESS
1824         MOV      R4,-(SP)
1825         MOV      R1,-(SP)
1826         MOV      @PRMSG0,-(SP)
1827         MOV      #3,-(SP)
1828         MOV      SP,RO
1829 TRAP     C:PNTX
1830 ADD      #10,SP
1831 PRINTX   @PRMSG1                   ;PRINT HEADER FOR CONTENTS
1832         MOV      @PRMSG1,-(SP)
1833         MOV      #1,-(SP)
1834         MOV      SP,RO
1835 TRAP     C:PNTX
1836 ADD      #4,SP
1837 CLR      R4                        ;NUMBER OF THE CURRENT WORD
1838 MOV      @EXPMSG,R1                ;GET EXPD BUFFER ADDRESS
1839 MOV      @RECMSG,R2                ;GET RECV BUFFER ADDRESS
1840 MOV      (R1),RO                   ;GET EXPD
1841 MOV      (R2),R3                   ;GET RECV
1842 XOR      RO,R3                     ;XOR EXPD/RCV
1843 PRINTX   @PRMSG2,R4,(R1),,(R2),,R3
1844         MOV      R3,-(SP)
1845         MOV      (R2),,-(SP)
1846         MOV      (R1),,-(SP)
1847         MOV      R4,-(SP)
1848         MOV      @PRMSG2,-(SP)
1849         MOV      #5,-(SP)
1850         MOV      SP,RO
1851 TRAP     C:PNTX
1852 ADD      #14,SP
1853 INC      R4                        ;NUMBER OF THE NEXT
1854 CMP      R4,R5                     ;DONE ALL YET?
1855 BGE      50$                       ;BR IF YES
1856 BR       20$                       ;DO ANOTHER
1857 RTS      PC                        ;RETURN
1858
1859 20$:
1860 50$:
1861
1862 1838 014542      045      116      045 PRMSG0: .ASCIZ '***A Message Buffer Address ***01*05'

```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS

SEQ 060

```

1839 014607      045      116      045 PRMSG1: .ASCIZ 'NNA Message Buffer Contents:'
1840 014645      045      116      045 PRMSG2: .ASCIZ 'NNA WORD #D2NA EXPD: #06NA RECV: #06NA XOR: #06'
1841
1842
1843              .SBTTL PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
1844
1845              ;+
1846              ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
1847              ; ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
1848
1849              ; RO - NUMBER OF BYTES IN BUFFER
1850
1851              ;IMPLICIT INPUTS:
1852
1853              ; EXPMSG - EXPECTED MESSAGE BUFFER
1854              ; RECMSG - RECEIVED MESSAGE BUFFER
1855
1856 014732 PRBYTEXP::
1857 014732 SAVREG
1858 014736 010005 MOV RO,R5 ;SAVE R1-R5 UNTIL NEXT RETURN
1859 014740 005037 002320' CLR PRMNO ;SAVE NUMBER OF BYTES
1860 014744 005004 CLR R4 ;INIT ERROR COUNT
1861 014746 012701 002322' MOV #EXPMSG,R1 ;NUMBER OF THE CURRENT BYTE
1862 014752 012702 002466' MOV #RECMSG,R2 ;GET EXPD BUFFER ADDRESS
1863 014756 111100 20$: MOVB (R1),R0 ;GET RECV BUFFER ADDRESS
1864 014760 042700 177400 BIC #C<377>,R0 ;GET EXPD BYTE
1865 014764 110037 015300' MOVB R0,PRBEXP ;CLEAR UPPER BYTE
1866 014770 111203 MOVB (R2),R3 ;SAVE FOR ERROR REPORT
1867 014772 042703 177400 BIC #C<377>,R3 ;GET RECV BYTE
1868 014776 110337 015302' MOVB R3,PRBREC ;CLEAR UPPER BYTE
1869 015002 XOR R0,R3 ;FOR ERROR REPORT
1870 015012 122122 CMPB (R1)+,(R2)+ ;XOR EXPD/RECV
1871 015014 001431 BEQ 30$ ;EXPD = RECV?
1872 015016 005237 002320' INC PRMNO ;BR IF YES
1873 015022 023727 002320' 000010 CMP PRMNO,#8. ;UPDATE ERROR COUNT
1874 015030 101023 BHI 30$ ;PRINTED 8?
1875 015032 27$: PRINTX #PRBMSG,R4,PRBEXP,PRBREC,R3 ;BR IF YES
1876 015032 010346 MOV R3,-(SP)
1877 015034 013746 015302' MOV PRBREC,-(SP)
1878 015040 013746 015300' MOV PRBEXP,-(SP)
1879 015044 010446 MOV R4,-(SP)
1880 015046 012746 015146' MOV #PRBMSG,-(SP)
1881 015052 012746 000005 MOV #5,-(SP)
1882 015056 010600 MOV SP,R0
1883 015060 104415 TRAP C$PNTX
1884 015062 062706 000014 ADD #14,SP
1885 015066 000404 FORCEEXIT 50$ 1880
1886 015076 30$: BR 35$ 1880
1887 015100 35$: FORCEERROR 27$,NOTSSR 1880
1888 015110 1880
1889 015110 005204 INC R4 ;NUMBER OF THE NEXT
1890 015112 020405 CMP R4,R5 ;DONE ALL YET?
1891 015114 002001 BGE 50$ ;BR IF YES
1892 015116 000717 BR 20$ ;DO ANOTHER
1893 015120 50$: PRINTX #PRBTOT,PRMNO ;PRINT TOTAL ERROR COUNT
1894 015120 013746 002320' MOV PRMNO,-(SP)

```

J5

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

SEQ 061

```

015124 012746 015233'      MOV    @PRBTOT, -(SP)
015130 012746 000002      MOV    @2, -(SP)
015134 010600      MOV    SP, R0
015136 104415      TRAP    C#PNTX
015140 062706 000006      ADD     @6, SP
1886 015144 000207      RTS     PC                ;RETURN
1887
1888 015146      045      116      045 PRBMSG: .ASCIZ 'N#A BYTE @D2#A EXPD: #03#A RECV: #03#A XOR: #03'
1889 015233      045      116      045 PRBTOT: .ASCIZ 'N#A NUMBER OF BYTES IN ERROR = #D2'
1890
1891 015300 000000      PRBEXP: .WORD 0                ;EXPD
1892 015302 000000      PRBREC: .WORD 0                ;RECV
1893
1894      .SBTTL EXPREC - PRINT EXPD/RECV WORD DATA
1895      ;*
1896      ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1897      ;
1898      ;INPUTS:
1899      ;
1900      ;
1901      ;      R1      RECEIVED DATA
1902      ;      R2      EXPECTED DATA
1903      ;
1904      ;-
1905
1906 015304      BGNMSG EXPREC
1907 015304 004737 007554' EXPREC: .JSR PC, PRIXOR                ;PRINT THE DATA
1908 015310      ENDMSG
1909 015310 104423      L10017: TRAP    C#MSG
1910
1911
1912      .SBTTL EXPBREC - PRINT EXPD/RECV BYTE DATA
1913      ;*
1914      ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
1915      ;
1916      ;INPUTS:
1917      ;
1918      ;
1919      ;      R1      RECEIVED DATA BYTE
1920      ;      R2      EXPECTED DATA BYTE
1921      ;
1922      ;-
1923
1924 015312      BGNMSG EXPBREC
1925 015312 004737 007424' EXPBREC: .JSR PC, PRIBXOR                ;PRINT THE DATA
1926 015316      ENDMSG
1927 015316 104423      L10020: TRAP    C#MSG
1928
1929
1930
1931

```

K5

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
EXPBREC - PRINT EXPD/RECV BYTE DATA

SEQ 062

```

1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953 015320
      015320
1954 015320 004737 013606'
1955 015324
      015324
      015324 104423
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980 015326
      015326
1981 015326 004737 010106'
1982 015332 004737 013606'
1983 015336
      015336

```

```

      .SBTTL  RAMERR  - PRINT RAM AND PACKET DATA
;+
;
;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
;
;INPUTS:
;
;      R4      POINTER TO COMMAND PACKET
;
;IMPLICIT INPUTS:
;
;      RAMDATA  DATA AS READ FROM THE RAM
;      RAMSIZ   NUMBER OF BYTES IN PACKET
;               IF RAMSIZ=0 THEN DEFAULT TO 8.
;
;IMPLICIT OUTPUTS:
;
;      RAMSIZ   SET TO 0
;-
;
;      BGNMSG   RAMERR
RAMERR::
;      JSR      PC,PRAMPKT      ;PRINT RAM/PACKET DATA
;      ENDMSG
L10021:
;      TRAP     C#MSG

      .SBTTL  RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
;+
;
;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
;
;INPUTS:
;
;      R4      POINTER TO COMMAND PACKET
;
;IMPLICIT INPUTS:
;
;      RAMDATA  DATA AS READ FROM THE RAM
;      RAMSIZ   NUMBER OF BYTES IN PACKET
;               IF RAMSIZ=0 THEN DEFAULT TO 8.
;      ERRHI    HIGH ORDER TEST ADDRESS
;      ERRLO    LOW ORDER TEST ADDRESS
;
;IMPLICIT OUTPUTS:
;
;      RAMSIZ   SET TO 0
;-
;
;      BGNMSG   RAMTADD
RAMTADD::
;      JSR      PC,PRITADD      ;PRINT TEST ADDRESS
;      JSR      PC,PRAMPKT      ;PRINT RAM/PACKET DATA
;      ENDMSG
L10022:

```

L5

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA

SEQ 063

```

1984 015336 104423 TRAP C$MSG
1985
1986 .SBTTL RAMEXP - PRINT RAM EXPD/RECV DATA
1987 ;+
1988 ;
1989 ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1990 ;
1991 ;INPUTS:
1992 ;
1993 ; R1 RECEIVED DATA
1994 ; R2 EXPECTED DATA
1995 ; R4 CONTROLLER RAM ADDRESS
1996 ;
1997
1998 015340 BGNMSG RAMEXP
1999 015340 RAMEXP::
2000 015340 042701 177400 BIC #C<377>,R1 ;SAVE EXPD RAM DATA BYTE
2001 015344 042702 177400 BIC #C<377>,R2 ;SAVE EXPD RAM DATA BYTE
2002 015350 004737 007700 JSR PC,PRIRAM ;PRINT THE RAM ADDRESS
2003 015354 004737 007554 JSR PC,PRIXOR ;PRINT THE DATA
2004 015360 ENDMMSG
2005 015360 L10023:
2006 015360 104423 TRAP C$MSG
2007
2008 .SBTTL TIMEXP - PRINT TIMER A,B AND EXP/REC
2009 ;+
2010 ;
2011 ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2012 ;AND TIMER A,B HEADER MESSAGE
2013 ;
2014 ;INPUTS:
2015 ;
2016 ; R1 RECEIVED DATA
2017 ; R2 EXPECTED DATA
2018 ;
2019
2020 BGNMSG TIMEXP
2021 TIMEXP::
2022 PRINTX #TIMSGO ;PRINT HEADER
2023 MOV #TIMSGO,-(SP)
2024 MOV #1,-(SP)
2025 MOV SP,R0
2026 TRAP C$PNTX
2027 ADD #4,SP
2028 JSR PC,PRIXOR ;PRINT THE DATA
2029 ENDMMSG
2030 L10024:
2031 015406 104423 TRAP C$MSG
2032
2033 015410 045 116 045 TIMSGO: .ASCIZ 'TIMER A STATUS IS IN BIT 3-TIMER B STATUS IS IN BIT 2'
2034 .EVEN
2035
2036 .SBTTL BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

```

M5

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

SEQ 064

```

2029      ;*
2030      ;
2031      ;PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
2032      ;
2033      ;INPUTS:
2034      ;
2035      ;      R1      CONTENTS OF TSSR
2036      ;      R2      DATA WRITTEN (8 BITS)
2037      ;
2038      ;--
2039
2040      BGNMSG  BADSSR
2041      BADSSR:
2042      MOV     R2, -(SP)          ;SAVE DATA TRANSFERRED
2043      BIC     #177400,R2        ;GET JUST ONE BYTE
2044      PRINTB  #XFERASC,R2
2045      MOV     R2, -(SP)
2046      MOV     #XFERASC, -(SP)
2047      MOV     #2, -(SP)
2048      MOV     SP, R0
2049      TRAP    C:PNTB
2050      ADD     #6, SP
2051      MOV     (SP), R2          ;RESTORE R2
2052      JSR     PC, PRITSSR       ;DECODE TSSR CONTENTS
2053      ENDMMSG
2054
2055      L10025:
2056      TRAP    C:MSG
2057      XFERASC: .ASCIZ  '%N%A Data Transferred = %03'
2058
2059      .SBTTL  GLOBAL SUBROUTINES SECTION
2060
2061      ;*
2062      ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2063      ; THAT ARE USED IN MORE THAN ONE TEST.
2064      ;--
2065
2066      .SBTTL  SOFINIT - SOFT INITIALIZE OF CONTROLLER
2067
2068      ;*
2069      ;ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
2070      ;BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
2071      ;THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
2072      ;DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
2073      ;
2074      ;INPUTS:
2075      ;
2076      ;      R5      ADDRESS OF FIRST REGISTER
2077      ;
2078      ;OUTPUTS:
2079      ;
2080      ;      R0      CONTENTS OF TSSR, IF ERROR
2081      ;      CARRY   SET IF INIT WAS OKAY
2082      ;              CLEAR IF FATAL ERROR
2083      ;
2084      ;CALLING SEQUENCE:

```

N5

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
SOFINIT - SOFT INITIALIZE OF CONTROLLER

SEQ 065

```

2077
2078      ;      MOV      ADDRESS,R5
2079      ;      JSR      PC,SOFINIT
2080      ;      BCS      CONTINUE
2081      ;      ERRDF      ;REPORT FATAL ERROR
2082      ;
2083      ;
2084      ;
2085 015604      SOFINIT::
2086 015604      SAVREG      ; SAVE THE REGISTERS
2087 015610 012765 000000 000002      MOV      @0,TSSR(R5)      ; DO THE INIT.
2088 015616 004737 016060      JSR      PC,WAITF      ; WAIT FOR SSR
2089 015622 016500 000002      MOV      TSSR(R5),R0      ;GET THE TSSR REGISTER
2090 015626 010004      MOV      R0,R4      ;TSSR CONTENTS
2091 015630 042704 176277      BIC      @C<HIADDR!OFL>,R4
2092 015634 052704 0022C0      BIS      @SSR!NBA,R4      ;R4 HAS EXPECTED CONTENTS
2093 015640 020400      CMP      R4,R0      ;ONLY EXPECTED BITS SET ?
2094 015642 001402      BEQ      5$      ;BRANCH IF OKAY
2095 015644 000241      CLC      ;CLEAR THE CARRY FOR ERROR
2096 015646 000401      BR      10$      ;GO TO EXIT
2097 015650 000261 5$:      SEC      ;SET THE CARRY BIT
2098 015652 000207 10$:      RTS      PC      ;RETURN TO CALLER
2099
2100      .SBTTL  CHKAMB - CHECK TSSR FOR AMBIGUITY
2101
2102      ;
2103      ;
2104      ;THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
2105      ;FOR AMBIGUITY
2106      ;
2107      ;INPUT:
2108      ;
2109      ;      R0      CONTENTS OF TSSR
2110      ;
2111      ;OUTPUT:
2112      ;
2113      ;      R0      CONTENTS OF TSSR
2114      ;
2115      ;      CARRY  SET - NO AMBIGUITY
2116      ;              CLR - AMBIGUOUS CONTENTS
2117      ;
2118      ;
2119      ;
2120 015654      CHKAMB:
2121 015654      SAVREG      ;SAVE THE GENERAL REGISTERS
2122 015660      MOV      R0,R4      ;CONTENTS OF TSSR
2123 015662 032700 100000      BIT      @SC,R0      ;IS BIT 15 SET ?
2124 015666 0010C4      BNE      5$      ;BRANCH IF YES
2125 015670 032700 174077      BIT      @C<NBA!OFL!SSR!HIADDR>,R0      ;ANY OTHER BITS SET ?
2126 015674 001023      BNE      40$      ;MUST BE AN ERROR
2127 015676 000424      BR      45$      ;RETURN WITH SUCCESS
2128 015700 032700 000200 5$:      BIT      @SSR,R0      ;IS READY BIT SET ?
2129 015704 001011      BNE      10$      ;BRANCH IF READY BIT IS SET.
2130 015706 032700 000040      BIT      @BIT5,R0      ;IS FATAL ERROR BIT SET ?
2131 015712 001414      BEQ      40$      ;ERROR IF NOT
2132 015714 042704 177761      BIC      @CTERCLS,R4      ;CLEAR ALL BUT TERMINATION CODE
2133 015720 020427 000016      CMP      R4,@16      ;ALL THREE BITS MUST BE SET

```

TSV3 - GLOBAL AREAS MACRO M113 01-FEB-84 17:02
 CHWAMB - CHECK TSSR FOR AMBIGUITY

SEQ 056

```

2134 015724 001007      BNE      401      ;ERROR IF NOT SET
2135 015726 000410      BR       451      ;OK IF ALL ARE SET
2136 015730 032700 000040 101:     BIT      @BIT5,R0      ;IS FATAL ERROR BIT SET ?
2137 015734 001405      BEQ      451      ;ERROR IF BIT IS SET WITH SSR
2138 015736 032700 000006      BIT      @BIT2|BIT1,R0      ;IS THIS A FUNCTION REJECT
2139 015742 001002      BNE      451      ;BR, IF TSSR IS OK
2140 015744 000241      CLC              ;AMBIGUOUS CONTENTS
2141 015746 000401      BR       501
2142 015750 000261      SEC              ;SHOW SUCCESS - NO AMBIGUITY
2143 015752 000207      RTS      PC      ;RETURN TO CALLER
2144
2145      .SBTTL ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS
2146
2147      ;
2148      ; DEFAULT DISPLAY INTERRUPT HANDLERS.
2149      ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
2150      ; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
2151      ;
2152      ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
2153      ;
2154      ; IOKCKIN=BIT7      ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
2155      ; IOKSTP=BIT0      ; EXPECT "STOP" INTERRUPT.
2156      ;
2157      ; INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
2158 015754      000      INTMASK: .BYTE 0
2159      ; INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
2160 015755      000      INTFLAG: .BYTE 0
2161
2162      ; SAVED INTERRUPT VECTOR:
2163 015756 000000      INTVEC: .WORD 0
2164      ; SAVE CPU PC
2165 015760 000000      INTCPC: .WORD 0
2166
2167      ; SUBROUTINE TO ENABLE INTERRUPTS:
2168 015762 010046      ENAINT: MOV      R0,-(SP)      ;SAVE R0
2169 015764 013700 002210'      MOV      IVEC,R0      ;GET POINTER TO VECTORS
2170 015770 012720 016026'      MOV      @INTR,(R0)      ;SET UP INTERRUPT VECTOR
2171 015774 012720 000340      MOV      @PRI07,(R0)
2172 016000 012600      MOV      (SP),R0      ;RESTORE R0
2173 016002 011646      MOV      (SP),-(SP)
2174 016004 012766 000000 000002      MOV      @0,2(SP)      ;SET CPU TO LEVEL 0
2175 016012 000002      RTI
2176
2177      ; SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
2178 016014 011646      DSBINT: MOV      (SP),-(SP)
2179 016016 012766 000340 000002      MOV      @PRI07,2(SP)
2180 016024 000002      RTI
2181
2182      .SBTTL INTR - INTERRUPT HANDLERS
2183
2184 016026      BGNSRV INTR      ;DEFINE INTERRUPT ENTRY
2185 016026 012737 000001 002224'      INTR1: MOV      @1,INTRECV      ;SET FLAG TO SHOW INTERRUPT RECEIVED
2186 016034 105037 015755'      CLRB      INTFLAG      ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2187 016040 132737 000001 015754'      BITB      @IOKSTP,INTMASK      ;EXPECTING STOP INTERRUPT?
2188 016046 001003      BNE      11      ;BR IF YES
2189 016050 152737 000001 015755'      BISB      @IOKSTP,INTFLAG      ;NO, SET THE ERROR FLAG.

```

```

2190
2191      ;SAVE REGISTERS, MSG BUFFER, ETC.
2192 016056 1:
2193 016056      ENDSRV
      016056 L10026:
      016056 000002      RTI
2194
2195      .SBTTL WAITF - WAIT FOR SUBSYSTEM READY
2196
2197      ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
2198
2199      ;INPUTS:
2200
2201      ;      R5      ADDRESS OF FIRST DEVICE REGISTER
2202
2203      ;OUTPUTS:
2204
2205      ;      R0      CONTENTS OF LAST TSSR READ
2206      ;      CARRY   SET - READY BIT SET
2207      ;              CLR - TIMEOUT WAITING FOR READY
2208
2209 016060 000401 WAITF:: BR      1:      ;NOP WHEN SUPER FIXED
2210 016062      BREAK      ; DO A SUPVSR BREAK FIRST.
      016062 104422      TRAP      C:BRK
2211 016064 012746 011000 1:      MOV      @11000, -(SP) ;25-APRIL-83 REV B - 1100 MSEC TIMER
2212 016070 016500 000002 2:      MOV      TSSR(R5),R0 ;READ THE TSSR REGISTER
2213 016074 105700      TSTB      R0      ;TEST FOR READY BIT SET
2214
2215 016076 100420      BMI      3:      ; EXIT ON STOP FLAG.
2216 016100      DELAY      1      ; WAIT 100 USEC
      016100 012727 000001      MOV      @1,(PC),
      016104 000000      .WORD      0
      016106 013727 002116      MOV      L:DLJ,(PC),
      016112 000000      .WORD      0
      016114 005367 177772      DEC      -6(PC)
      016120 001375      BNE      -.4
      016122 005367 177756      DEC      -22(PC)
      016126 001367      BNE      -.20
2217 016130 005316      DEC      (SP)      ;REDUCE DELAY COUNT
2218 016132 001356      BNE      2:      ;RETRY UNTIL TIMER EXPIRES
2219 016134 000241      CLC      ; C = 0, CONTROLLER STILL RUNNING...
2220 016136 000401      BR      4:      ;...OR HUNG-UP AFTER 300 MSEC.
2221 016140 000261 3:      SEC      ; C = 1, CONTROLLER IS STOPPED.
2222 016142 005326 4:      DEC      (SP),      ;RESTORE STACK WITHOUT CHANGING CARRY BIT
2223 016144 000207      RTS      PC
2224
2225      .SBTTL CHKTSSR - CHECK TSSR FOR READY
2226
2227
2228
2229      ; THIS ROUTINE WAITS FOR READY IN THE TSSR
2230      ; AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
2231
2232      ;INPUT:
2233
2234      ;      R5      ADDRESS OF CSR REGISTERS
2235

```

```

2236      ;OUTPUT:
2237      ;
2238      ;      R0      CONTENTS OF TSSR
2239      ;      CARRY   SET - OKAY
2240      ;
2241      ;      CLR - NOT READY AMBIGUOUS, OR SC SET
2242      ;
2243      ;
2244      016146      CHKTSSR:
2245      016146      004737      016060'      JSR      PC, WAITF      ;WAIT FOR READY
2246      016152      103014      BCC      20$      ;BRANCH IF TIME OUT
2247      016154      004737      015654'      JSR      PC, CHKAMB      ;TSSR AMBIGUOUS?
2248      016160      103006      BCC      10$      ;BR IF YES
2249      016162      032700      100000      BIT      *SC, R0      ;SPECIAL CONDITION SET?
2250      016166      001405      BEQ      15$      ;BR IF NO
2251      016170      032700      074000      BIT      *SCE!BIE!RMR!NXM>, R0      ;ANY ERROR BITS SET?
2252      016174      001402      BEQ      15$      ;BR IF NO
2253      016176      000241      10$:      CLC      ;SET FAILURE
2254      016200      000401      BR      20$      ;
2255      016202      000261      15$:      SEC      ;SET SUCCESS
2256      016204      000207      20$:      RTS      PC      ;RETURN TO CALLER
2257
2258      ;SBTTL  NXNM      - CHECK FOR NONEXISTENT MEMORY
2259
2260      ;*
2261      ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
2262      ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
2263      ; "C" = 0, ALL ADDRESSES OK.
2264
2265      ;CALL:  MOV ADR1, R1
2266      ;        MOV ADR2, R2
2267      ;        JSR PC, NXNM
2268      ;        RETURN      ;TEST "C" AND PROCEED.
2269      016206      012737      016242'      000004      NXNM:  MOV      *2$, *04      ; SET BUSERR VECTOR.
2270      016214      012737      000200      000006      MOV      *PRI04, *06
2271      016222      005003      CLR      R3      ;FLAG.
2272      016224      000241      CLC      ;CLEAR THE CARRY FOR NO NXM FOUND
2273      016226      005711      1$:      TST      (R1)      ;TEST THE ADDRESS(ES).
2274
2275      016230      020102      CMP      R1, R2      ;IF ANY TRAP, CONTINUE AT 2$.
2276      016232      001407      BEQ      3$      ;OTHERWISE, CONTINUE HERE.
2277      016234      062701      000002      ADD      *2, R1      ;BR IF FINISHED (NO NEXM'S).
2278      016240      000772      BR      1$      ;SET NEXT ADDRESS...
2279
2280      016242      005103      2$:      COM      R3      ;...AND CONTINUE.
2281      016244      012716      016252'      MOV      *3$, (SP)      ;GOT ONE, SET FLAG...
2282      016250      000002      RTI      ;...AND DISMISS INTERRUPT...
2283      016252      012700      000004      3$:      CLRVEC      *4      ;...AND GIVE BACK THE VECTOR.
2284      016256      104436      MOV      *4, R0
2285      016260      005703      TRAP      C1CVEC
2286      016262      001401      TST      R3      ;DID WE CATCH ONE ??
2287      016264      000261      BEQ      *+4      ;NO, "C" = 0, SKIP NEXT.
2288      016266      000207      SEC      ;YES, "C" = 1, (R1) = NEXM ADDR.
2289      RTS      PC
2290

```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 XNXM - CHECK FOR NONEXISTENT MEMORY

SEQ 069

```

2291
2292
2293
2294
2295
2296
2297
2298
2299
2300 016270
2301 016270 005737 002170'
2302 016274 001006
2303 016276 005737 002204'
2304 016302 100403
2305 016304 005337 002216'
2306 016310 001002
2307 016312 000241
2308 016314 000401
2309 016316 000261
2310 016320 000207
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338 016322
2339 016322 010046
2340 016324 005037 003154'
2341 016330 005037 016570'
2342 016334 005037 005604'
2343 016340 105037 015754'
2344 016344 013700 002202'
2345 016350 006300
2346 016352 005737 003114'
2347 016356 001430

      .SBTTL TSTLOOP - CHECK ITERATION COUNT
      ;
      ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
      ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
      ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
      ;
      ; CALL: LOOPTO ARG
      ;
TSTLOOP:
      TST      NOITS          ; ITERATIONS INHIBITED?
      BNE      1$             ; YES.
      TST      QVP            ; NO.
      BMI      1$             ; LOOPS DISALLOWED IN QUICK PASS.
      DEC      LOOPCNT        ; BUMP LOOP COUNTER.
      BNE      2$
1$:      CLC                  ; LOOP DISALLOWED, OR DONE.
      BR       3$
2$:      SEC                  ; LOOP ENABLED.
3$:      RTS      PC

      .SBTTL TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
      ;
      ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
      ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
      ; IN THE CURRENT RUN SEQUENCE.
      ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
      ;
      ; INPUT:
      ;
      ; R0      POINTER TO TEST ID ASCIZ STRING
      ;
      ; OUTPUT:
      ;
      ; R5      ADDRESS OF FIRST DEVICE REGISTER
      ;
      ; IMPLICIT OUTPUTS:
      ;
      ; TSTCNT  UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
      ;
      ; SIDE EFFECTS:
      ;
      ; INTERRUPT LEVEL IS RAISED TO LEVEL OF
      ; THE DEVICE UNDER TEST
      ;
      ;
TSTSETUP:
      MOV      R0, -(SP)      ; SAVE THE TEST ID MESSAGE
      CLR      SIFLAG         ; CLEAR "SOFT INIT" FLAG
      CLR      ERRK           ; CLEAR LOCAL ERROR COUNTER.
      CLR      EXTA           ; CLEAR ERROR EXTENSION FLAG.
      CLR      INTMA3K        ; CLEAR INTERRUPT MASK (CHECK ERROR)
      MOV      UNITN, R0      ; GET THE UNIT NUMBER.
      ASL      R0              ; ... AND MAKE IT A WORD OFFSET.
      TST      NODEV          ; DID STARTUP FIND THE DEVICE?
      BEQ      4$             ; BR IF YES

```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS

SEQ 070

```

2348 016360 100010      BPL      3$      ; BR IF NOT IDLE
2349 016362 052760 160000 003176'  BIS      $160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
2350 016370      ERRDF      1,NXR,NXRERR ; NO DEVICE HERE -- PRINT IT
      016370 104455      TRAP      C$ERRDF
      016372 000001      .WORD      1
      016374 003731'      .WORD      NXR
      016376 005550'      .WORD      NXRRERR
2351 016400 000407      BR      2$
2352 016402 052760 160001 003176' 3$: BIS      $160001,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
2353 016410      ERRDF      2,NOINIT ; DEVICE NOT IDLE
      016410 104455      TRAP      C$ERRDF
      016412 000002      .WORD      2
      016414 004331'      .WORD      NOINIT
      016416 000000      .WORD      0
2354 016420 012737 177777 003112' 2$: MOV      $-1,DUFLG ; DROP THE UNIT
2355 016426      DODU      UNITN
      016426 013700 002202'      MOV      UNITN,R0
      016432 104451      TRAP      C$DODU
2356 016434      DOCLN
      016434 104444      TRAP      C$DOCLN ; ABORT THE PASS
2357 016436 000423      BR      5$
2358
2359 016440      4$: RFLAGS      R0 ; GET THE OPERATOR FLAGS.
      016440 104421      TRAP      C$RFLA
2360 016442 032700 001000      BIT      $PNT,R0 ; PRINT THE TEST NUMBERS?
2361 016446 001412      BEQ      1$ ; BR IF NO
2362 016450 011600      MOV      (SP),R0 ; GET THE ID MESSAGE
2363 016452      PRINTF      $TNAM,R0 ; DISPLAY THE TEST ID
      016452 010046      MOV      R0,-(SP)
      016454 012746 016516'      MOV      $TNAM,-(SP)
      016460 012746 000002      MOV      $2,-(SP)
      016464 010600      MOV      SP,R0
      016466 104417      TRAP      C$PNTF
      016470 062706 000006      ADD      $6,SP
2364 016474 005237 002214' 1$: INC      TSTCNT ; BUMP TEST COUNTER.
2365 016500      SETPRI      IPRI ; PRIORITY THAT OF DEVICE
      016500 013700 002212'      MOV      IPRI,R0
      016504 104441      TRAP      C$SPRI
2366 016506 005726      5$: TST      (SP)+ ; FIX UP THE STACK
2367 016510 013705 002206'      MOV      C$RADDR,R5 ; ADDRESS OF TSV REGISTERS ON UNIBUS
2368 016514 000207      RTS      PC
2369 016516 045 123 045 TNAM: .ASCIZ 'TSV Test'
2370
2371
2372      .SBTTL TSTEND - PRINT ERRORS RECEIVED
2373
2374 ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
2375 ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
2376
2377 016532      TSTEND: RFLAGS      R0
      016532 104421      TRAP      C$RFLA
2378 016534 030027 020000      BIT      R0,$IER
2379 016540 001412      BEQ      1$ ; BR IF "IER" NOT SET.
2380 016542      PRINTF      $ESUM,ERRK ; PRINT ERROR COUNT.
      016542 013746 016570'      MOV      ERRK,-(SP)
      016546 012746 016572'      MOV      $ESUM,-(SP)
      016552 012746 000002      MOV      $2,-(SP)

```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
TSTEND - PRINT ERRORS RECEIVED

SEQ 071

```

016556 010600      MOV      SP,RO
016560 104417      TRAP     C#PNTF
016562 062706      ADD      #6,SP
2381 016566 000207 1$:      RTS      PC
2382
2383 016570 000000      ERRK:    0          ; LOCAL ERROR COUNT.
2384 016572      045      101      040  ESUM:    .ASCIZ  /MA *DMA ERRORS/
2385 016611      105      122      122  EMAXDU: .ASCIZ  /ERROR LIMIT REACHED -- DROPPING UNIT/
2386                      .EVEN
2387
2388                      .SBTTL  INCERK - INCREMENT LOCAL ERROR COUNT
2389
2390                      ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
2391                      ;
2392 016656 005237 016570' INCERK: INC      ERRK          ; INCREMENT LOCAL ERROR COUNT
2393 016662 010046      MOV      RO,-(SP)      ; SAVE RO
2394 016664 013700 002202'      MOV      UNITN,RO      ; GET UNIT NUMBER,
2395 016670 006300      ASL      RO          ; ... AND MAKE IT A WORD OFFSET.
2396 016672 062700 003176'      ADD      #ERTABL,RO      ; RO GETS ADDRESS OF ERROR TABLE ENTRY.
2397 016676 005210      INC      (RO)          ; INCREMENT THE DEVICE ERROR COUNT
2398 016700 032710 007777      BIT      #7777,(RO)      ; DID WE OVERFLOW THE FIELD?
2399 016704 001001      BNE      1$          ; BR IF NO.
2400 016706 005310      DEC      (RO)          ; YES -- BACK IT UP TO 7777.
2401 016710 012600 1$:      MOV      (SP)+,RO      ; RESTORE RO
2402 016712 000207      RTS      PC          ; RETURN TO CALLER.
2403
2404 016714 010046      CKEMAX: MOV      RO,-(SP)      ; SAVE RO
2405 016716 013700 002202'      MOV      UNITN,RO      ; GET UNIT NUMBER
2406 016722 006300      ASL      RO          ; ... AND MAKE IT A WORD OFFSET
2407 016724 016000 003176'      MOV      ERTABL(RO),RO      ; GET ERROR TABLE ENTRY
2408 016730 042700 170000      BIC      #170000,RO      ; EXTRACT ERROR COUNT FIELD
2409 016734 020037 002174'      CMP      RO,GERRMAX      ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
2410 016740 103004      BHIS     1$          ; BR IF YES
2411 016742 023737 016570' 002172'      CMP      ERRK,LERRMAX      ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
2412 016750 103417      BLO      2$          ; BR IF NO
2413 016752      1$:      RFLAGS    RO          ; GET OPERATOR FLAGS
2414 016752 104421      TRAP     C#RFLA
2415 016754 032700 000040      BIT      #IDU,RO          ; IS DROPPING INHIBITED?
2416 016760 001013      BNE      2$          ; BR IF YES.
2417 016762 012737 177777 003112'      MOV      #-1,DUFLG      ; NO -- DROP THE UNIT
2418 016770      ERRDF      4,EMAXDU
2419 016770 104455      TRAP     C#ERDF
2420 016772 000004      .WORD    4
2421 016774 016611'      .WORD    EMAXDU
2422 016776 000000      .WORD    0
2423 017000      DODU      UNITN
2424 017000 013700 002202'      MOV      UNITN,RO
2425 017004 104451      TRAP     C#DODU
2426 017006      DOCLN
2427 017006 104444      TRAP     C#DCLN
2428 017010 012600 2$:      MOV      (SP)+,RO      ; RESTORE RO
2429 017012 000207      RTS      PC          ; RETURN TO CALLER
2430
2431                      .SBTTL  CKDROP - CHECK IF UNIT SHOULD BE DROPPED
2432
2433                      ; CHECK IF UNIT SHOULD BE DROPPED
2434
2435                      ;
2436

```

H6

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 CKDROP - CHECK IF UNIT SHOULD BE DROPPED

SEQ 072

```

2427 017014 010046          CKDROP: MOV    R0, -(SP)
2428 017016          FORCERROR 1$,NOTSSR
2429 017026          RFLAGS R0
      017026 104421          TRAP  C$RFLA
2430 017030 032700 000040          BIT    $IDU,R0
2431 017034 001010          BNE     1$
2432 017036 011600          MOV     (SP),R0
2433 017040 012737 177777 003112'  MOV     $-1,DUFLG
2434 017046          DODU      UNITN
      017046 013700 002202'  MOV     UNITN,R0
      017052 104451          TRAP  C$DODU
2435 017054          DOCLN          ;ABORT THE PASS
      017054 104444          TRAP  C$DOCLN
2436 017056 012600 1$: MOV     (SP)+,R0
2437 017060 000207          RTS     PC
2438
2439          .SBTTL  CONFIG - DETERMINE CONFIGURATION OF SYSTEM
2440          ;
2441          ; SUBROUTINE - DETERMINE CONFIGURATION OF TSU05 SYSTEM.
2442          ;
2443 017062          CONFIG:
2444 017062 004737 015604'          JSR     PC,SOFINIT
2445 017066 000207          RTS     PC
2446
2447          .SBTTL  KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT
2448          ;
2449          ; SUBROUTINE - ENABLE MEM MGT.
2450          ;
2451 017070 005737 003132'  KTON:  TST     KIFLG          ; GOT KT?
2452 017074 001403          BEQ     1$                  ; NO.
2453 017076 012737 000001 177572  MOV     $1,SRO          ; YES. ENABLE KT11.
2454 017104 000207 1$: RTS     PC
2455
2456
2457
2458          ;
2459          ; SUBROUTINE - DISABLE MEM MGT.
2460          ;
2461 017106 005737 003132'  KTOFF: TST     KIFLG          ; GOT KT11?
2462 017112 001405          BEQ     1$                  ; NO.
2463 017114 000240          NOP
2464 017116 000240          NOP
2465 017120 012737 000000 177572  MOV     $0,SRO          ; DISABLE KT.
2466 017126 000207 1$: RTS     PC
2467
2468          .SBTTL  SETMAP - SETUP PAR6 MAPPING
2469
2470          ;
2471          ;
2472          ; THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
2473          ; AN 22 BIT ADDRESS. THE OFFSET INTO THE PAGE
2474          ; IS RETURNED BIASED TO PAR6.
2475          ;
2476          ; INPUTS:
2477          ;
2478          ; R0      HIGH ORDER ADDRESS BITS
2479          ; R1      LOW ORDER ADDRESS BITS

```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 SETMAP - SETUP PAR6 MAPPING

SEQ 073

```

2480
2481
2482
2483
2484
2485
2486
2487 017130
2488 017130
2489 017134 005737 003132'
2490 017140 001433
2491 017142 010102
2492 000006
2493
2494
2495
2496 017174 042701 000177
2497 017200 020137 003132'
2498 017204 103011
2499 017206 010137 172354
2500 017212 042702 160000
2501 017216 062702 140000
2502 017222 010200
2503 017224 000261
2504 017226 000401
2505 017230 000241
2506 017232 000207
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524 017234
2525 017234
2526 017240 004737 017106'
2527 017244 010003
2528 017246 013701 003124'
2529 017252 013702 003126'
2530 017256 010321
2531 017260 005302
2532 017262 003375
2533 017264 005737 003132'
2534 017270 001502
2535 017272 004737 017070'
2536 017276 005000

;
; OUTPUTS:
;
; R0 OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
; CARRY SET IF SUCCESS
; CLR IF ERROR
;
; SETMAP:
; SAVREG ;SAVE R1-R4 UNTIL NEXT RETURN
; TST KTFLG ;SYSTEM HAVE ABOVE 28K?
; BEQ 10$ ;BR IF NO
; MOV R1,R2 ;SAVE LOW ORDER BITS
; .REPT 6
; ASR R0 ;CONVERT WORD ADDRESS TO 32W BLOCKS
; ROR R1 ;MAKE IT DOUBLE PRECISION
; .ENDR
; BIC #177,R1 ;ALINE FOR LOWER 4K BOUNDARY
; CMP R1,KTFLG ;HIGHER THAN EXISTING MEMORY?
; BHIS 10$ ;BR IF YES
; MOV R1,#KIPAR6 ;SETUP MAPPING REGISTER PAR6
; BIC #160000,R2 ;SETUP DISPLACEMENT IN PAGE
; ADD #140000,R2 ;ADD IN PAR6 BIAS
; MOV R2,R0 ;RETURN IN R0
; SEC ;SET SUCCESS
; BR 15$
10$: CLC ;SET FAILURE
15$: RTS PC ;RETURN

; SBTTL FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN
;
; FILL MEMORY WITH A BACKGROUND PATTERN
;
; INPUTS:
;
; R0 = BACKGROUND PATTERN
; FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
; KTFLG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
;
; OUTPUTS:
;
; NONE
;
; FILLMEM:
; SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
; JSR PC,KTOFF ;DISABLE KT.
; MOV R0,R3 ;COPY TEST PATTERN
; MOV FREE,R1 ;GET FIRST FREE LOCATION
; MOV FRESIZ,R2 ;SIZE OF FREE SPACE BELOW 28K.
10$: MOV R3,(R1)+ ;STORE A BACKGROUND WORD
; DEC R2 ;DONE ALL MEMORY IN FREE SPACE?
; BGT 10$ ;BR IF NO
; TST KTFLG ;GOT KT?
; BEQ 55$ ;NO, GET OUT.
; JSR PC,KTON ;YES, ENABLE KT.
; CLR R0 ;HIGH ORDER ADDRESS START

```

Address	Offset	Hex	Symbol	Instruction	Comment
2537	017300	013701	003152'	MOV	PST32W,R1 ;GET >28K START ADDRESS (IN 32W BLOCKS)
2538		000006		.REPT	6
2539				CLC	;CLEAR C BIT
2540				ROL	R1 ;CONVERT BLOCKS TO WORDS
2541				ROL	R0 ;MAKE IT DOUBLE PRECISION
2542				.ENDR	
2543	017350	004737	017130'	JSR	PC,SETMAP ;SETUP PAR6 MAPPING REGISTER
2544	017354	010320		MOV	R3,(R0)+ ;STORE TEST PATTERN IN >28K ADDRESS
2545	017356	020027	160000	CMP	R0,#160000 ;END OF PAR6 MAPPING AREA?
2546	017362	103774		BLO	30\$;BR IF NO
2547	017364	162700	020000	SUB	#20000,R0 ;BACKUP INTO PAR6 MAPPING BEGIN
2548	017370	062737	000200	ADD	#200,#KIPAR6 ;POINT TO NEXT 4K BLOCK >28K.
2549	017376	013705	003132'	MOV	KTFLG,R5 ;GET VALUE FROM MEMORY SIZER
2550	017402	042705	170000	BIC	#170000,R5 ;ONLY 18 BITS PASS
2551	017406	023705	172354	CMP	#KIPAR6,R5 ;END OF MEMORY?
2552	017412	001427		BEQ	50\$;BR IF YES
2553	017414	005737	003144'	TST	T23A ;PROCESSOR TYPE A
2554	017420	001407		BEQ	35\$;NO KEEP GOING
2555	017422	013704	177572	MOV	SRO,R4 ;GET SRO CONTENTS
2556	017426	042704	177761	BIC	#177761,R4 ;CLEAR ALL BUT PAGE NUMBER
2557	017432	022704	000016	CMP	#16,R4 ;SEE IF PAGE 7
2558	017436	001415		BEQ	50\$;EXIT IF THERE
2559	017440	005737	003146'	TST	T23B ;PROCESSOR TYPE B
2560	017444	001410		BEQ	45\$;NO KEEP GOING
2561	017446	023727	172354	CMP	#KIPAR6,#7600 ;REACHED 18 BITS?
2562	017454	103001		BHIS	40\$;YES
2563	017456	000403		BR	45\$;NO KEEP GOING
2564	017460	012737	000020	MOV	#20,SR3 ;SET MMU RELOCATION
2565	017466	000137	017354'	JMP	30\$;KEEP GOING ON ETC.
2566	017472	004737	017106'	JSR	PC,KTOFF ;DISABLE KT.
2567	017476	000207		RTS	PC

```

2568                                     .SBTTL  CMPMEM  - COMPARE MEMORY TO BACKGROUND PATTERN
2569
2570                                     ;+
2571                                     ; COMPARE MEMORY WITH A BACKGROUND PATTERN
2572                                     ;
2573                                     ; INPUTS:
2574                                     ;
2575                                     ;         RO = BACKGROUND PATTERN
2576                                     ;         FREE      = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2577                                     ;         KTFLG      = SET TO HIGHEST MEMORY LOCATION IF > 28K.
2578                                     ;
2579                                     ; OUTPUTS:
2580                                     ;
2581                                     ;         CARRY      - SET IF NO ERROR
2582                                     ;         CARRY      - CLR IF ERROR
2583                                     ;
2584                                     ; IMPLICIT OUTPUTS:
2585                                     ;
2586                                     ;         ERRHI      - ERROR HIGH ADDRESS
2587                                     ;         ERRLO      - ERROR LOW ADDRESS
2588                                     ;         EXPD       - EXPECTED DATA
2589                                     ;         RECV       - RECEIVED DATA
2590                                     ;
2591 CMPMEM:                                ;
2592 017500                                ;         SAVREG                                ; SAVE R1-R5 UNTIL NEXT RETURN
2593 017504 010003                        ;         MOV      RO,R3                                ; COPY TEST PATTERN

```

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN

SEQ 075

```

2594 017506 004737 017106'      JSR      PC,KTOFF      ;DISABLE KT.
2595 017512 013701 003124'      MOV      FREE,R1      ;GET FIRST FREE LOCATION
2596 017516 013702 003126'      MOV      FRESIZ,R2     ;SIZE OF FREE SPACE BELOW 28K.
2597 017522 020311          10$: CMP      R3,(R1)      ;FREE SPACE LOCATION EQUAL TO EXPD?
2598 017524 001411          BEQ      15$      ;BR IF YES
2599 017526 010137 002240'      MOV      R1,ERRLO     ;SAVE ADDRESS IN ERROR
2600 017532 005037 002236'      CLR      ERRHI      ;NO HIGH ADDRESS
2601 017536 010337 002232'      MOV      R3,EXPD     ;SAVE EXPD FOR ERROR REPORT
2602 017542 011137 002234'      MOV      (R1),RECV    ;SAVE RECV FOR ERROR REPORT
2603 017546 000474          BR       50$      ;
2604 017550 005721          15$: TST      (R1)+      ;POINT TO NEXT ADDRESS
2605 017552 005302          DEC      R2      ;DONE ALL MEMORY IN FREE SPACE?
2606 017554 003362          BGT      10$      ;BR IF NO
2607 017556 005737 003132'      TST      KTFLG      ; GOT KT?
2608 017562 001472          BEQ      55$      ; NO. GET OUT.
2609 017564 004737 017070'      JSR      PC,KTON      ; YES. ENABLE KT.
2610 017570 005000          CLR      R0      ;HIGH ORDER ADDRESS START
2611 017572 013701 003152'      MOV      PST32W,R1    ;GET >28K START ADDRESS (IN 32W BLOCKS)
2612          .REPT      6
2613          ROL      R1      ;CONVERT BLOCKS TO WORDS
2614          ROL      R0      ;MAKE IT DOUBLE PRECISION
2615          .ENDR
2616 017626 042701 000177      BIC      #177,R1      ;ALINE 4K BOUNDARY
2617 017632 010046          MOV      R0,-(SP)      ;SAVE HIGH ORDER
2618 017634 010146          MOV      R1,-(SP)      ;SAVE LOW ORDER
2619 017636 004737 017130'      JSR      PC,SETMAP    ;SETUP PAR6 MAPPING REGISTER
2620 017642 010004          MOV      R0,R4      ;COPY ADDRESS BIASED TO PAR6
2621 017644 012601          MOV      (SP),R1      ;RESTORE LOW ORDER IN NON PAR6 FORMAT
2622 017646 012600          MOV      (SP),R0      ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
2623 017650 020314          30$: CMP      R3,(R4)      ;ABOVE 28K LOCATION EQUAL EXPD?
2624 017652 001411          BEQ      32$      ;BR IF YES
2625 017654 010037 002236'      MOV      R0,ERRHI     ;SAVE HIGH ORDER IN ERROR
2626 017660 010137 002240'      MOV      R1,ERRLO     ;SAVE LOW ORDER IN ERROR
2627 017664 010337 002232'      MOV      R3,EXPD     ;SAVE EXPD FOR ERROR REPORT
2628 017670 011437 002234'      MOV      (R4),RECV    ;SAVE RECV FOR ERROR REPORT
2629 017674 000421          BR       50$      ;
2630 017676 062701 000002      32$: ADD      #2,R1      ;UPDATE NON PAR6 ADDRESS
2631 017702 005500          ADC      R0      ;MAKE IT DOUBLE PRECISION ADD
2632 017704 062704 000002      ADD      #2,R4      ;UPDATE PAR FORMAT ADDRESS
2633 017710 020427 160000      CMP      R4,#160000    ;END OF PAR6 MAPPING AREA?
2634 017714 103755          BLO      30$      ;BR IF NO
2635 017716 162704 020000      SUB      #20000,R4     ;BACKUP INTO PAR6 MAPPING BEGIN
2636 017722 062737 000200 172354 ADD      #200,#KIPAR6 ;POINT TO NEXT 4K BLOCK >28K.
2637 017730 023737 172354 003132' CMP      #KIPAR6,KTFLG ;END OF MEMORY?
2638 017736 101744          BLOS      30$      ;BR IF NO
2639 017740 004737 017106'      50$: JSR      PC,KTOFF    ;TURN OFF MEMORY MAPPING
2640 017744 000241          CLC          ;SET FAILURE
2641 017746 000403          BR       60$      ;
2642 017750 004737 017106'      55$: JSR      PC,KTOFF    ;TURN OFF MEMORY MAPPING
2643 017754 000261          SEC          ;SET SUCCESS
2644 017756 000207      60$: RTS      PC
2645
2646          .SBTTL  REGSAV - SAVE R1-R5 ON STACK
2647
2648          ;
2649          ;ROUTINE TO
2650          ;SAVE R1 THROUGH R5 ON THE STACK

```

L6

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 REGSAV - SAVE R1-R5 ON STACK

SEQ 076

```

2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666 017760
2667 017760 010446
2668 017762 010346
2669 017764 010246
2670 017766 010146
2671 017770 010546
2672 017772 016605 000012
2673 017776 004736
2674 020000 012601
2675 020002 012602
2676 020004 012603
2677 020006 012604
2678 020010 012605
2679 020012 000207
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700 020014
2701 020014
2702 020020
      020020 104443
      020022 000406
      020024 020050
      020026 000022
      020030 020052

```

```

; CALLING SEQUENCE:
;
;      JSR      R5,REGSAV
;
; THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
; THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
; THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
; REGISTERS.
;
; THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
; CALLED VIA A JSR PC INSTRUCTION
;
; -
REGSAV:
      MOV      R4,-(SP)
      MOV      R3,-(SP)
      MOV      R2,-(SP)
      MOV      R1,-(SP)
      MOV      R5,-(SP)
      MOV      10.(SP),R5
      JSR      PC,8(SP)+
      MOV      (SP)+,R1
      MOV      (SP)+,R2
      MOV      (SP)+,R3
      MOV      (SP)+,R4
      MOV      (SP)+,R5
      RTS      PC

      .SBTTL   GETPAT - GET 8 BIT PATTERN FROM OPERATOR
; +
;
; ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
;
; INPUTS:
;
;      NONE.
;
; OUTPUTS:
;
;      R0      OCTAL NUMBER FROM THE OPERATOR
;
; CALLING SEQUENCE:
;
;      JSR      PC,GETPAT
;
; -
GETPAT::
      SAVREG          ;SAVE THE GENERAL REGISTERS
1$:      GMANID  DATASC,PATDAT,0,377,0,377,NO
      TRAP      C$GMAN
      BR        10000$
      .WORD     PATDAT
      .WORD     T$CODE
      .WORD     DATASC

```

M6

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 GETPAT - GET 8 BIT PATTERN FROM OPERATOR

SEQ 077

```

020032 000377          .WORD 377
020034 000000          .WORD T$LOLIM
020036 000377          .WORD T$HILIM
020040
2703 020040          10000$: BNCOMPLETE 1$ ;RETRY IF ERROR
020040 103367          BCC 1$
2704 020042 013700 020050' MOV PATDAT,RO ;DATA PATTERN FROM OPERATOR
2705 020046 000207          RTS PC ;RETURN TO CALLER
2706
2707
2708 ;LOCAL DATA AREA
2709 ;-
2710
2711 020050 000000          PATDAT: .WORD 0 ;TEMPORARY STORAGE FOR DATA
2712 020052 105 116 124 DATASC: .ASCIZ 'ENTER DATA PATTERN'
2713          .EVEN
2714
2715          .SBTTL GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE
2716
2717 ;
2718 ;ROUTINE TO ISSUE A MENU AND GET
2719 ;THE OPERATOR'S RESPONSE.
2720 ;
2721 ;INPUTS:
2722 ;
2723 ; R0 ADDRESS OF ASCIZ STRING OF MENU
2724 ; R1 MAXIMUM ALLOWABLE OPERATOR RESPONSE
2725 ;
2726 ;OUTPUTS:
2727 ;
2728 ; R0 NUMBER OF THE OPERATOR'S SELECTION
2729 ;
2730 ;-
2731
2732 020076          GETSEL::
2733 020076          SAVREG          ;SAVE GENERAL REGISTERS
2734 020102 010002          MOV R0,R2 ;SAVE THE MENU ADDRESS
2735 020104 010203          1$: MOV R2,R3 ;START OF MENU STRING
2736 020106 005713          2$: TST (R3) ;END OF ASCII ?
2737 020110 001412          BEQ 3$ ;BRANCH IF ALL LINES DISPLAYED
2738 020112          PRINTF %SELASC,(R3)+ ;DISPLAY THE MENU
020112 012346          MOV (R3)+,-(SP)
020114 012746 020262'          MOV %SELASC,-(SP)
020120 012746 000002          MOV %2,-(SP)
020124 010600          MOV SP,R0
020126 104417          TRAP C$PNTF
020130 062706 000006          ADD %6,SP
2739 020134 000764          BR 2$
2740 020136          3$: GMANID MENASC,MENRES,D,-1,0,-1,NO
020136 104443          TRAP C$GMAN
020140 000406          BR 10001$
020142 020316'          .WORD MENRES
020144 000042          .WORD T$CODE
020146 020267'          .WORD MENASC
020150 177777          .WORD -1
020152 000000          .WORD T$LOLIM
020154 177777          .WORD T$HILIM

```

N6

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE

SEQ 078

```

020156      10001$:
2741 020156      BNCOMplete 1$      ;RETRY IF ERROR
      020156 103352      BCC 1$
2742 020160 013700 020316'      MOV MENRES,R0      ;GET THE OPERATOR'S REPLY
2743 020164 020001      CMP R0,R1      ;COMPARE TO MAXIMUM ALLOWED
2744 020166 101411      BLOS 5$      ;BRANCH IF OK
2745 020170      PRINTF 0MENERR      ;DISPLAY ERROR MESSAGE
      020170 012746 020214'      MOV 0MENERR,-(SP)
      020174 012746 000001      MOV 01,-(SP)
      020200 010600      MOV SP,R0
      020202 104417      TRAP C$PNTF
      020204 062706 000004      ADD 04,SP
2746 020210 000735      BR 1$      ;RETRY
2747 020212 000207      S$:      RTS PC      ;RETURN TO CALLER
2748 020214 045 116 045 0MENERR: .ASCIZ 'N/A *** Menu Selection Too Large ***'
2749 020262 045 116 045 SELASC: .ASCIZ 'N/T'
2750 020267 105 156 164 MENASC: .ASCIZ 'Enter Menu Selection: '
2751      .EVEN
2752 020316 000000      MENRES: .WORD 0
2753
2754      .SBTTL CHKMAN - CHECK MANUAL INTERVENTION LEGALITY
2755
2756      ;+
2757      ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
2758
2759      ;INPUT:
2760
2761      ;      NONE.
2762
2763      ;OUTPUT:
2764
2765      ;      CARRY 0      MANUAL INTERVENTION NOT ALLOWED
2766      ;      1      MANUAL INTERVENTION IS OK
2767
2768      ;SIDE EFFECTS:
2769
2770      ;      A MESSAGE IS DISPLAYED WARNING THAT TEST IS
2771      ;      NOT EXECUTED IF MANUAL INTERVENTION IS NOT
2772      ;      ALLOWED.
2773
2774      ;-
2775
2776 020320      CHKMAN:
2777 020320      SAVREG      ;SAVE THE REGISTERS
2778 020324      MANUAL      ;SEE IF MANUAL INTERVENTION OK
      020324 104450      TRAP C$MANI
2779 020326      BCOMplete 1$      ;BRANCH IF ALLOWED
      020326 103411      BCS 1$
2780 020330      PRINTF 0NOMAN      ;PRINT THE WARNING MESSAGE
      020330 012746 020354'      MOV 0NOMAN,-(SP)
      020334 012746 000001      MOV 01,-(SP)
      020340 010600      MOV SP,R0
      020342 104417      TRAP C$PNTF
      020344 062706 000004      ADD 04,SP
2781 020350 000241      CLC      ;CLEAR CARRY FOR ERROR
2782 020352 000207      1$:      RTS PC      ;RETURN
2783

```

B7

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
CHKMAN - CHECK MANUAL INTERVENTION LEGALITY

SEQ 079

```

2784 020354      045      116      045  NOMAN:  .ASCIZ  'NMA *** Manual Intervention not Allowed - Test Aborted ***'
2785                                     .even
2786
2787                                     .SBTTL  ENVIRN  - SETUP FREE DIAGNOSTIC SPACE
2788
2789                                     ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
2790                                     ;
2791 020450      ENVIRN: MEMORY  RO
2792 020450      104431      TRAP  C#MEM
2793 020452      010037      003124'      MOV  RO,FREE      ; GET 1ST FREE ADDRESS...
2794 020456      062737      000002      003124'      ADD  #2,FREE
2795 020464      011037      003126'      MOV  (RO),FRESIZ      ; ...AND WORD COUNT.
2796 020470      162737      000004      003126'      SUB  #4,FRESIZ
2797 020476      013702      002012'      MOV  L#UNIT,R2      ; GET NUMBER OF UNITS
2798 020502      162737      000007      003126'      10#: SUB  #7,FRESIZ      ; TAKE AWAY 7 WORDS PER UNIT
2799 020510      005302      DEC  R2
2800 020512      001373      BNE  10#
2801 020514      013700      003124'      MOV  FREE,RO      ; GET FIRST FREE ADDRESS
2802 020520      063700      003126'      ADD  FRESIZ,RO      ; POINT TO LAST FREE ADDRESS
2803 020524      162700      000002      SUB  #2,RO      ; BACKUP 1 WORD
2804 020530      010037      003130'      MOV  RO,FREEHI      ; STORE LAST FREE ADDRESS
2805 020534      000207      40#:  RTS  PC      ; RETURN
2806
2807                                     .SBTTL  KTINIT  - SETUP KT11 MEMORY MANAGEMENT REGISTERS
2808
2809                                     ;
2810                                     ; ROUTINE TO INIT KT-11
2811                                     ;
2812                                     ;
2813 020536      KTINIT:
2814 020536      005037      003132'      CLR  KTFLG      ; INIT >28K MEMORY FLAG
2815 020542      005037      003134'      CLR  KTENABLE      ; INIT TEST >28K FLAG
2816 020546      023727      002120'      001577      L#HIME,#1577      ; GOT ENOUGH MEMORY (>28K)?
2817 020554      101453      BLOS  9#      ; NO.
2818 020556      023727      002120'      001777      L#HIME,#1777      ; GOT ENOUGH MEMORY (>32K)?
2819 020564      101447      BLOS  9#      ; NO.
2820 020566      013700      000004      MOV  #ERRVEC,RO      ; SAVE OLD ERR VEC PTR.
2821 020572      012737      020664'      000004      MOV  #21,#ERRVEC      ; SET ERR VEC PTR.
2822 020600      005737      177572      TST  #SRO      ; GOT KT11?
2823 020604      000240      NOP
2824 020606      013737      002120'      003132'      MOV  L#HIME,KTFLG      ; (TRAP IF NO).
2825 020614      042737      000177      003132'      BIC  #177,KTFLG      ; YES. SET KT FLAG.
2826 020622      010037      000004      MOV  RO,#ERRVEC      ; RESTORE OLD ERR VEC PTR.
2827 020626      005000      CLR  RO      ; RO = AR DATA.
2828 020630      012701      172340      MOV  #KIPAR,R1      ; R1 = KI REGS PTR.
2829 020634      012761      077406      177740      10#: MOV  #77406,-40(R1)      ; SET DESCRIPTOR REG.
2830 020642      010021      MOV  RO,(R1)      ; SET KIPAR REG.
2831 020644      062700      000200      ADD  #200,RO      ; BUMP AR DATA BY "4K".
2832 020650      020027      002600      CMP  RO,#2000      ; AT "I/O"?
2833 020654      001367      BNE  1#      ; NO.
2834 020656      012741      177600      MOV  #177600,-(R1)      ; YES. SET KIPAR7 FOR I/O.
2835 020662      000410      BR  9#
2836
2837 020664      012716      020700'      2#:  MOV  #6#,(SP)      ; SET UP RETURN
2838 020670      000002      RTI      ; RTI TO NEXT LOCATION
2839

```

C7

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
KTINIT - SETUP RTI1 MEMORY MANAGEMENT REGISTERS

SEQ 080

```

2840
2841 020672 012716 020726' 3*: MOV #10*,(SP) ; SET UP RETURN
2842 020676 000002 RTI ; RTI TO NEXT LOCATION
2843
2844 020700 010037 000004 6*: MOV R0,#ERRVEC ; RESTORE OLD ERR VEC PTR.
2845
2846 020704 9*:
2847 020704 013700 000004 MOV #ERRVEC,R0 ; SAVE OLD ERR VEC PTR.
2848 020710 012737 020672' 000004 MOV #3*,#ERRVEC ; SET ERR VEC PTR.
2849 020716 042737 000001 170200 BIC #BIT0,#MMR0 ;BE SURE UNIBUS MAP IS OFF
2850 020724 000240 NOP
2851 020726 010037 000004 10*: MOV R0,#ERRVEC ; RESET VECTOR BACK TO ERROR POINTER
2852 020732 000207 RTS PC
2853
2854
2855
2856 ; SUBROUTINE TO SET EXTENDED FEATURES SWITCH
2857 ;
2858 ; Requires that SOFINIT and WRTHR have been done previous to call.
2859 ;
2860 ;
2861 ; INPUTS:
2862 ; R5 CURRENT UNIT NUMBER
2863 ; OUTPUTS:
2864 ; The Extended Features Switch is set.
2865 ;
2866 ;
2867 ;
2868 020734 INVERT::
2869
2870 020734 005737 002226' TST EXTFEA ; IS SWITCH SET?
2871 020740 001020 BNE 1* ; YES,EXIT STAGE RIGHT!(or the next one outa town!)
2872 020742 012737 100206 021010' MOV #100206,CMDPKT ; WRT SUB-SYS MEM CMD
2873 020750 012737 021020' 021012' MOV #WSMBK,CMDPKT+2 ; MSG BUF ADDR
2874 020756 012737 000006 021016' MOV #6,CMDPKT+6 ; BYTE COUNT
2875 020764 012737 100010 021020' MOV #100010,WSMBK ; INVERT THE SWITCH
2876 020772 012704 021010' MOV #CMDPKT,R4 ; SET CMDPKT INTO R4
2877 020776 004737 010472' JSR PC,WRTHR ; DO IT
2878 021002 000207 1*: RTS PC ; RETURN
2879
2880
2881 ; COMMAND PACKET.
2882 ;
2883
2884 021004 .BLKB 10-<,-TSV2&7>
2885
2886
2887 021010 000700 CMDPKT:: 0 ; 1ST WORD IS TSV3 COMMAND.
2888 021012 000000 0 ; 2ND WORD IS THE BUFFER LOW ADDRESS.
2889 021014 000000 0 ; 3RD WORD IS THE BUFFER HIGH ADDRESS.
2890 021016 000000 0 ; 4TH WORD IS THE BYTE/RECORD/FILE COUNT.
2891
2892
2893 ; WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
2894 ;
2895 021020 000000 WSMBK:: 0 ; 1ST WORD:: SEL 0
2896 021022 000000 0 ; 2ND WORD:: SEL 2
2897 021024 000000 0 ; 3RD WORD:: SEL 4
2898 ; EVEN

```

```

2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910 021026
2911
2912 021026
2913 021032 005037 003136'
2914 021036 005037 003140'
2915 021042 005037 003142'
2916 021046 032737 170000 002120'
2917
2918 021054 001050
2919 021056 005737 003146'
2920 021062 001407
2921 021064 023727 002120' 007777
2922 021072 103406
2923 021074 004737 021222'
2924 021100 000427
2925 021102 005737 003144' 14:
2926 021106 001413
2927 021110 023727 002120' 005777 24:
2928 021116 101027
2929 021120 023727 002120' 003777
2930 021126 103403
2931 021130 004737 021222'
2932 021134 000411
2933 021136 023727 002120' 001577 44:
2934 021144 103414
2935 021146 004737 021222'
2936 021152 062737 000077 003142'
2937 021160 032737 177774 003142' 134:
2938 021166 001014
2939 021170 005237 003136'
2940 021174 000411
2941 021176 000410
2942 021200
    021200 012746 005454'
    021204 012746 000001
    021210 010600
    021212 104417
    021214 062706 000004
2943 021220 000207
2944
2945
2946
2947
2948
2949
2950

```

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

MEMCK::

```

    SAVREG
    CLR NXMFLG
    CLR NXMLO
    CLR NXMHI
    BIT 0170000,L#HIME
    BNE 14:
    TST T23B
    BEQ 14:
    CMP L#HIME,07777
    BLO 24:
    JSR PC,NXMTST
    BR 134:
    TST T23A
    BEQ 44:
    CMP L#HIME,05777
    BHI 144:
    CMP L#HIME,03777
    BLO 44:
    JSR PC,NXMTST
    BR 134:
    CMP L#HIME,01577
    BLO 144:
    JSR PC,NXMTST
    ADD 077,NXMHI
    BIT 0177774,NXMHI
    BNE 154:
    INC NXMFLG
    BR 154:
    BR 154:
    PRINTF 0NOMEM
    MOV 0NOMEM,-(SP)
    MOV 01,-(SP)
    MOV SP,R0
    TRAP C#PNTF
    ADD 04,SP
    RTS PC
    RETURN

```

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

E7

TSV3 - GLOBAL AREAS MACRO M1113 01-FEB-84 17:02
 KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

SEQ 082

2951						
2952						
2953	021222	013701	002120'	NXMTST: MOV	L#HIME,R1	;GET TOP OF MEMORY
2954	021226	062701	000200		ADD #200,R1	;MAKE IT I/O BLOCK OR OTHER NXM
2955	021232	042701	000177		BIC #177,R1	
2956	021236	010102			MOV R1,R2	;RESAVE RESULTS
2957		000006			.REPT 6	
2958					ASL R1	;PUT IN PLACE FOR XFER
2959					.ENDR	
2960	021254	010137	003140'	MOV	R1,NXML0	;SAVE TEST ADDRESS LOW
2961		000012			.REPT 10.	
2962					ASR R2	;PUT IN PLACE FOR XFER
2963					.ENDR	
2964	021304	042702	177700	BIC	#177700,R2	;DON'T WANT ILA!
2965	021310	010237	003142'	MOV	R2,NXMHI	;SAVE TEST ADDRESS HIGH
2966	021314	001207		RTS	PC	;RETURN
2967						
2968						
2969						
2970						
2971	021316			ENDMOD		

TSV4 - MISCELLANEOUS SECTIONS MACRO M113 01-FEB-84 17:02
 KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

SEQ 083

```

6          .TITLE  TSV4 - MISCELLANEOUS SECTIONS
7
8 021316    BGNMOD  TSV4
9 021316    TSV4::
10
11          .SBTTL  PROTECTION TABLE
12          BGNPROT
13          L$PROT::
14          .WORD  -1, -1, -1, -1          ;NO DEVICE PROTECTION REQUIRED.
15          ENDPROT
16
17          .SBTTL  INITIALIZE SECTION
18
19          ;**
20          ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
21          ;AT THE BEGINNING OF EACH PASS.
22
23          ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
24          ;IF "CONTINUE", NOTHING IS REQUIRED.
25
26          ;--
27          ;*
28          ;INSERT TEMPORARY JUMP TO ODT
29          ;-
30          BGNINIT
31          L$INIT::
32          40$: CLR      EXTFEA
33             CLR      NXMFLG
34             MOV      @EPT1,EPRTSW          ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
35             CLR      SIFLAG                ;CLEAR "SOFT INIT" FLAG
36             CLR      KTENABLE              ;CLEAR TEST ABOVE 28K FLAG
37             CLR      RAMSIZ                ;CLEAR RAM SIZE FOR RAMERR ROUTINE
38             READEF   @EF,CONTINUE
39             MOV      @EF,CONTINUE,RO
40             TRAP     C$REFG
41             BNCOMPLETE 1$
42             BCC      1$
43             CMP      UNITN,L$UNIT          ;UNIT IN RANGE?
44             BHS      4$                    ;BR IF NO.
45             TST      DUFLG                ;DROPPED UNIT?
46             BMI      NXTU                 ;BR IF YES
47             MOV      UNITN,R1
48             ASL      R1
49             TST      ERTABL(R1)
50             BEQ      SETU
51             BIT      @BIT14,ERTABL(R1)    ;DROPPED?
52             BNE      NXTU
53             EXIT     INIT                 ;DO NOTHING IF "CONTINUE".
54             TRAP     C$EXIT
55             .WORD    L1003C
56             READEF   @EF,NEW
57             MOV      @EF,NEW,1$
58             TRAP     C$REFG
59             BNCOMPLETE NXTU
60             BCC      NXTU
61             READEF   @EF,START
62             ;TAKE NEXT UNIT IF NOT NEW PASS.

```

TSV4 - MISCELLANEOUS SECTIONS MACRO M1113 81-FEB-84 17:02

SEQ 084

```

021446 012700 000040      MOV     VER, START, RO
021452 104447      TRAP     C$PREFG
57 021454      BCOMPLETE 2$
021454 103404      BCS     2$
58 021456      READEF     REF, RESTART
021456 012700 000037      MOV     REF, RESTART, RO
021462 104447      TRAP     C$REFG
59 021464      BNCOMPLETE 31$
021464 103031      BCC     31$
60 021466      2$:
61 021466      BRESET
021466 104433      TRAP     C$RESET
62 021470 005037 002214'    CLR     TSTCNT
63 021474 005037 002222'    CLR     FATFLG
64 021500 005037 003144'    CLR     T23A
65 021504 005037 003146'    CLR     T23B
66      MOV     $340, -(SP)
67      MOV     $20$, -(SP)
68      JMP     0.0DT
69 021510 005037 003400'    CLR     SKIPT
70 021514      20$:
71 021514 012737 177777 002204'    MOV     $-1, QVP
72 021522 004737 020450'    JSR     PC, ENVIRN
73 021526 004737 020536'    JSR     PC, KINIT
74 021532 012700 003176'    MOV     $ERTABL, RO
75 021536 005020      30$:    CLR     (RO)+
76 021540 020027 003376'    CMP     RO, $ERTABE
77 021544 103774      BLO     30$
78 021546 000404      BR     4$
79 021550 005037 002204'    CLR     QVP
80 021554 000137 021624'    JMP     PASRPT
81
82 021560      4$:
83 021560 012737 177777 002202'    NEWPAS: MOV     $-1, UNITN
84 021566 005037 002220'    CLR     DEVCNT
85 021572      NXTU:    BREAK
021572 104422      TRAP     C$BRK
86 021574 005237 002202'    INC     UNITN
87 021600 023737 002202' 002012'    CMP     UNITN, L$UNIT
88 021606 103423      BLO     SETU
89 021610 012737 177777 003112'    MOV     $-1, DUFLG
90 021616 000401      BR     11$
91 021620      DOCLN
021620 104444      TRAP     C$DCLN
92 021622 000240      NOP
93 021624      11$:
94 021624 023727 002012' 000001    PASRPT: CMP     L$UNIT, $1
95 021632 101752      BLOS    NEWPAS
96 021634 005737 002220'    TST     DEVCNT
97 021640 001747      BEQ     NEWPAS
98 021642      RFLAGS    RO
021642 104421      TRAP     C$RFLA
99 021644 032700 000100      BIT     $ISR, RO
100 021650 001343      BNE     NEWPAS
101
102 021652      DORPT
021652 104424      TRAP     C$DRPT

```

```

;1ST PASS, BUS-INIT...
;BUS RESET.

```

```

;NUMBER OF TESTS RUN IN PASS
;CLEAR FATAL ERROR COUNT
;CLEAR PROCSSOR TYPE A FLAG
;CLEAR PROCSSOR TYPE B FLAG

```

```

;RETURN TO DEBUGGER
;ENTER THE DEBUGGER
;CLEAR THE SUBTEST "SKIPPER"

```

```

;...QUICK VERIFY...
;SET ENVIRONMENT.
;INITIALIZE KT MEMORY MANAGEMENT
;CLEAR THE ERROR TABLE

```

```

;GO REPORT THE STATUS

```

```

;INIT UNIT NUMBER...
;CLEAR COUNT OF DEVICES RUNNING

```

```

;...AND SET NEXT UNIT NUMBER.

```

```

;ABORT, NO MORE UNITS.

```

```

;HOW MANY UNITS SELECTED?
;BR IF ONLY 1
;ARE ANY STILL RUNNING?
;BR IF NO

```

```

;SHOULD WE PRINT STATISTICS
;BR IF NO

```

H7

TSV4 - MISCELLANEOUS SECTIONS MACRO M1113 01-FEB-84 17:02
INITIALIZE SECTION

SEQ 085

```

103 021654 000741
104 021656
105
106 021656 013700 002202'
    021662 104442
107 021664 013342
    021666 005037 003112'
108 021672 005237 002220'
109 021676 012001
110 021700 010137 002206'
111
112
113 021704 012001
114
115
116 021706 010137 002210'
117 021712 012721 016026'
118 021716 013721 002212'
119
120 021722
121
122
123
124
125
126
127
128 021722 013701 002202'
129 021726 006301
130 021730 052761 100000 003176'
131 021736 005037 005604'
132 021742 023727 002012' 000001
133 021750 101416
134 021752 104421
    021754 032700 001000
135 021760 001412
136 021762 013746 002202'
    021766 012746 022054'
    021772 012746 000002
    021776 010600
    022000 104417
    022002 062706 000006
138 022006
139 022006 005037 003114'
140 022012 013701 002206'
141 022016 010102
142 022020 062702 000002
143 022024 004737 016206'
144 022030 103005
145 022032 010137 003114'
146 022036 012737 177777 003112'
147 022044
148
149

10$: BR NEWPAS

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

      MOV (R0)+,R1 ;GET VECTOR ADDRESS.
      MOV (R0),R2 ;GET INTERRUPT PRIORITY
      MOV R2,IPRI ;SET INTERRUPT PRIORITY.
      MOV R1,IVEC ;SET INTERRUPT VECTOR POINTER...
      MOV #INTR,(R1)+ ;...VECTOR...
      MOV IPRI,(R1)+ ;...AND PRIORITY.

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

;
;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
;
      MOV UNITN,R1
      ASL R1
      BIS #BIT15,ERTABL(R1) ;SAY DEVICE RUNNING
      CLR EXTA ;CLEAR ERROR EXTENSION FLAG.
      CMP L$UNIT,#1 ;ARE WE TESTING MULTIPLE UNITS?
      BLOS 10$ ;BR IF NO.
      RFLAGS RO ;YES -- GET OPERATOR FLAGS.
      TRAP C$RFLA
      BIT #PNT,RO
      BEQ 10$ ;SHOULD WE PRINT UNIT #?
      PRINTF #PUNIT,UNITN ;BR IF NOT.
      MOV UNITN,-(SP) ;PRINT THE UNIT #
      MOV #PUNIT,-(SP)
      MOV #2,-(SP)
      MOV SP,RO
      TRAP C$PNTF
      ADD #6,SP

10$: CLR NODEV
      MOV CSRADDR,R1 ;ADDRESS OF FIRST REGISTER
      MOV R1,R2 ;START OF REGISTERS
      ADD #TSSR,R2 ;ADDRESS OF TSSR REGISTER
      JSR PC,XNXM ;TEST BOTH CONTROLLER REGISTERS...
      BCC 2$ ;...AND BR IF ALL OK.
      MOV R1,NODEV ;FLAG DEVICE AS NON-EXISTENT
      MOV #-1,DUFLG ;DRCP THIS UNIT.

2$:
;
;FINALLY, SET CPU PRIORITY AND WE'RE DONE.

```

TSV4 - MISCELLANEOUS SECTIONS MACRO M1113 01-FEB-84 17:02
INITIALIZE SECTION

SEQ 086

```

150
151 022044      5$:  SETPRI  #PRI00      ;ENABLE INTERRUPTS.
      022044 012700 000000      MOV    #PRI00,R0
      022050 104441      TRAP    C$SPRI
152 022052      L10030:  ENDINIT
      022052      TRAP    C$INIT
      022052 104411
153
154 022054      045      116      045 PUNIT: .ASCIZ  /NNNA***** TESTING UNIT #D2A *****/
155      .EVEN
156
157      .SBTTL  ADD AND DROP UNITS SECTIONS
158
159      ;**
160      ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
161      ; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
162      ; OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
163      ;--
164 022122      BGN AU
      022122      L$AU::
165 022122 010001      MOV    R0,R1      ; GET UNIT TO BE ADDED (R0)
166 022124 006301      ASL     R1          ; MAKE IT A WORD INDEX
167 022126 052761 100000 003176'      BIS    #100000,ERTABL(R1)      ; SET THE "ACTIVE" BIT
168 022134 042761 040000 003176'      BIC    #40000,ERTABL(R1)      ; CLEAR THE "DROPPED" BIT
169 022142      PRINTF  #1$,R0
      022142 010046      MOV    R0,-(SP)
      022144 012746 022170'      MOV    #1$,-(SP)
      022150 012746 000002      MOV    #2,-(SP)
      022154 010600      MOV    SP,R0
      022156 104417      TRAP    C$PNTF
      022160 062706 000006      ADD     #6,SP
170 022164      EXIT    AU
      022164 000167      .WORD   J$JMP
      022166 000026      .WORD   L10031-2-
171 022170      045      116      045 1$: .ASCIZ  /NNA UNIT #D2A ADDED/
172      .EVEN
173
174 022216      ENDAU      ; UNUSED.
      022216      L10031:
      022216 104452      TRAP    C$AU
175
176      ;**
177      ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
178      ; TO BE REMOVED FROM THE TEST LIST.
179      ;
180      ; SUPVSR DOES THE "DROPPING". THIS IS JUST TO TELL THE MAN.
181      ; "DROPPED" UNITS ARE RE-SELECTED ON OPERATOR "STA" OR "ADD"
182      ; COMMAND, OTHERWISE REMAIN INACTIVE. THE "DISPLAY" COMMAND
183      ; WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
184      ; WHICH ARE STILL ACTIVE.
185      ; UPON ENTRY, R0 CONTAINS THE UNIT TO BE DROPPED.
186 022220      BGN DU
      022220      L$DU::
187 022220 012737 177777 003112'      MOV    4-1,DUF$LG
188 022226 010001      MOV    R0,R1
189 022230 006301      ASL     R1
190 022232 052761 140000 003176'      BIS    #140000,ERTABL(R1)      ; SAY DROPPED

```

TSV4 - MISCELLANEOUS SECTIONS MACRO M1113 01-FEB-84 17:02
ADD AND DROP UNITS SECTIONS

SEQ 087

```

191 022240 000240 000240 000240      240,240,240      ; ??????????
192 022246      010046      PRINTF  #1$,R0
      022250 012746 022274'      MOV   RO,-(SP)
      022254 012746 000002      MOV   #1$,-(SP)
      022260 010600      MOV   #2,-(SP)
      022262 104417      MOV   SP,R0
      022264 062706 000006      TRAP  C$PNTF
193 022270      000167      ADD    #6,SP
      022272 000030      EXIT   DU
      022274      045      116      045 1$: .WORD  J$JMP
      022274      045      116      045 1$: .WORD  L10032-2-
194 022274      045      116      045 1$: .ASCIZ  /NWA UNIT #DWA DROPPED/
195      022274      045      116      045 1$: .EVEN
196 022324      104453      ENDDU
      022324      104453      L10032: TRAP  C$DU
197      022326      104453      ;++
198      022326      104453      ; AUTO-DROP CODE SECTION.
199      022326      104453      ;--
200 022326      104453      BGNAUTO
      022326      104453      L$AUTO:;
201 022326 013705 002206'      MOV   CSRADDR,R5      ;POINT TO DEVICE REGISTER
202 022332 012703 000550      MOV   #360.,R3      ;ENOUGH TIME FOR 2400' REEL TO REWIND
203 022336 004737 016060'      10$: JSR   PC,WAITF      ;WAIT FOR SSR TO SET
204 022342 103420      BCS    20$      ;LEAVE WHEN SSR IS SET
205 022344      012727 000372      DELAY 250.      ;WAIT FOR .25 SECONDS
      022344 012727 000372      MOV   #250.,(PC)+
      022350 000000      .WORD  0
      022352 013727 002116'      MOV   L$DLY,(PC)+
      022356 000000      .WORD  0
      022360 005367 177772      DEC   -6(PC)
      022364 001375      BNE    -4
      022366 005367 177756      DEC   -22(PC)
      022372 001367      BNE    -20
206 022374 005303      DEC   R3      ;BUMP COUNTER DOWN
207 022376 001357      BNE    10$      ;KEEP GOING
208 022400 004737 017014'      JSR   PC,CKDROP      ;TRY AND DROP UNIT
209 022404      017014'      20$: ENDAUTO      ; UNUSED.
210 022404      017014'      L10033:
      022404 104461      TRAP  C$AUTO
211      022406      104461      .SBTTL  CLEAN-UP AND REPORT CODING SECTIONS
212      022406      104461      ;++
213      022406      104461      ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
214      022406      104461      ; EXECUTED AT THE END OF EACH PASS (OR SUB-PASS).
215      022406      104461      ; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
216      022406      104461      ;--
217      022406      104461      BGNCLN
218      022406      104461      L$CLEAN:;
219 022406 013705 002206'      MOV   CSRADDR,R5      ;POINT TO DEVICE REGISTER
      022406 013705 002206'      TST   DUFLG      ;"DROPPED" FLAG IS SET ON...
220 022406 013705 002206'      BMI   1$      ;...AND GROSS CONTROLLER FAULT...
221 022412 005737 003112'      ;...DON'T TRY TO XCT CLEANUP CODE.
222 022416 100405      MOV   #0,TSSR(R5)      ;DO SOFT INIT
223      022416 100405      225 022420 012765 000000 000002
224      022416 100405      225 022420 012765 000000 000002
225 022420 012765 000000 000002

```

K7

TSV4 - MISCELLANEOUS SECTIONS MACRO M1113 01-FEB-84 17:02
CLEAN-UP AND REPORT CODING SECTIONS

SEQ 088

226	022426	004737	016060'		JSR	PC, WAITF	
227	022432			1\$:			
228	022432			2\$:	ENDCLN		
	022432			L10034:			
	022432	104412			TRAP	C\$CLEAN	
229							
230							
231							
232							
233	022434				BGNRPT		
	022434			L\$RPT::			
234	022434				PRINTS	#DEVSUM	
	022434	012746	022676'		MOV	#DEVSUM, -(SP)	
	022440	012746	000001		MOV	#1, -(SP)	
	022444	010600			MOV	SP, R0	
	022446	104416			TRAP	C\$PNTS	
	022450	062706	000004		ADD	#4, SP	
235	022454	010246			MOV	R2, -(SP)	
236	022456	010346			MOV	R3, -(SP)	
237	022460	010446			MOV	R4, -(SP)	
238	022462	012704	003176'		MOV	#ERTABL, R4	
239	022466	005003			CLR	R3	; GET START OF ERROR TABLE.
240	022470	011402		1\$:	MOV	(R4), R2	; CLEAR UNIT NUMBER
241	022472	001467			BEQ	4\$	; GET ERROR TABLE ENTRY & TEST IT.
242	022474	100066			BPL	4\$	; ZERO IF UNIT NOT RUN
243	022476	032702	040000		BIT	#BIT14, R2	; WAS UNIT DROPPED?
244	022502	001015			BNE	2\$	; BR IF YES
245	022504	042702	170000		BIC	#1C7777, R2	; GET ERROR COUNT FIELD
246	022510				PRINTS	#DEVONL, R3, R2	; PRINT
	022510	010246			MOV	R2, -(SP)	
	022512	010346			MOV	R3, -(SP)	
	022514	012746	022733'		MOV	#DEVONL, -(SP)	
	022520	012746	000003		MOV	#3, -(SP)	
	022524	010600			MOV	SP, R0	
	022526	104416			TRAP	C\$PNTS	
	022530	062706	000010		ADD	#10, SP	
247	022534	000446			BR	4\$	
248	022536	020227	160000	2\$:	CMP	R2, #160000	; WAS UNIT NON-EXISTENT?
249	022542	001012			BNE	3\$	; BR IF NO
250	022544				PRINTS	#DEVNXR, R3	
	022544	010346			MOV	R3, -(SP)	
	022546	012746	023003'		MOV	#DEVNXR, -(SP)	
	022552	012746	000002		MOV	#2, -(SP)	
	022556	010600			MOV	SP, R0	
	022560	104416			TRAP	C\$PNTS	
	022562	062706	000006		ADD	#6, SP	
251	022566	000431			BR	4\$	
252	022570	020227	160001	3\$:	CMP	R2, #160001	; WAS UNIT NOT READY AT STARTUP?
253	022574	001012			BNE	30\$	; BR IF NO.
254	022576				PRINTS	#DEVNRD, R3	
	022576	010346			MOV	R3, -(SP)	
	022600	012746	023065'		MOV	#DEVNRD, -(SP)	
	022604	012746	000002		MOV	#2, -(SP)	
	022610	010600			MOV	SP, R0	
	022612	104416			TRAP	C\$PNTS	
	022614	062706	000006		ADD	#6, SP	
255	022620	000414			BR	4\$	

L 7

TSV4 - MISCELLANEOUS SECTIONS MACRO M1113 01-FEB-84 17:02
CLEAN-UP AND REPORT CODING SECTIONS

SEQ 089

```

256 022622 042702 170000      30$: BIC      #C7777,R2
257 022626      PRINTS      #DEVDR0,R3,R2
      022626 010246      MOV      R2,-(SP)
      022630 010346      MOV      R3,-(SP)
      022632 012746 023146'    MOV      #DEVDR0,-(SP)
      022636 012746 000003      MOV      #3,-(SP)
      022642 010600      MOV      SP,R0
      022644 104416      TRAP      C$PNTS
      022646 062706 000010      ADD      #10,SP
258 022652 062704 000002      4$:  ADD      #2,R4
259 022656 005203      INC      R3
260 022660 020427 003376'    CMP      R4,#ERTABE
261 022664 103701      BLO      1$
262 022666 012604      MOV      (SP),R4
263 022670 012603      MOV      (SP),R3
264 022672 012602      MOV      (SP),R2
265 022674      ENCRPT      ; UNUSED.
      022674      L10035:
      022674 104425      TRA?    C$RPT
266
267
268 022676      045      116      045 DEVSUM: .ASCIZ /##ADEVICE STATUS SUMMARY:##N/
269 022733      045      101      040 DEVONL: .ASCIZ /##A UNIT #D3##A ONLINE, ERRORS = #D##N/
270 023003      045      101      040 DEVNXR: .ASCIZ /##A UNIT #D3##A DROPPED, NON-EXISTENT REGISTER##N/
271 023065      045      101      040 DEVNRD: .ASCIZ /##A UNIT #D3##A DROPPED, NOT READY AT STARTUP##N/
272 023146      045      101      040 DEVDR0: .ASCIZ /##A UNIT #D3##A DROPPED, ERRORS = #D##N/
273
274
275 023216      ENDMOD
276
277
278

```

M7

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
CLEAN-UP AND REPORT CODING SECTIONS

SEQ 090

```

1          .TITLE  TSV5 - HARDWARE TESTS
2
3
10 023216  BGNMOD  TSV5
    023216  TSV5::

16
24
25          .SBTTL  TEST  1: INITIALIZE AFTER WRITE CHARACTERISTICS
26
27          ; TEST DESCRIPTION:
28
29          ; This test verifies that a Hardware Initialize command
30          ; invoked after a Write Characteristics command sets up
31          ; the Command, Message and Characteristic image blocks
32          ; in the controller ram correctly.
33
34          ; TEST STEPS:
35
36          ; REPEAT FOR LOOPCNT
37          ; BEGIN
38          ; Do WRITE CHARACTERISTICS command.
39          ; If the NBA bit in the TSSR register is NOT=0 then Print Error.
40          ; Write to TSSR register to soft initialize the controller
41          ; If controller RAM 310-377 NOT=0 then Print Error
42          ; END
43          ; --
44
45
46 023216  BGNTST
    023216
52 023216  012700  023662'  MOV      #TST13ID,R0          ;ASCII MESSAGE TO IDENTIFY TEST
53 023222  004737  016322'  JSR      PC,TSTSETUP        ;DO INITIAL TEST SETUP
54 023226  012737  000012  002216'  MOV      #10.,LOOPCNT      ;PERFORM 10 ITERATIONS
55 023234  T13LOOP:
56 023234  004737  024136'  JSR      PC,T13REST        ;SET PACKET TO START-UP VALUES
57
58 023240  012703  002764'  MOV      #TSTBLK+10.,R3      ;START OF TEST DATA
59 023244  012704  023620'  MOV      #T13PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
60 023250  012764  000010  000006  MOV      #8.,PKBCNT(R4)      ;START WITH MINIMUM ALLOWABLE VALUE
61 023256  54:
62 023256  004737  015604'  JSR      PC,SOFINIT        ;WRITE TO TSSR TO SOFT INITIALIZE
63 023262  103405  BCS      10$          ;BR IF SOFT INIT OKAY
64 023264  010001  MOV      R0,R1        ;SAVE CONTENTS OF TSSR
65 023266  ERRDF  ERRNO,SFIERR,SFIMSG  ;DEVICE FATAL DURING INIT
    023266  104455  TRAP      C$ERDF
    023270  000144  .WORD    100
    023272  003646'  .WORD    SFIERR
    023274  011644'  .WORD    SFIMSG
66
67          ;Do WRITE CHARACTERISTICS command.
68 023276  005037  002222'  10$:  CLR      FATFLG          ;CLEAR FATAL ERROR FLAG
69 023302  010465  000000  MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
70 023306  004737  016146'  JSR      PC,CHKTSSR        ;WAIT FOR SSR TO SET
71 023312  FORCERROR 12$  ;000FORCE ERROR IF FORCER=1
72 023326  103407  BCS      15$          ;BR IF CARRY SET (GOOD RETURN)
73 023330  010001  MOV      R0,R1        ;SAVE CONTENTS OF TSSR
74 023332  NEXT,ERRNO

```

N7

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

SEQ 091

```

75 023332      12$:  ERRDF  ERRNO,T13SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
    023332 104455                                TRAP      C$ERDF
    023334 000145                                .WORD      101
    023336 024047'                                .WORD      T13SSR
    023340 011656'                                .WORD      PKTSSR
76 023342 005237 002222'
77 023346      15$:  INC    FATFLG      ;SET FATAL ERROR FLAG
    023346 104406                                ;LOOP ON ERROR, IF FLAG SET
    023346 104406                                TRAP      C$CLP1
78 023350 016501 000002      MOV    TSSR(R5),R1      ;GET THE CONTENTS OF TSSR
79 023354 012702 000200      MOV    $SSR,R2      ;EXPECTED CONTENTS OF TSSR
80 023360 032701 000100      BIT    $OFL,R1      ;IS OFF-LINE BIT SET ?
81 023364 001402      BEQ    25$      ;BRANCH IF NOT OFF-LINE
82 023366 052702 000100      BIS    $OFL,R2      ;SET OFF-LINE IN EXPECTED DATA
83
84      ;If the NBA bit in the TSSR register is NOT=0 then Print Error.
85 023372      25$:  FORCERROR      27$      ;880
86 023372      CMP    R2,R1      ;DOES EXPECTED MATCH RECEIVED ?
87 023406 020201      BEQ    30$      ;OKAY IF MATCH
88 023410 001404      NEXT,ERRNO
89 023412
90 023412      27$:  ERRHRD  ERRNO,T13NBA,PKTSSR      ;NBA NOT ZERO
    023412 104456                                TRAP      C$ERHRD
    023414 000146                                .WORD      102
    023416 023774'                                .WORD      T13NBA
    023420 011656'                                .WORD      PKTSSR
91 023422      30$:  CKLOOP      ;LOOP ON ERROR ?
    023422 104406                                TRAP      C$CLP1
92
93      ;Write to TSSR register to soft initialize the controller
94 023424      40$:  JSR    PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
95 023424 004737 015604'      FORCERROR      42$      ;880
96 023430      BCS    50$      ;BR IF SOFT INIT OKAY
97 023444 103405      MOV    R0,R1      ;SAVE CONTENTS OF TSSR
98 023446 010001      NEXT,ERRNO
99 023450
100 023450      42$:  ERRDF  ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
    023450 104455                                TRAP      C$ERDF
    023452 000147                                .WORD      103
    023454 003646'                                .WORD      SFIERR
    023456 011644'                                .WORD      SFIMSG
101
102      ;If controller RAM 310-377 NOT=0 then Print Error
103 023460 012704 000310      50$:  MOV    $310,R4      ;START WITH LOC 310
104 023464 005002      CLR    R2      ;MEMORY EXPECTED SHOULD BE 000000
105 023466 105065 000000      CLRB   TSDB(R5)      ;SET MAINTENANCE MODE
106 023472 004737 016146'      JSR    PC,CHKTSSR      ;WAIT FOR SSR READY
107 023476 010465 000000      60$:  MOV    R4,TSDB(R5)      ;SELECT RAM ADDRESS
108 023502 004737 016146'      JSR    PC,CHKTSSR      ;WAIT FOR SSR READY
109 023506 116501 000000      MOVB   TSBA(R5),R1      ;READ LOC CONTENTS
110 023512      FORCERROR      62$,NOTSSR      ;880
111 023522 120102      CMPB   R1,R2      ;CHECK MEMORY FOR 000000
112 023524 001406      BEQ    70$      ;BRANCH IF DATA OKAY
113 023526      NEXT,ERRNO
114 023526      62$:  ERRDF  ERRNO,T13MEM,RAMEXP      ;MEMORY NOT ZERO AFTER INIT.
    023526 104455                                TRAP      C$ERDF
    023530 000150                                .WORD      104
    023532 023735'                                .WORD      T13MEM

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

SEQ 092

```

115 023534 015340' 002222' 701: INC FATFLG ;SET THE FATAL ERROR FLAG .WORD RAMEXP
116 023536 005237 002222' 701: CKLOOP
117 023542 104406 ESCAPE TST ;EXTT ON FATAL ERROR TRAP C#CLP1
118 023544 104410 TRAP C#ESCAPE
119 023546 000436 .WORD L10036-.
120 023550 005204 821: INC R4 ;LOOK AT NEXT RAM LOC.
121 023552 020427 000400 CMP R4,0400 ;AT TOP OF RAM ADDRESS SPACE
122 023556 001347 BNE 601 ;BRANCH TILL ALL MEMORY TESTED
123
124 023560 005737 002222' TST FATFLG ;ANY FATAL ERRORS ?
125 023564 001402 BEQ 1601 ;BRANCH IF NOT
126 023566 004737 017014' JSR PC,CKDROP ;TRY TO DROP THE UNIT
127 023572 004737 016270' JSR PC,TSTLOOP ;DONE ALL ITERATIONS?
128 023576 103002 BCC 1651 ;BR IF YES
129 023600 000137 023234' JMP T13LOOP ;LOOP UNTIL ITERATION COUNT DONE
130 023604 1651: EXIT TST
131 023604 104432 TRAP C#EXIT
132 023606 000376 .WORD L10036-.
133
134 ;LOCAL STORAGE FOR THIS TEST
135 ;
136
137 023610 .BLKB 10-<.-TSV2&7>
138 023620 T13PACKET: ;COMMAND PACKET FOR TEST
139 023620 100004 .WORD 100004 ;WRITE CHARACTERISTICS COMMAND, WITH ACK
140 023622 023630' .WORD T13DATA ;ADDRESS OF CHARACTERISTICS BLOCK
141 023624 000000 .WORD 0
142 023626 000010 .WORD 8. ;STARTING VALUE OF BLOCK SIZE
143
144 023630 T13DATA: ;CHARACTERISTICS DATA BLOCK
145 023630 023642' .WORD T13BFR ;ADDRESS OF MESSAGE BUFFER
146 023632 000000 .WORD 0
147 023634 000016 .WORD 14. ;LENGTH OF MESSAGE BUFFER
148 023636 000000 000000 .WORD 0,0
149
150 023642 T13BFR: .BLKW 8. ;MESSAGE BUFFER
151
152 ;LOCAL TEXT MESSAGES FOR TEST
153 ;
154
155
156 023662 111 156 151 T13ID: .ASCII 'Initialization After WRITE CHARACTERISTICS'
157 023735 111 156 143 T13MEM: .ASCII 'Incorrect RAM Data After Init'
158 .EVEN
159 023774 127 122 111 T13NBA: .ASCII 'WRITE CHARACTERISTICS Command Not Accepted'
160 024047 103 157 156 T13SSR: .ASCII 'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
161
162
163
164 ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
165 ;
166 ;
167 ;

```


TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

SEQ 094

220	024272	004737	010472'		JSR	PC,WRCHRR		ISSUE WRITE CHARACTERISTICS	
221	024276	103405			BCS	11\$		BR, IF COMMAND ISSUED OK	
225	024300	010001			MOV	R0,R1		SAVE CONTENTS OF TSSR	
226	024302				ERRHRD	ERRNO,WRMSG,SFMSG		WRITE CHARACTERISTICS FAILED	
	024302	104456						TRAP	C\$ERHRD
	024304	000312						.WORD	202
	024306	005052'						.WORD	WRMSG
	024310	011644'						.WORD	SFMSG
227	024312			11\$:					
228	024312	005037	002222'		CLR	FATFLG		CLEAR FATAL ERROR FLAG	
229	024316	005037	002224'		CLR	INTRECV		CLEAR INTERRUPT RECEIVED FLAG	
230	024322	012704	024740'		MOV	#T:4PACKET,R4		SET UP NEW WRT. SUBSYS MEM. COMMAND	
231	024326	010465	000000		MOV	R4,TSD8(R5)		SET THE PACKET ADDRESS	
232	024332	004737	016146'		JSR	PC,CHKTSSR		WAIT FOR SSR TO SET	
233	024336	103407			BCS	15\$		BR IF CARRY SET (GOOD RETURN)	
234	024340	010001			MOV	R0,R1		SAVE CONTENTS OF TSSR	
238	024342				ERRDF	ERRNO,T14SSR,PKTSSR		DEVICE FATAL SSR FAILED TO SET	
	024342	104455						TRAP	C\$ERDF
	024344	000313						.WORD	203
	024346	025553'						.WORD	T14SSR
	024350	011656'						.WORD	PKTSSR
239	024352	005237	002222'		INC	FATFLG		SET FATAL ERROR FLAG	
240	024356			15\$:	CKLOOP			LOOP ON ERROR, IF FLAG SET	
	024356	104406						TRAP	C\$CLP1
241	024360				ESCAPE	SEG		BY-PASS SUBTEST IF FATAL ERROR	
	024360	104410						TRAP	C\$ESCAPE
	024362	000074						.WORD	10000\$-
242	024364	005737	002224'		TST	INTRELV		DID AN INTERRUPT OCCUR ?	
243	024370	001004			BNE	22\$		BRANCH IF YES	
247	024372				ERRHRD	ERRNO,T14NINT,PKTSSR			
	024372	104456						TRAP	C\$ERHRD
	024374	000314						.WORD	204
	024376	025643'						.WORD	T14NINT
	024400	011656'						.WORD	PKTSSR
248	024402	016501	000002	22\$:	MOV	TSSR(R5),R1		GET THE CONTENTS OF TSSR	
249	024406	012702	000200		MOV	#SSR,R2		EXPECTED CONTENTS OF TSSR	
250	024412	032701	000100		BIT	#OFL,R1		IS OFF-LINE BIT SET ?	
251	024416	001402			BEQ	25\$		BRANCH IF NOT OFF-LINE	
252	024420	052702	000100		BIS	#OFL,R2		SET OFF-LINE IN EXPECTED DATA	
253	024424	020201		25\$:	CMP	R2,R1		DOES EXPECTED MATCH RECEIVED ?	
254	024426	001404			BEQ	30\$		OKAY IF MATCH	
258	024430				ERRHRD	ERRNO,T14NBA,PKTSSR		NBA NOT ZERO	
	024430	104456						TRAP	C\$ERHRD
	024432	000315						.WORD	205
	024434	025402'						.WORD	T14NBA
	024436	011656'						.WORD	PKTSSR
259	024440			30\$:					
260	024440	004737	010724'	35\$:	JSR	PC,CKRAM		CHECK RAM TO MEMORY	
261	024444	103405			BCS	59\$		RAM OK GO ON	
265	024446				ERRHRD	ERRNO,PKTRAM,RAMERR		THEY DON'T MATCH	
	024446	104456						TRAP	C\$ERHRD
	024450	000316						.WORD	206
	024452	004741'						.WORD	PKTRAM
	024454	015320'						.WORD	RAMERR
266	024456				ENDSEG			END SEGMENT	
	024456							10000\$:	
	024456	104405						TRAP	C\$ESEG

[illegible]

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

SEQ 096

```

315 024610 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
319 024612      ERRDF   ERRNO,T14SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD    209
                                .WORD    T14SSR
                                .WORD    PKTSSR
                                SET
320 024622      15$:    CKLOOP      ;LOOP ON ERROR, IF FLAG
                                TRAP      C$CLP1
                                .WORD    10000$-
321 024624      ESCAPE  SEG      ;BY-PASS CHECKS IF FATAL ERROR
                                TRAP      C$ESCAPE
                                .WORD    10000$-
322 024630 005737 002224'    TST      INTRECV      ;DID AN INTERRUPT OCCUR ?
323 024634 001004          BNE      22$          ;BRANCH IF YES
327 024636          ERRHRD   ERRNO,T14NINT,PKTSSR
                                TRAP      C$ERHRD
                                .WORD    210
                                .WORD    T14NINT
                                .WORD    PKTSSR
328 024646 016501 000002    22$:    MOV      TSSR(R5),R1      ;GET THE CONTENTS OF TSSR
329 024652 012702 100206    MOV      #SC!SSR!TSREJ,R2      ;EXPECTED CONTENTS OF TSSR
330 024656 032701 000100    BIT      #OFL,R1      ;IS OFF-LINE BIT SET ?
331 024662 001402          BEQ      25$          ;BRANCH IF NOT OFF-LINE
332 024664 052702 000100    BIS      #OFL,R2      ;SET OFF-LINE IN EXPECTED DATA
333 024670 020201          CMP      R2,R1      ;DOES EXPECTED MATCH RECEIVED ?
334 024672 001404          BEQ      30$          ;OKAY IF MATCH
338 024674          ERRHRD   ERRNO,T142REJ,PKTSSR      ;COMMAND NOT REJECTED
                                TRAP      C$ERHRD
                                .WORD    211
                                .WORD    T142REJ
                                .WORD    PKTSSR
339 024704          30$:    JSR      PC,CKRAM
340 024704 004737 010724'    35$:    BCS      59$
341 024710 103405          ERRHRD   ERRNO,PKTRAM,RAMERR      ;CHECK RAM TO MEMORY
345 024712          104456      ;RAM OK GO ON
346 024722          000323      ;THEY DON'T MATCH
347 024724          004741'      TRAP      C$ERHRD
348 024724 104403          015320'      .WORD    212
349 024726          104405      .WORD    PKTRAM
350 024726          001246      .WORD    RAMERR
351 024732          .BLKB      10-<.-TSV2&7>
352 024740          T14PACKET:
353 024740          .WORD      100206
354 024740          100206      ;COMMAND PACKET FOR TEST
355 024740          100206      ;WRITE SUBSYSTEM MEM. COMMAND, WITH IE, ACK

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

SEQ 097

```

361 024742 024750'      .WORD  T14DATA      ;ADDRESS OF CHARACTERISTICS BLOCK
362 024744 000000      .WORD  0              ;
363 024746 000006      .WORD  6              ;STARTING VALUE OF BLOCK SIZE
364 024750      T14DATA:  ;CHARACTERISTICS DATA BLOCK
365 024750      T14BS0: .BYTE  0              ;BSEL0 BYTE
366 024751      T14BS1: .BYTE  0              ;BSEL1 BYTE
367 024752 000000      T14BS2: .WORD  0        ;BSEL1 WORD
368 024754 000000      .WORD  0              ;DATA
369 024756      T14BFR: .BLKW  128.           ;MESSAGE BUFFER
370
371
373 025356      T14PK2: .BLKB  10-<.-TSV2&7> ;COMMAND PACKET FOR TEST
375 025360      .WORD  100204                ;WRITE CHARA. MEM. CMND., WITH IE, ACK
376 025360 100204      .WORD  T14DTA         ;ADDRESS OF SELECT DATA BLOCK
377 025362 025370'      .WORD  0              ;
378 025364 000000      .WORD  8              ;STARTING VALUE OF BLOCK SIZE
379 025366 000010      .WORD
380
381
382 025370      T14DTA:  ;SELECT DATA BLOCK
383 025370 024756'      .WORD  T14BFR         ;ADDRESS OF MESSAGE BUFFER
384 025372 000000      .WORD  0              ;
385 025374 000400      .WORD  256.           ;LENGTH OF MESSAGE BUFFER
386 025376 000000 000000 .WORD  0,0
387
388
389
390      ;+
391      ;LOCAL TEXT MESSAGES FOR TEST
392      ;-
393 025402      127      122      111 T14N8A: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted'
394 025456      127      122      111 T142REJ: .ASCIZ 'WRITE SUBSYSTEM MEMORY Not Rejected With Non-Zero Mode Field'
395 025553      103      157      156 T14SSR: .ASCIZ 'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
396 025643      105      170      160 T14NINT: .ASCIZ 'Expected Interrupt Not Received On WRITE SUBSYSTEM MEMORY'
397 025735      111      156      143 T14TSBA: .ASCIZ 'Incorrect TSBA Address After WRITE SUBSYSTEM MEMORY'
398 026021      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 026066      T14REST:
410 026066      SAVREG
411 026072 012701 024740'  MOV  #T14PACKET,R1      ;SAVE THE REGISTERS
412 026076 012721 100206  MOV  #100206,(R1)+      ;START OF THE PACKET
413 026102 012721 024750'  MOV  #T14DATA,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK, IE
414 026106 005021          CLR  (R1)+              ;ADDRESS OF DATA BLOCK
415 026110 012721 000006  MOV  #6,(R1)+          ;EXTENDED ADDRESS
416 026114 005021          CLR  (R1)+              ;SIZE OF DATA BLOCK IN BYTES
417 026116 005021          CLR  (R1)+              ;CLEAR BSEL0 AND BSEL1
418 026120 005011          CLR  (R1)+              ;CLEAR SEL2
419 026122 000207          RTS  PC                  ;CLEAR DATA AREA
                                         ;RETURN

```

```

420
421
422 026124          T14RST:
423 026124          SAVREG
424 026130 012701 025360'  MOV    #T14PK2,R1      ;SAVE THE REGISTERS
425 026134 012721 100204'  MOV    #100204,(R1)+  ;START OF THE PACKET
426 026140 012721 025370'  MOV    #T14DTA,(R1)+  ;WRITE CHARA. WITH ACK, IE
427 026144 005021        CLR    (R1)+  ;ADDRESS OF CHARAISTICS DATA BLOCK
428 026146 012721 000010'  MOV    #8,(R1)+  ;EXTENDED ADDRESS
429 026152 012721 024756'  MOV    #T14BFR,(R1)+  ;SIZE OF DATA BLOCK IN BYTES
430 026156 005021        CLR    (R1)+  ;MESSAGE BUFFER ADDRESS
431 026160 012721 000400'  MOV    #256,(R1)+  ;LENGTH OF MESSAGE BUFFER
432 026164 005021        CLR    (R1)+
433 026166 005011        CLR    (R1)
434 026170 005037 024756'  CLR    T14BFR      ;CLEAR 1ST LOC IN MESSAGE BUFFER
435 026174 000007        RTS     PC      ;RETURN
436 026176          ENDTST
      026176
      026176 104401          L10037:    TRAP    C$ETST
437
438          .SBTTL TEST 3: DMA MEMORY ADDRESSING
439
440
441          ;**
442          ; TEST 3
443          ; TEST DESCRIPTION
444          ;
445          ; This test verifies that the controller can properly address and
446          ; access all available CPU memory (other than that occupied by the
447          ; diagnostic and diagnostic supervisor code) for both reading (DATI)
448          ; and writing (DATO). Verified are the LSI-11 Bus drivers for all
449          ; available address lines. Up to this point only 16 bits have been
450          ; used for DMA transfers.
451          ;
452          ; TEST STEPS
453          ;
454          ; REPEAT FROM 1 TO LOOPCNT
455          ; BEGIN
456          ; Do Subtest 1 - Verify GET STATUS selected locations
457          ; Do Subtest 2 - Verify message packets selected locations
458          ; Do Subtest 3 - Verify Characteristic data selected locations
459          ; Do Subtest 4 - Verify NXM to selected invalid addresses
460          ; END
461          ;
462          ;--
463
464 026200          BGNTST
      026200
469 026200 012700 030200'  MOV    #TST12ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
470 026204 004737 016322'  JSR    PC,TSTSETUP  ;DO INITIAL TEST SETUP
471 026210 012737 000012'  MOV    #10,LOOPCNT  ;PERFORM 10 ITERATIONS
472 026216 005237 003150'  INC     T3BFLG      ;SET TEST FLAG
473 026222 004737 021026'  JSR    PC,MEMCK     ;CHECK MEMORY
474
475 026226          T12LOOP:      ;LOOP ON TEST LABEL
476
477

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 3: SUBTEST 1: GET STATUS SELECTED LOCATIONS

SEQ 099

```

478 .SBTTL TEST 3: SUBTEST 1: GET STATUS SELECTED LOCATIONS
479
480 ;++
481 ; TEST 3: SUBTEST 1:
482 ;
483 ; SUBTEST DESCRIPTION.
484 ;
485 ; This subtest verifies the controller can fetch a get status
486 ; command from all available memory locations.
487 ; Two word blocks are tested one at a time by first setting
488 ; all available memory to a background pattern of 125252.
489 ; A Get Status command is then executed to various addresses in
490 ; each available memory 4k word block. The various addresses
491 ; are determined by floating a 1 then a 0 through the address bits.
492 ;
493 ; TEST STEPS:
494 ;
495 ; BEGIN
496 ; Write to ISSR to soft initialize
497 ; Do a WRITE CHARACTERISTICS to setup a message buffer
498 ;
499 ; REPEAT FOR SELECTED VALID ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
500 ; BEGIN
501 ; Get a valid modulo-4 test address
502 ; Do a GET STATUS command from the test address
503 ; END
504 ; END
505 ;--
506
507 BGNSUB ;//////////////// BEGIN SUBTEST //////////////////
508 ; T3.1: TRAP C$BSUB
509
510 ;Write to ISSR to soft initialize
511 JSR PC,SOFINIT ;DO SOFT INIT OF CONTROLLER
512 BCS 15$ ;BR IF SOFT INIT OK
513 NEXT.ERRNO
514 MOV R0,R1 ;SAVE CONTENTS OF ISSR
515 ERRDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
516
517 TRAP C$ERDF
518 .WORD 301
519 .WORD SFIERR
520 .WORD SFIMSG
521
522 ;Do a WRITE CHARACTERISTICS to setup a message buffer
523 15$:
524 MOV #T12PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
525 JSR PC,T12SWRT ;RESTORE PACKET TO STARTING VALUES
526 CLR KTENABLE ;TURN OFF KT-11
527 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS
528 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
529 FORCERROR 17$
530 BCS 20$ ;BR IF SSR SET IN CHKTSSR
531 MOV R0,R1 ;SAVE CONTENTS OF TSSR
532 NEXT.ERRNO
533 17$: ERRDF ERRNO,T12WRTSSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
534 TRAP C$ERDF
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600

```

```

026316 000456
026320 030302'
026322 011656'
528
529 ;Verify a Get Status can be fetched from each address
530 ;Get a valid modulo-4 test address
531 ;Do a GET STATUS command from the test address
532 026324 005037 002222' 20$: CLR FATFLG ;CLEAR FATAL ERROR FLAG
533 026330 005037 030040' CLR T12KT ;TEST ABOVE 28K SWITCH
534 026334 012702 030044' MOV @T12BLK,R2 ;POINT TO TEST PATTERN TABLE
535 026340 T121LOOP:
536 026340 005037 003134' CLR KTENABLE ;TURN OFF ABOVE 28K TEST FLAG
537 026344 012201 MOV (R2)+,R1 ;GET TEST PATTERN ADDRESS
538 026346 005000 CLR R0 ;ASSUME NO TEST ABOVE 28K
539 026350 005737 030040' TST T12KT ;TEST ABOVE 28K THIS TIME?
540 026354 001407 BEQ 25$ ;BR IF NO
541 026356 016200 177776 MOV -2(R2),R0 ;GET TEST PATTERN AGAIN
542 026362 042700 177774 BIC @C<A1716>,R0 ;SAVE 18 BIT ADDRESS ONLY
543 026366 012737 000001 003134' MOV @1,KTENABLE ;TURN ON ABOVE 28K TEST FLAG
544 026374 004737 031046' 25$: JSR PC,T12CONVERT ;CONVERT TEST PATTERN TO TEST ADDRESS
545 026400 103034 BCC 65$ ;BR IF INVALID PACKET ADDRESS
546 026402 013704 030034' MOV T12LOAD,R4 ;COPY CURRENT PACKET LOW ADDRESS
547 026406 013703 030032' MOV T12HIADD,R3 ;COPY CURRENT PACKET HIGH ADDRESS
548 026412 004737 031416' JSR PC,T12SETGET ;SETUP CURRENT PACKET TO GET STATUS
549 026416 042703 177774 BIC @C<A1716>,R3 ;SAVE ADDRESS BITS 17+16
550 026422 050304 BIS R3,R4 ;SETUP 18 BIT PACKET ADDRESS
551 026424 004737 017106' JSR PC,KTOFF ;TURN OFF KT-11
552 026430 010465 000000 MOV R4,TSD8(R5) ;SET THE PACKET ADDRESS TO EXECUTE
553 026434 004737 016146' JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
554 026440 FORCERROR 32$
555 026454 103405 BCS 40$ ;BR IF SSR SET IN CHKTTSSR
556 026456 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
557 026460 NEXT.ERRNO
558 026460 32$: ERRDF ERRNO,T12GETSSR,PKTGETS ;DEVICE FATAL SSR FAILED TO SET
026460 104455 TRAP C$ERDF
026462 000457 .WORD 303
026464 030226' .WORD T12GETSSR
026466 011674' .WORD PKTGETS
559 026470 40$: CKLOOP ;LOOP ON ERROR. IF FLAG SET
026470 104406 TRAP C$CLP1
560 026472 65$:
561 026472 FORCEEXIT 80$
562 026502 020227 030176' CMP R2,@T12TBE ;DONE ALL TSTBLK TEST PATTERNS?
563 026506 103002 BHIS 70$ ;BR IF YES
564 026510 000137 026340' JMP T121LOOP ;DO ANOTHER MODULO- 4 ADDRESS
565 026514 005737 030040' 70$: TST T12KT ;DONE ABOVE 28K TESTING TOO?
566 026520 003012 BGT 80$ ;BR IF YES
567 026522 005737 003132' TST KTF LG ;ANY MEMORY ABOVE 28K ON SYSTEM?
568 026526 001407 BEQ 80$ ;BR IF NO
569 026530 012737 000001 030040' MOV @1,T12KT ;SET SWITCH
570 026536 012702 030044' MOV @T12BLK,R2 ;RESET TEST PATTERN TABLE
571 026542 000137 026340' JMP T121LOOP ;DO ABOVE 28K TESTING
572 026546 004737 017106' 80$: JSR PC,KTOFF ;TURN OFF KT11
573 026552 ENDSUB ;////////// END SUBTEST ////////////
026552 L10043:
026552 104403 TRAP C$ESUB
574 026554 005737 002222' TST FATFLG ;ANY FATAL ERRORS ?

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 3: SUBTEST 1: GET STATUS SELECTED LOCATIONS

SEQ 101

```

575 026560 001402      REQ      100$      ;BRANCH IF NOT
576 026562 004737 017014' JSR      PC,CKDROP      ;TRY TO DROP THE UNIT
577 026566      100$;
578
579      .SBTTL TEST 3: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS
580
581      ;++
582      ; TEST 3: SUBTEST 2:
583      ; SUBTEST DESCRIPTION:
584      ;
585      ; This subtest verifies the controller can deposit message packets
586      ; to all available memory locations.
587      ; Write Characteristics commands are executed with message
588      ; buffer addresses set to various addresses in each available
589      ; memory location.
590      ; The various addresses are determined by floating a 1 then a 0
591      ; through the address bits.
592
593      ; TEST STEPS:
594
595      ; BEGIN
596      ; Write to TSSR to soft initialize
597      ; Do a WRITE CHARACTERISTICS to setup a message buffer to compare
598
599      ; REPEAT FOR SELECTED ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
600      ; BEGIN
601      ; Get a valid modulo-4 test address
602      ; Set the packet message buffer to the TEST ADDRESS
603      ; Do a WRITE CHARACTERISTICS
604      ; Restore the test message buffer to background pattern
605
606      ; END
607      ; END
608      ;--
609 026566      BGNSUB      ;////////// BEGIN SUBTEST ///////////
        026566      T3.2:
        026566 104402      TRAP      C$BSUB
610
611
612      ;Write to TSSR to soft initialize
613 026570 004737 015604' JSR      PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
614 026574 103405      BCS      15$      ;BR IF SOFT INIT = OK
615 026576      NEXT.ERRNO
616 026576 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
617 026600      ERDIF      ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
        026600 104455      TRAP      C$ERDF
        026602 000460      .WORD      304
        026604 003646'      .WORD      SFIERR
        026606 011644'      .WORD      SFIMSG
618
619      ;Do a WRITE CHARACTERISTICS to setup a message buffer to compare
620 026610 15$;
621 026610 012704 027770' MOV      #T12PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
622 026614 004737 031350' JSR      PC,T12SWRT      ;SET PACKET TO WRITE CHARACTERISTICS
623 026620 004737 017106' JSR      PC,KTOFF      ;TURN OFF KT-11
624 026624 010465 000000 MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS
625 026630 004737 016146' JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET

```

L8

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 3: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS

SEQ 102

```

626 026634          FORCERROR          17$
627 026650 103405   BCS          20$          ;BR IF SSR SET IN CHKTSSR
628 026652 010001   MOV          R0,R1          ;SAVE CONTENTS OF TSSR
629 026654          NEXT,ERRNO
630 026654          17$: ERRDF          ERRNO,T12WRTSSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
        026654 104455          TRAP          C$ERDF
        026656 000461          .WORD          305
        026660 030302'          .WORD          T12WRTSSR
        026662 011656'          .WORD          PKTSSR

631
632          ;Get a valid modulo-4 test address
633          ;Set the packet message buffer to the test address
634          ;Do a WRITE CHARACTERISTICS
20$:      CLR          FATFLG          ;CLEAR FATAL ERROR FLAG
        MOV          $T12BLK,R3          ;POINT TO TEST PATTERN TABLE
T122LOOP:
        MOV          (R3)+,R1          ;GET TEST PATTERN ADDRESS
        MOV          R1,R0          ;GET ADDRESS ALL "18 BITS"
        BIC          $177774,R0          ;LEAVE ONLY A17 AND A16
        BIC          $3,R1          ;GET RID OF A17 AND A16
        JSR          PC,T12CONVERT          ;CONVERT TEST PATTERN TO TEST ADDRESS
        BCS          25$          ;BR IF VALID MESSAGE BUFFER ADDRESS
        JMP          150$          ;GET ANOTHER TEST PATTERN TO TRY
25$:      MOV          $T12PACKET,R4          ;SET THE COMMAND PACKET ADDRESS
        JSR          PC,T12SWRT          ;SETUP T12PACKET TO WRITE CHAR.
        MOV          T12LOAD,R1          ;SETUP LOW ORDER MESSAGE BUFFER ADD.
        MOV          T12HIADD,T12DATA+2          ;SETUP HIGH ORDER MESSAGE BUFFER ADD.
        JSR          PC,KTOFF          ;TURN OFF KT-11
        MOV          R4,TSD(R5)          ;SET THE PACKET ADDRESS TO EXECUTE
        JSR          PC,CHKTSSR          ;WAIT FOR SSR TO SET
        FORCERROR          32$
        BCS          50$          ;BR IF SSR SET IN CHKTSSR
        MOV          R0,R1          ;SAVE CONTENTS OF TSSR
        NEXT,ERRNO
        32$: ERRDF          ERRNO,T12WRTSSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                TRAP          C$ERDF
                .WORD          306
                .WORD          T12WRTSSR
                .WORD          PKTSSR
657 027012          50$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
        027012 104406          TRAP          C$CLP1
658 027014          150$:
659 027014          FORCEEXIT          160$
660 027024 020327 030176'      CMP          R3,$T12TBE          ;DONE ALL TST12BLK TEST PATTERNS?
661 027030 103002          BHS          160$          ;BR IF YES
662 027032 000137 026674'      JMP          T122LOOP          ;DO ANOTHER MODULO- 4 ADDRESS
663 027036 004737 017106'      JSR          PC,KTOFF          ;TURN OFF KT11
664 027042          160$: ENDSUB          ;////////// END SUBTEST //////////
        027042          L10044:
        027042 104403          TRAP          C$ESUB
665 027044 005737 002222'      TST          FATFLG          ;ANY FATAL ERRORS ?
666 027050 001402          BEQ          180$          ;BRANCH IF NOT
667 027052 004737 017014'      JSR          PC,CKDROP          ;TRY TO DROP THE UNIT
668 027056          180$:
669
670          .SBTTL TEST 3: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS
671          ;++

```

```

672 ; TEST 3: SUBTEST 3:
673 ;
674 ; SUBTEST DESCRIPTION:
675 ;
676 ; This subtest verifies the controller can fetch a
677 ; Write Characteristics data block from all available
678 ; memory locations.
679 ; Write Characteristics commands are executed with
680 ; characteristic data blocks at various memory addresses.
681 ; The various memory addresses are determined by floating
682 ; a 1 then a 0 through the address bits.
683 ;
684 ; TEST STEPS:
685 ;
686 ; BEGIN
687 ; Write to TSSR to soft initialize
688 ;
689 ; REPEAT FOR SELECTED VALID ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
690 ; BEGIN
691 ; Get a valid test address
692 ; Set the test packet characteristics data pointer to the
693 ; test address.
694 ; Store expected characteristic data in test address block
695 ; Do a WRITE CHARACTERISTIC command
696 ;
697 ; END
698 ;
699 ;--
700 027056 BCNSUB ;//////////////// BEGIN SUBTEST //////////////////
    027056 ; T3.3:
    027056 104402 TRAP C$BSUB
701
702
703 ;Write to TSSR to soft initialize
704 027060 004737 015604' JSR PC,SOFINIT ;DO SOFT INIT OF CONTROLLER
705 027064 103405 BCS 20$ ;BR IF SOFT INIT = OK
706 027066 NEXT,ERRNO
707 027066 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
708 027070 ERDF, ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
    027070 104455 TRAP C$ERDF
    027072 000463 .WORD 307
    027074 003646' .WORD SFIERR
    027076 011644' .WORD SFIMSG
709
710 ;Get a valid test address
711 027100 005037 002222' 20$: CLR FATFLG ;CLEAR FATAL ERROR FLAG
712 027104 005037 030040' CLR T12KT ;TEST ABOVE 28K SWITCH
713 027110 012703 030044' MOV #T12BLK,R3 ;POINT TO TEST PATTERN TABLE
714 027114 T123LOOP:
715 027114 005037 003134' CLR KTENABLE ;TURN OFF ABOVE 28K TEST FLAG
716 027120 012301 MOV (R3)+,R1 ;GET TEST PATTERN ADDRESS
717 027122 010100 MOV R1,R0 ;GET ADC "SS ALL "18 BITS"
718 027124 042700 177774 BIC #177774,R0 ;LEAVE ONLY A17 AND A16
719 027130 042701 000003 BIC #3,R1 ;GET RID OF A17 AND A16
720 027134 005737 030040' TST T12KT ;TEST ABOVE 28K THIS TIME?
721 027140 001407 BEQ 25$ ;BR IF NO
722 027142 016300 177776 MOV -2(R3),R0 ;GET TEST PATTERN AGAIN

```

N8

TSVS - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 3: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS

SEQ 104

```

723 027146 042700 177774          BIC    #C<A1716>,R0          ;SAVE 18 BIT ADDRESS ONLY
724 027152 012737 000001 003134' 25$: MOV    #1,KTENABLE      ;TURN ON ABOVE 28K TEST FLAG
725 027160 004737 031046'          JSR    PC,T12CONVERT      ;CONVERT TEST PATTERN TO TEST ADDRESS
726 027164 103402          BCS    30$          ;BR IF VALID TEST ADDRESS
727 027166 000137 027270'          JMP    60$          ;GET NEXT TEST PATTERN
728                                     ;Set the test packet characteristics data pointer to the test address
729 027172 012704 027770'          30$: MOV    #T12PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
730 027176 004737 031350'          JSR    PC,T12SWRT          ;RESTORE PACKET TO STARTING VALUES
731 027202 013764 030034' 000002' MOV    T12LOADD,PKLOW(R4)    ;STORE CHAR. DATA PTR LOW ADDRESS
732 027210 013764 030032' 000004' MOV    T12HIADD,PKHI(R4)    ;STORE CHAR. DATA PTR HIGH ADDRESS
733 027216 004737 031460'          JSR    PC,T12CHAR          ;STORE EXPECTED DATA IN DATA BLOCK
734                                     ;Do a WRITE CHARACTERISTIC command
735 027222 004737 017106'          JSR    PC,KTOFF            ;TURN OFF KT-11
736 027226 010465 000000          MOV    R4,TSDB(R5)          ;SET THE PACKET ADDRESS TO EXECUTE
737 027232 004737 016146'          JSR    PC,CHKTSSR          ;WAIT FOR SSR TO SET
738 027236          FORCERROR    32$          ;
739 027252 103405          BCS    40$          ;BR IF SSR SET IN CHKTSSR
740 027254 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
741 027256          NEXT.ERRNO
742 027256          32$: ERRDF    ERRNO,T12WRTSSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
743 027256 104455          TRAP    C$ERDF
744 027260 000464          .WORD    308
745 027262 030302'          .WORD    T12WRTSSR
746 027264 011656'          .WORD    PKTSSR
747 027266          40$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
748 027266 104406          TRAP    C$CLP1
749 027270          60$:
750 027270 020327 030176'          CMP    R3,#T12TBE          ;DONE ALL TSTBLK TEST PATTERNS?
751 027274 103002          BHI     65$          ;BR IF YES
752 027276 000137 027114'          JMP    T123LOOP          ;DO ANOTHER MODULO- 4 ADDRESS
753 027302 005737 030040'          65$: TST    T12KT          ;DONE ABOVE 28K TESTING TOO?
754 027306 003012          BGT     70$          ;BR IF YES
755 027310 005737 003132'          TST    KTFLG          ;ANY MEMORY ABOVE 28K ON SYSTEM?
756 027314 001407          BEQ     70$          ;BR IF NO
757 027316 012737 000001 030040' MOV    #1,T12KT          ;SET SWITCH
758 027324 012703 030044'          MOV    #T12BLK,R3          ;RESET TEST PATTERN TABLE
759 027330 000137 027114'          JMP    T123LOOP          ;DO ABOVE 28K TESTING
760 027334 004737 017106'          70$: JSR    PC,KTOFF          ;TURN OFF KT11
761 027340          ENDSUB          ;////////////////// END SUBTEST ////////////////////
762 027340          L10045:
763 027340 104403          TRAP    C$ESUB
764 027342 005737 002222'          TST    FATELG          ;ANY FATAL ERRORS ?
765 027346 001402          BEQ     75$          ;BRANCH IF NOT
766 027350 004737 017014'          JSR    PC,CKDROP          ;TRY TO DROP THE UNIT
767 027354          75$:

```

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

;***
; TEST 3: SUBTEST 4:

; SUBTEST DESCRIPTION:

; This subtest verifies the NXM error bit in the TSSR
; register is set when attempting to fetch data (a characteristic
; data block) from selected nonexistent locations.
; If NXM fails to set it is likely that an LSI-11 Bus driver is
; failing to assert an address line.

```

773 | Addresses tested include all combinations of high-order address
774 | bits (i.e. bits 16-21).
775 | *****
776 |                                     CAUTION
777 |
778 | The LSI BUS drivers for all available address lines(16-21)
779 | are only checked when running on a PDP-11 system with more than
780 | 128K words of memory!
781 | *****
782 |
783 | TEST STEPS:
784 |
785 | BEGIN
786 |   Write to TSSR to soft initialize
787 |   Do a write characteristic command
788 |   Invert the extended features switch
789 |
790 | REPEAT FOR SELECTED NON-EXISTENT MEMORY ADDRESSES
791 |   BEGIN
792 |     Get an invalid test address
793 |     Set the test packet characteristics data pointer to the
794 |     test address.
795 |     Do a WRITE CHARACTERISTIC command
796 |     If TSSR register NXM bit not set then print error message
797 |   END
798 | END
799 |
800 |
801 | BGNSUB                                ;////////// BEGIN SUBTEST ///////////
      |                                     T3.4:
      |                                     TRAP    C#BSUB
802 |
803 |
804 | 54:
805 |   TST    NXMFLG                        ;GOT ENOUGH MEMORY?
806 |   BNE    104                                ;IF SET STAY
807 |   JMP    NOEXTF                          ;LEAVE IF NOT SET
808 |
809 |   Write to TSSR to soft initialize
810 |
811 | 104:   JSR    PC,SOFINIT                ;DO SOFT INIT OF CONTROLLER
812 |       BCS    114                        ;BR IF SOFT INIT = OK
813 |       NEXT.ERRNO
814 |       MOV    R0,R1
815 |       ERDF   ERRNO,SFIERR,SFIMSG       ;SAVE CONTENTS OF TSSR
      |                                     ;DEVICE FATAL ERROR DURING INIT
      |                                     TRAP    C#ERDF
      |                                     .WORD   309
      |                                     .WORD   SFIERR
      |                                     .WORD   SFIMSG
816 |
817 |   Do a WRITE CHARACTERISTIC command so to invert switch
818 |
819 | 114:   CKLOOP                          ;LOOP IF SELECTED
      |                                     TRAP    C#CLP1
820 |       MOV    #T12PACKET,R4
821 |       JSR    PC,T12SWRT
822 |       CLR    KTENABLE
      |                                     ;GET THE ADDRESS OF COMMAND PACKET
      |                                     ;RESTORE PACKET TO STARTING VALUES
      |                                     ;TURN OFF KT-11

```

TSV5 : HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

SEQ 106

```

823 027426 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS
824 027432 004737 016146'     JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
825 027436                FORCERROR      15#
826 027452 103405                BCS      17#      ;BR IF SSR SET IN CHKISSR
827 027454 010001                MOV      R0,R1      ;SAVE CONTENTS OF TSSR
828 027456                NEXT,ERRNO
829 027456 15#:                ERRDF      ERRNO,T12WRTSSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C#ERDF
                                .WORD      310
                                .WORD      T12WRTSSR
                                .WORD      PKTSSR
                                TRAP      C#CLP1
                                027456 104455
                                027460 000466
                                027462 030302'
                                027464 011656'
830 027466 17#:                CKLOOP                ;LOOP IF SELECTED
                                TRAP      C#CLP1
                                027466 104406
831 027470 004737 020734'     JSR      PC,INVERT      ;INVERT THE SWITCH
832
833 ;Get an invalid test address
834
835 027474 005037 002222'     20#:      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
836 027500 25#:
837 027500 013737 003142' 030032' MOV      NXMMHI,T12HIADD      ;SAVE TEST ADDRESS HIGH
838 027506 013737 003140' 030034' MOV      NXML0,T12LOADD      ;SAVE TEST ADDRESS LOW
839 027514 T124LOOP:
840
841 ;Set the test packet characteristics data pointer to the
842 ; test address.
843
844 027514 012704 027770'     30#:      MOV      #T12PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
845 027520 004737 031350'     JSR      PC,T12SWRT      ;RESTORE PACKET TO STARTING VALUES
846 027524 013764 030034' 000002     MOV      T12LOADD,PKLOW(R4)      ;STORE CHAR. DATA PTR LOW ADDRESS
847 027532 013764 030032' 000004     MOV      T12HIADD,PKHI(R4)      ;STORE CHAR. DATA PTR HIGH ADDRESS
848
849 ;Do a WRITE CHARACTERISTIC command
850 027540 004737 017106'     JSR      PC,KTOFF      ;TURN OFF KT-11
851 027544 010465 000000     MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
852 027550 004737 016060'     JSR      PC,WAITF      ;WAIT FOR SSR TO SET
853 027554                FORCERROR      32#
854 027570 103407                BCS      40#      ;BR IF SSR SET IN CHKISSR
855 027572 010001                MOV      R0,R1      ;SAVE CONTENTS OF TSSR
856 027574                NEXT,ERRNO
857 027574 32#:                ERRDF      ERRNO,T12WRTSSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C#ERDF
                                .WORD      311
                                .WORD      T12WRTSSR
                                .WORD      PKTSSR
                                027574 104455
                                027576 000467
                                027600 030302'
                                027602 011656'
858 027604 005237 002222'     40#:      INC      FATFLG      ;SET FATAL ERROR FLAG
859 027610 40#:                CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
                                027610 104406
860 027612                FORCERROR      45# NOTSSR
861 027622                ESCAPE      SUB      ;BY-PASS SUBTEST IF FATAL ERROR
                                TRAP      C#ESCAPE
                                .WORD      L10046-
                                027622 104410
                                027624 000076
862
863 ;If TSSR register NXM bit not set then print error message
864 027626 016501 000002     45#:      MOV      TSSR(R5),R1      ;GET TSSR CONTENTS
865 027632                FORCERROR      52#
866 027646 032701 004000     BIT      #NXM,R1      ;NXM SET?
867 027652 001020                BNE      60#      ;BR IF YES

```

Address	Offset	Value	Label	Operation	Comment
868	027654	005237	030032'	INC	T12HIADD
869	027660	022737	000004	CMP	44,T12HIADD
870	027666	001312		BNE	T124LOOP
871	027670			NEXT	ERRNO
872	027670	013737	030034'	MOV	T12LOADD,ERRLO
873	027676	013737	030032'	MOV	T12HIADD,ERRHI
874	027704			ERRHRD	ERRNO,T12NXM,ADDSSR
	027704	104456			
	027706	000470			
	027710	030737'			
	027712	011736'			
875					
876	027714		60\$:	CKLOOP	
	027714	104406			
877	027716		90\$:		
878	027716		NOEXTF:		
879	027716	004737	017106'	JSR	PC,KTOFF
880	027722			ENDSUB	
	027722				
	027722	104403			
881	027724	005737	002222'	TST	FATFLG
882	027730	001402		BEQ	100\$
883	027732	004737	017014'	JSR	PC,CKDROP
884	027736	004737	016270'	JSR	PC,TSTLOOP
885	027742	103002		BCC	105\$
886	027744	000137	026226'	JMP	T12LOOP
887	027750		105\$:		
888	027750	004737	017106'	JSR	PC,KTOFF
889	027754	005037	003150'	CLR	T38FLG
890	027760			EXIT	TST
	027760	104432			
	027762	001540			
891					
892					
893					
894					
895					
896					
898	027764			.BLKB	10-<.-TSV2&7>
900	027770		T12PACKET:		
901	027770	100004		.WORD	100004
902	027772	030000'		.WORD	T12DATA
903	027774	000000		.WORD	0
904	027776	000010		.WORD	8.
905					
906	030000		T12DATA:		
907	030000	030012'		.WORD	T128FR
908	030002	000000		.WORD	0
909	030004	000016		.WORD	14.
910	030006	000000	000000	.WORD	0,0
911					
912	030012		T128FR:	.BLKW	8.
913					
914	030032	000000	T12HIADD:	.WORD	0
915	030034	000000	T12LOADD:	.WORD	0
916	030036	000000	T12PAR6:	.WORD	0
917	030040	000000	T12KT:	.WORD	0

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

SEQ 108

918	030042	000000	T124TST:	.WORD	0	;ADDRESS TEST BIT
919			;			
920			;			
921			;			
922			;			
923			;			
924	030044	000001	T12BLK:	.WORD	000001	
925	030046	000002		.WORD	000002	
926	030050	000003		.WORD	000003	
927	030052	000005		.WORD	000005	
928	030054	000006		.WORD	000006	
929	030056	000007		.WORD	000007	
930	030060	000011		.WORD	000011	
931	030062	000012		.WORD	000012	
932	030064	000013		.WORD	000013	
933	030066	000021		.WORD	000021	
934	030070	000022		.WORD	000022	
935	030072	000023		.WORD	000023	
936	030074	000041		.WORD	000041	
937	030076	000042		.WORD	000042	
938	030100	000043		.WORD	000043	
939	030102	000101		.WORD	000101	
940	030104	000102		.WORD	000102	
941	030106	000103		.WORD	000103	
942	030110	000201		.WORD	000201	
943	030112	000202		.WORD	000202	
944	030114	000203		.WORD	000203	
945	030116	000401		.WORD	000401	
946	030120	000402		.WORD	000402	
947	030122	000403		.WORD	000403	
948	030124	001001		.WORD	001001	
949	030126	001002		.WORD	001002	
950	030130	001003		.WORD	001003	
951	030132	002001		.WORD	002001	
952	030134	002002		.WORD	002002	
953	030136	002003		.WORD	002003	
954	030140	004001		.WORD	004001	
955	030142	004002		.WORD	004002	
956	030144	004003		.WORD	004003	
957	030146	010001		.WORD	010001	
958	030150	010002		.WORD	010002	
959	030152	010003		.WORD	010003	
960	030154	020001		.WORD	020001	
961	030156	020002		.WORD	020002	
962	030160	020003		.WORD	020003	
963	030162	040001		.WORD	040001	
964	030164	040002		.WORD	040002	
965	030166	040003		.WORD	040003	
966	030170	100001		.WORD	100001	
967	030172	100002		.WORD	100002	
968	030174	100003		.WORD	100003	
969	030176	177777	T12TBE:	.WORD	177777	
970			;			
971			;			
972			;			
973			;			
974	030200	104	TST12ID:	.ASCIZ	'DMA Memory Addressing'	

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

SEQ 109

```

975 030226      103      157      156 T12GETSSR: .ASCIZ 'Contents of TSSR Incorrect After GET STATUS'
976 030302      103      157      156 T12WRTSSR: .ASCIZ 'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
977 030371      115      145      163 T12MSGBUF: .ASCIZ 'Message Buffer Contents Incorrect After WRITE CHARACTERISTICS'
978 030467      102      141      143 T12BKGD: .ASCIZ 'Background Pattern Disturbed By WRITE CHARACTERISTICS'
979 030555      105      170      160 T12NINT: .ASCIZ 'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
980 030646      127      162      151 T12DPR: .ASCIZ 'Write Characteristic data in ram does not match expected'
981 030737      124      123      123 T12NXM: .ASCIZ 'TSSR NXM bit failed to set when non-existent memory address specifi
ed'

982                                     .EVEN
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004 031046
1005 031046
1006 031052 005037 030034'
1007 031056 005037 030032'
1008 031062 005037 030036'
1009 031066 042701 170000
1010 031072 010005
1011 031074 004737 017106'
1012 031100 013702 003124'
1013 031104 062702 000020
1014 031110 060102
1015 031112 042702 000003
1016 031116 013703 003130'
1017 031122 162703 000020
1018 031126 010237 030034'
1019 031132 010237 030036'
1020 031136 020203
1021 031140 101007
1022 031142 020237 003124'
1023 031146 103007
1024 031150 005737 003134'
1025 031154 001004
1026 031156 000424
1027 031160 162702 000020
1028 031164 000754
1029 031166
1030 031166 005737 003134'
1031 031172 001420

ROUTINE TO CONVERT A TEST PATTERN TO A VALID ADDRESS IN DIAGNOSTIC FREE SPACE
DIAGNOSTIC FREE SPACE IS BETWEEN THE END OF THE DIAGNOSTIC AND THE
BEGINNING OF THE SUPERVISOR. THIS IS ALWAYS BELOW 24K.
IF MEMORY ABOVE 28K SPECIFIED (VIA R1) THEN PAR 6 IS SET
TO THE RELOCATION BASE.

INPUTS:
R0      HIGH ORDER ADDRESS BITS
R1      LOW ORDER ADDRESS BITS

OUPUTS:
T12PAR6 = ADDRESS BIASED TO PAR6 IF >28K UNDER TEST
T12HIADD = HIGH ORDER ADDRESS IN NON PAR6 FORMAT
T12LOADD = LOW ORDER ADDRESS IN NON PAR6 FORMAT
C BIT = 1 IF GOOD ADDRESS RETURNED
C BIT = 0 IF TEST PATTERN DID NOT YIELD A VALID ADDRESS

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

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

1032	031174	005737	003132'	TST	KTF LG	;ANY MEMORY ABOVE 28K?
1033	031200	001413		BEQ	90\$	;BR IF NO
1034	031202	004737	017070'	JSR	PC,KTCN	;TURN ON MEMORY MANAGEMENT
1035	031206	010500		MOV	R5,R0	;GET HIGH ORDER ADDRESS
1036	031210	010037	030032'	MOV	R0,T12HIADD	;SAVE POSSIBLE HIGH ADDRESS
1037	031214	010201		MOV	R2,R1	;GET COMPUTED LOW ORDER ADDRESS
1038	031216	004737	017130'	JSR	PC,SETMAP	;RETURN PAR6 BIASED ADDRESS IN R0
1039	031222	010037	030036'	MOV	R0,T12PAR6	;COPY PAR6 BIASED ADDRESS
1040	031226	103403		BCS	105\$	;BR IF VALID ADDRESS
1041	031230	000241		90\$: CLC		;CLR C BIT FOR FAILURE
1042	031232	000401		BR	105\$	
1043	031234	000261		100\$: SEC		;SET SUCCESS
1044	031236	000207		105\$: RTS	PC	;RETURN
1045						
1046						
1047						
1048						
1049						
1050						
1051						
1052						
1053						
1054						
1055						
1056						
1057						
1058						
1059						
1060						
1061						
1062						
1063						
1064						
1065						
1066						
1067						
1068						
1069						
1070						
1071						
1072						
1073	031240					
1074	031240					
1075	031244	012701	002242'	SAVREG		;SAVE THE GENERAL REGISTERS
1076	031250	012702	000201	MOV	@RAMDATA,R1	;ADDRESS TO SAVE THE RAM DATA
1077	031254	005003		MOV	@RMPKTBEG,R2	;BYTE ADDRESS OF FIRST RAM DATA
1078	031256	004737	016146'	CLR	R3	;CLEAR THE ERROR FLAG
1079	031262	112765	000000 000000	JSR	PC,CHKTSSR	;WAIT FOR SSR
1080	031270	004737	016146'	MOVB	@C,TSD8(R5)	;SET MAINTENANCE MODE
1081	031274	010265	000000	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
1082	031300	004737	016146'	MOV	R2,TSD8(R5)	;SELECT NEXT RAM ADDRESS
1083	031304	116511	000000	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
1084	031310	122124		MOVB	TSBA(R5),(R1)	;READ THE RAM DATA
1085	031312	001401		CMPB	(R1)+,(R4)+	;COMPARE TO EXPECTED
1086	031314	005203		BEQ	20\$	;BRANCH IF OK
1087	031316	005202		INC	R3	;SET ERROR FLAG
1088	031320	020227	000203	20\$: INC	R2	;ADDRESS OF NEXT RAM LOCATION
				CMP	R2,@RMPKTBEG+2	;DONE 2 BYTES?

ROUTINE TO READ THE FIRST 2 BYTES FROM RAM
 MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.

INPUT:

R4 ADDRESS OF THE COMMAND PACKET
 R5 FIRST DEVICE UNIBUS ADDRESS

OUTPUT:

CARRY SET - RAM MATCHES PACKET
 CLR - RAM DOES NOT MATCH PACKET

IMPLICIT OUTPUT:

THE TABLE RAMDATA IS FILLED WITH THE
 DATA HELD IN RAM.
 RAMSIZ SET TO 2 FOR PRAMPKT ROUTINE

SIDE EFFECTS:

THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE

T12CKRAM::

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

SEQ 111

```

1089 031324 002761          BLT      10$          ;BR IF NO
1090 031326 005703          TST      R3           ;WAS AN ERROR FOUND ?
1091 031330 001402          BEQ      30$          ;BRANCH IF NOT
1092 031332 000241          CLC              ;CLEAR CARRY TO SHOW ERROR
1093 031334 000401          BR       50$          ;AND EXIT
1094 031336 000261          SEC              ;SHOW GOOD COMPARE
1095 031340 012737 000002 002302' 50$: MOV     $2,RAMSIZ ;SETUP RAMSIZ
1096 031346 000207          RTS      PC          ;RETURN
1097
1098          ;+
1099          ;
1100          ;ROUTINE TO SETUP PACKET TO WRITE CHARACTERISTICS
1101          ;-
1102
1103 031350          T12SWRT:
1104 031350          SAVREG          ;SAVE THE REGISTERS
1105 031354 012701 027770'      MOV     $T12PACKET,R1 ;START OF THE PACKET
1106 031360 012721 100004'      MOV     $100004,(R1)+ ;WRITE CHARACTERISTICS WITH ACK
1107 031364 012721 030000'      MOV     $T12DATA,(R1)+ ;ADDRESS OF CHAR DATA BLOCK
1108 031370 005021          CLR      (R1)+         ;EXTENDED ADDRESS
1109 031372 012721 000010'      MOV     $8,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
1110 031376 012721 030012'      MOV     $T12BFR,(R1)+ ;ADDRESS OF MESSAGE BUFFER
1111 031402 005021          CLR      (R1)+
1112 031404 012721 000016'      MOV     $14,(R1)+      ;LENGTH OF MESSAGE BUFFER
1113 031410 005021          CLR      (R1)+
1114 031412 005011          CLR      (R1)
1115 031414 000207          RTS      PC          ;RETURN
1116          ;+
1117          ;
1118          ;ROUTINE TO SETUP A GET STATUS COMMAND PACKET AT CURRENT PACKET ADDRESS
1119          ;
1120          ;      R3      HIGH ORDER PACKET ADDRESS
1121          ;      R4      LOW ORDER PACKET ADDRESS
1122          ;      NOTE: R3 IS IGNORED IF KTENABLE FLAG CLEAR
1123          ;
1124          ;-
1125
1126 031416          T12SETGET:
1127 031416          SAVREG          ;SAVE THE REGISTERS
1128 031422 010401          MOV     R4,R1           ;GET LOW ORDER ADDRESS
1129 031424 005737 003134'      TST      KTENABLE      ;TESTING ABOVE 28K?
1130 031430 001404          BEQ      10$          ;BR IF NO
1131 031432 010300          MOV     R3,R0           ;GET HIGH ORDER ADDRESS
1132 031434 004737 017130'      JSR     PC,SETMAP      ;RETURN ADDRESS BIASED TO PAR6 IN R0
1133 031440 010001          MOV     R0,R1           ;GET ADDRESS
1134 031442 012700 000017' 10$: MOV     $P.GETSTATUS,R0 ;GET STATUS COMMAND CODE NO IE
1135 031446 052700 100000'      BIS     $P.ACK,R0      ;SET ACK
1136 031452 010021          MOV     R0,(R1)+        ;STORE GET STATUS IN PACKET
1137 031454 005021          CLR      (R1)+        ;CLEAR UNUSED WORD
1138 031456 000207          RTS      PC          ;RETURN
1139
1140          ;+
1141          ;
1142          ;ROUTINE TO SETUP A CHARACTERISTIC DATA BLOCK AT A TEST ADDRESS
1143          ;
1144          ;
1145          ;-

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

SEQ 112

```

1146 031460          T12CHAR:
1147 031460          SAVREG
1148 031464 012700 030000'  MOV    #T12DATA,R0      ;SAVE R1-R5 UNTIL NEXT RETURN
1149 031470 013701 030034'  MOV    T12LOAD,R1      ;GET T12PACKET DATA POINTER
1150 031474 005737 003134'  TST     KTENABLE      ;ASSUME NOT ABOVE 28K
1151 031500 001402          BEQ     10$      ;TESTING ABOVE 28K?
1152 031502 013701 030036'  MOV    T12PAR6,R1      ;BR IF NO
1153 031506 012021          10$: MOV    (R0)+,(R1)+      ;SET TEST ADDRESS ABOVE 28K
1154 031510 012021          MOV    (R0)+,(R1)+      ;STORE DATA WORD 1
1155 031512 012021          MOV    (R0)+,(R1)+      ;STORE DATA WORD 2
1156 031514 012021          MOV    (R0)+,(R1)+      ;STORE DATA WORD 3
1157 031516 012021          MOV    (R0)+,(R1)+      ;STORE DATA WORD 4
1158 031520 000207          MOV    (R0)+,(R1)+      ;STORE DATA WORD 5
1159                      RTS     PC      ;RETURN
1160 031522          ENDTST
1161                      L10042:
1162                      031522          TRAP    C$ETST
1163                      031522 104401
1164                      .SBTTL TEST 4: RAM EXERCISER TEST
1165                      ;
1166                      ;THIS TEST USES THE READ AND WRITE RAM (BOTH SINGLE AND 256
1167                      ;LOCATIONS) SELECT CODES OF THE WRITE SUBSYSTEM MEMORY COMMAND
1168                      ;TO EXERCISE THE CONTROLLER'S RAM MEMORY AND DMA LOGIC
1169                      ;
1170 031524          BGNTST
1171                      T4::
1172                      031524
1177 031524 005737 002214'  TST     TSTCNT      ;CHECK FOR RUN MODE
1178 031530 001402          BEQ     10$      ;BR, IF NOT ONLY PROGRAM RUN
1179 031532 005237 003400'  INC     SKIPT      ;SET SKIP SW
1180 031536 012700 034163'  10$: MOV    #TST15ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
1181 031542 004737 016322'  JSR     PC,TSTSETUP      ;DO INITIAL TEST SETUP
1182 031546 012737 000005 002216'  MOV    #5,LOOPCNT      ;PERFORM 5 ITERATIONS
1183 031554          T15LOOP:
1184                      ;
1185                      ;
1186                      ;TEST 4, SUBTEST 1
1187                      ;
1188                      ;THIS SUBTEST WRITES THE ADDRESS (8 BITS) INTO THE
1189                      ;RAM MEMORY SINGLE WORD (8 BITS) MODE
1190                      ;
1191                      ;
1192 031554          BGNSUB      ;//////////////// BEGIN SUBTEST //////////////////
1193 031554          031554          T4.1:
1194 031554 104402          TRAP    C$BSUB
1195 031556          SETPRI    #PRI00      ;LOWER PRIORITY TO ALLOW INTERRUPTS
1196 031556 012700 000000'  MOV    #PRI00,R0
1197 031562 104401          TRAP    C$SPRI
1198 031564 005737 003400'  TST     SKIPT      ;SHOULD WE SKIP THIS SUBTEST
1199 031570 001402          BEQ     10$      ;BR, IF NOW SKIP REQUIRED
1200 031572 000137 032054'  JMP     50$      ;SKIP SUBTEST
1201 031576 004737 034202'  10$: JSR     PC,T15REST      ;SET COMMAND PACKET
1202 031602 004737 034254'  JSR     PC,T15R72      ;SET UP OTHER COMMAND PACKET
1203 031606 004737 015504'  JSR     PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 4: RAM EXERCISER TEST

SEQ 113

1200	031612	103405		BCS	20:		;BR IF INIT WAS OK	
1204	031614	010001		MOV		R0,R1	;CONTENTS OF TSSR REGISTER	
1205	031616			ERRDF		ERRNO,SFIERR,SFIMSG	;FATAL ERROR TSSR WAS NOT OK	
	031616	104455					TRAP	C\$ERDF
	031620	000621					.WORD	401
	031622	003646'					.WORD	SFIERR
	031624	011644'					.WORD	SFIMSG
1206	031626		20:					
1207	031626	012704	033100'	MOV		#T15PACKET,R4	;SUBROUTINE NEEDS PACKET ADDRESS	
1208	031632	004737	010472'	JSR		PC,WRTCHR	;ISSUE WRITE CHARACTERISTICS	
1209	031636	103405		BCS		23:	;BR, IF COMMAND ISSUED OK	
1213	031640	010001		MOV		R0,R1	;SAVE CONTENTS OF TSSR	
1214	031642			ERRHRD		ERRNO,WRTMSG,SFIMSG	;WRITE CHARACTERISTICS FAILED	
	031642	104456					TRAP	C\$ERHRD
	031644	000622					.WORD	402
	031646	005052'					.WORD	WRTMSG
	031650	011644'					.WORD	SFIMSG
1215	031652	012703	000400	23:	MOV	#256,R3	;STARTING ADDRESS FOR RAM WRITE	
1216	031656	112737	000001 033611'	MOVB		#1,T15BS1	;SIZE OF TRANSFER	
1217	031664	112737	000002 033610'	MOVB		#2,T15BS0	;WRITE RAM "COMMAND"	
1218	031672			25:				
1219	031672	010337	033612'	MOV		R3,T15S2	;ADDRESS FOR RAM	
1220	031676	012704	033600'	MOV		#T15PK2,R4	;WRITE SUBSYS MEM PACKET	
1221	031702	110337	033614'	MOVB		R3,T15S3	;DATA FOR WRITE (ADDRESS)	
1222	031706	010465	000000	MOV		R4,TSD8(R5)	;ISSUE COMMAND	
1223	031712	004737	016146'	JSR		PC,CHKTSSR	;WAIT FOR SSR	
1224	031716	103407		BCS		30:	;BR, IF NO ERROR	
1225	031720	010001		MOV		R0,R1	;ERROR, SAVE TSSR	
1229	031722			ERRHRD		ERRNO,T15SSR,PKTSSR	;TSSR NOT CORRECT AFTER WRITE SUB MEM	
	031722	104456					TRAP	C\$ERHRD
	031724	000623					.WORD	403
	031726	033616'					.WORD	T15SSR
	031730	011656'					.WORD	PKTSSR
1230	031732			ESCAPE		SUB	;DON'T CONTINUE IF ERROR ON WRITE	
	031732	104410					TRAP	C\$ESCAPE
	031734	000122					.WORD	L10050-
1231	031736			30:	CKLOOP		;SCOPE LOOP	
	031736	104406					TRAP	C\$CLP1
1232								
1233								
1234	031740	005203		INC		R3	;NEXT ADDRESS	
1235	031742	020327	010000	CMP		R3,#10000	;END OF RAM MEMORY CHECK	
1236	031746	001351		BNE		25:	;LOOP TILL ALL RAM WRITTEN	
1237	031750	005002		CLR		R2	;CLEAR OUT R2 HIGH BITS	
1238	031752	005303		DEC		R3	;SET BACK TO 7777	
1239	031754	110337	033614'	40:	MOVB	R3,T15S3	;GET DATA PATTERN BACK IN SHAPE	
1240	031760	010337	033612'	MOV		R3,T15S2	;ADDRESS FOR RAM READ	
1241	031764	112737	000001 033610'	MOVB		#1,T15BS0	;READ RAM COMMAND	
1242	031772	010465	000000	MOV		R4,TSD8(R5)	;SEND OUT PACKET ADDRESS TO CONTR.	
1243	031776	004737	016146'	JSR		PC,CHKTSSR	;WAIT FOR READY, NON-AMBIGUOUS	
1244	032002	103405		BCS		43:	;BR, IF NO PROBLEM	
1245	032004	010001		MOV		R0,R1	;SAVE TSSR	
1249	032006			ERRDF		ERRNO,T15SSR,PKTSSR	;TSSR NOT CORRECT	
	032006	104455					TRAP	C\$ERDF
	032010	000624					.WORD	404
	032012	033616'					.WORD	T15SSR
	032014	011656'					.WORD	PKTSSR

TSSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 4: RAM EXERCISER TEST

SEQ 114

```

1250 032016          43$: CKLOOP          ;SCOPE LOOP
      032016 104406          ;GET RAM READ DATA          TRAP C$CLP1
1251 032020 013701 033142'      MOV T15BFR+20,R1      ;SET UP FOR COMPARE
1252 032024 010302          MOV R3,R2      ;CHECK WITH DATA WRITTEN
1253 032026 120102          CMPB R1,R2      ;BR IF OK, DATA IN = DATA OUT
1254 032030 001404          BEQ 45$      ;WRITTEN DATA NOT = TO READ
1258 032032          ERRHRD ERRNO,T15AM4,EXPBREC      TRAP C$ERHRD
      032032 104456          .WORD 405
      032034 000625          .WORD T15AM4
      032036 034075'          .WORD EXPBREC
      032040 015312'
1259 032042          45$: CKLOOP          ;SCOPE LOOP          TRAP C$CLP1
      032042 104406          ;DROP DATA COUNTER (PATTERN)
1260 032044 005303          DEC R3      ;AT BOTTOM YET
1261 032046 020327 000377      CMP R3,#255.      ;BR, IF MORE TO CHECK
1262 032052 001340          BNE 40$      ;SCOPE LOOP
1263 032054          50$: CKLOOP          ;SCOPE LOOP          TRAP C$CLP1
      032054 104406          ;///////////////// END SUBTEST ///////////////////
1264 032056          ENDSUB          L10050:          TRAP C$ESUB
      032056          ;
      032056 104403          ;
1265          ;
1266          ;
1267 032060          BCNSUB          ;///////////////// BEGIN SUBTEST ///////////////////
      032060          ;
      032060 104402          ;T4.2:          TRAP C$BSUB
1268          ;
1269          ;
1270          ;
1271          ;TEST 4, SUBTEST 2
1272          ;
1273          ;
1274          ;
1275          ;
1276          ;
1277 032062 004737 034202'      JSR PC,T15REST      ;RESTORE PACKET FOR WRITE CHARA
1278 032066 004737 034254'      JSR PC,T15RT2      ;RESTORE PACKET FOR WRT SUB SYS MEM
1279 032072 004737 015604'      JSR PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
1280 032076 103405          BCS 20$      ;BR IF INIT WAS OK
1284 032100 010001          MOV R0,R1      ;CONTENTS OF TSSR REGISTER
1285 032102          ERRDF ERRNO,SFIERR,SFIMSG      ;FATAL ERROR TSSR WAS NOT OK
      032102 104455          TRAP C$ERDF
      032104 000626          .WORD 406
      032106 003646'          .WORD SFIERR
      032110 011644'          .WORD SFIMSG
1286 032112          20$:
1287 032112 012704 033100'      MOV #T15PACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
1288 032116 004737 010472'      JSR PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
1289 032122 103405          BCS 25$      ;BR, IF COMMAND ASSURED OK
1293 032124 010001          MOV R0,R1      ;SAVE CONTENTS OF TSSR
1294 032126          ERRHRD ERRNO,WRTMSG,SFIMSG      ;WRITE CHARACTERISTICS FAILED
      032126 104456          TRAP C$ERNRD
      032130 000627          .WORD 407
      032132 005052'          .WORD WRTMSG
      032134 011644'          .WORD SFIMSG
1295 032136          25$:
1296 032136 112737 000001 033611'      MOVB #1,T15BS1      ;SET SIZE OF TRANSFER 1 BYTE

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 4: RAM EXERCISER TEST

SEQ 115

1297	032144	012704	033600'		MOV	#T15PK2,R4	;SET NEW PACKET ADDRESS		
1298	032150	012703	000400		MOV	#256.,R3	;STARTING ADDRESS IN RAM		
1299	032154	112737	000002	033610'	MOVB	#2,T15B50	;WRITE RAM COMMAND		
1300	032162	105037	033614'		CLRB	T15S3	;SET DATA TO 000		
1301	032166	010337	033612'	30\$:	MOV	R3,T15S2	;ADDRESS TO PACKET DATA AREA		
1302	032172	010465	000000		MOV	R4,TSDB(R5)	;SEND OUT PACKET ADDRESS		
1303	032176	004737	016146'		JSR	PC,CHKTSSR	;WAIT FOR SSR		
1304	032202	103405			BCS	33\$	;BR, IF NO PROBLEM		
1305	032204	010001			MOV	R0,R1	;SAVE TSSR		
1309	032206				ERRHRD	ERRNO,T15SSR,PKTSSR	;TSSR NOT CORRECT	TRAP	C\$ERHRD
	032206	104456						.WORD	408
	032210	000630						.WORD	T15SSR
	032212	033616'						.WORD	PKTSSR
	032214	011656'							
1310	032216			33\$:	CKLOOP		;SCOPE LOOP	TRAP	C\$CLP1
	032216	104406							
1311									
1312									
1313	032220	005203			INC	R3	;NEXT ADDRESS		
1314	032222	020327	010000		CMP	R3,#10000	;END OF RAM MEMORY CHECK		
1315	032226	001357			BNE	30\$	;BR, MORE RAM TO GO		
1316	032230	005303		35\$:	DEC	R3	;SET BACK TO 7777		
1317	032232	005002		40\$:	CLR	R2	;SET TO ALL ZEROS		
1318	032234	112737	000001	033610'	MOVB	#1,T15B50	;READ RAM COMMAND		
1319	032242	010337	033612'		MOV	R3,T15S2	;ADDRESS TO BE READ TO PACKET DATA		
1320	032246	010465	000000		MOV	R4,TSDB(R5)	;SEND OUT PACKET ADDRESS		
1321	032252	004737	016146'		JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET		
1322	032256	103405			BCS	41\$	;BR, IF ALL IS WELL		
1323	032260	010001			MOV	R0,R1	;SAVE TSSR		
1327	032262				ERRHRD	ERRNO,T15SSR,PKTSSR	;TSSR NOT CORRECT	TRAP	C\$ERHRD
	032262	104456						.WORD	409
	032264	000631						.WORD	T15SSR
	032266	033616'						.WORD	PKTSSR
	032270	011656'							
1328	032272			41\$:	CKLOOP		;SCOPE LOOP	TRAP	C\$CLP1
	032272	104406							
1329	032274	013701	033142'		MOV	T15BFR+20,R1	;PICK UP READ DATA		
1330	032300	120102			CMPB	R1,R2	;BOTH SHOULD BE 00000000 BINARY		
1331	032302	001404			BEQ	42\$	;BR, IF DATA IS GOOD		
1335	032304				ERRHRD	ERRNO,T15AM3,EXPBREC	;CHARACTERISTICS DATA NOT CORRECT	TRAP	C\$ERHRD
	032304	104456						.WORD	410
	032306	000632						.WORD	T15AM3
	032310	033773'						.WORD	EXPBREC
	032312	015312'							
1336	032314			42\$:	CKLOOP		;SCOPE LOOPER	TRAP	C\$CLP1
	032314	104406							
1337	032316	012702	000377		MOV	#000377,R2	;SET ALL ONES WORD		
1338	032322	112737	000002	033610'	MOVB	#2,T15B50	;WRITE RAM COMMAND		
1339	032330	112737	000377	033614'	MOVB	#000377,T15S3	;ALL ONES PATTERN		
1340	032336	010465	000000		MOV	R4,TSDB(R5)	;PASS PACKET ADDRESS TO CONTR.		
1341	032342	004737	016146'		JSR	PC,CHKTSSR	;WAIT FOR SSR		
1342	032346	103405			BCS	43\$	;BR, IF OK (NO ERROR)		
1343	032350	010001			MOV	R0,R1	;SAVE TSSR		
1347	032352				ERRHRD	ERRNO,T15SSR,PKTSSR	;TSSR NOT CORRECT	TRAP	C\$ERHRD
	032352	104456						.WORD	411
	032354	000633						.WORD	T15SSR
	032356	033616'							

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 4: RAM EXERCISER TEST

SEQ 116

```

134 032360 011656'
134 032362 104406 43$: CKLOOP ;SCOPE LOOP .WORD PKTSSR
1349 032364 112737 000001 033610' MOVB #1,T15BS0 ;SET UP FOR RAM READ TRAP C$CLP1
1350 032372 010465 000000 MOV R4,TSD8(R5) ;ISSUE RAM READ
1351 032376 004737 016146' JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
1352 032402 103405 BCS 44$ ;BR, IF OK (NO ERROR)
1353 032404 010001 MOV R0,R1 ;SAVE TSSR
1357 032406 ERRDF ERRNO,T15SSR,PKTSSR ;TSSR NOT CORRECT
032406 104455 TRAP C$ERDF
032410 000634 .WORD 412
032412 033616' .WORD T15SSR
032414 011656' .WORD PKTSSR
1358 032416 013701 033142' 44$: MOV T15BFR+20,R1 ;PICK UP REC'D DATA
1359 032422 120102 CMPB R1,R2 ;CHECK WITH DATA WRITTEN
1360 032424 001404 BEQ 45$ ;BR IF OK, DATA IN = DATA OUT
1364 032426 ERRHRD ERRNO,T15AM2,EXPBREC ;WRITTEN DATA NOT = TO READ
032426 104456 TRAP C$ERHRD
032430 000635 .WORD 413
032432 033672' .WORD T15AM2
032434 015312' .WORD EXPBREC
1365 032436 45$: CKLOOP ;SCOPE LOOP TRAP C$CLP1
032436 104406 ;DROP RAM ADDRESS POINTER
1366 032440 005303 DEC R3 ;AT START YET
1367 032442 020327 000377 CMP R3,#255. ;BR, IF MORE RAM TO CHECK
1368 032446 001271 BNE 40$
1369 ENDSUB ;////////////////// END SUBTEST ////////////////////
1370 032450 L10051: TRAP C$ESUB
032450 104403 ;////////////////// BEGIN SUBTEST ////////////////////
1371 032452 BGNSUB T4.3: TRAP C$BSUB
032452 104402
1373 ;
1374 ;
1375 ;TEST 4, SUBTEST 3
1376 ;
1377 ;
1378 ; THIS SUBTEST WRITES RAM WITH ALL ONES
1379 ; THEN WALKS AN ALL ZEROS WORD DOWN THROUGH MEMORY
1380 ;
1381 032454 005737 003400' TST SKIPT ;CHECK RUN MODE
1382 032460 001402 BEQ 10$ ;BR, IF NO SKIP
1383 032462 000137 033056' JMP 50$ ;SKIP SUBTEST
1384 032466 004737 034202' 10$: JSR PC,T15REST ;RESTORE PACKET FOR WRITE CHARA
1385 032472 004737 034254' JSR PC,T15RT2 ;RESTORE PACKET FOR WRT SUB SYS MEM
1386 032476 004737 015604' JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
1387 032502 103405 BCS 20$ ;BR IF INIT WAS OK
1391 032504 010001 MOV R0,R1 ;CONTENTS OF TSSR REGISTER
1392 032506 ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
032506 104455 TRAP C$ERDF
032510 000636 .WORD 414
032512 003646' .WORD SFIERR
032514 011644' .WORD SFIMSG
1393 032516 20$: MOV #T15PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
1394 032516 012704 033100'

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 4: RAM EXERCISER TEST

SEQ 117

1395	032522	004737	010472'		JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS	
1396	032526	103405			BCS	25\$		;BR, IF COMMAND ISSUED OK	
1400	032530	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR	
1401	032532				ERRHRD	ERRNO,WRTMSG,SFIMSL		;WRITE CHARACTERISTICS FAILED	
	032532	104456						TRAP	C\$ERHRD
	032534	000637						.WORD	415
	032536	005052'						.WORD	WRTMSG
	032540	011644'						.WORD	SFIMSG
1402	032542			25\$:					
1403	032542	112737	000001	033611'	MOVB	#1,T15BS1		;SET SIZE TO 1 BYTE	
1404	032550	012704	033600'		MOV	#T15PK2,R4		;SET NEW PACKET ADDRESS	
1405	032554	012703	000400		MOV	#256.,R3		;STARTING ADDRESS IN RAM	
1406	032560	112737	000002	033610'	MOVB	#2,T15BS0		;WRITE RAM COMMAND	
1407	032566	112737	000377	033614'	MOVB	#377,T15S3		;SET DATA TO 377	
1408	032574	010337	033612'		MOV	R3,T15S2		;ADDRESS TO PACKET DATA AREA	
1409	032600	010465	000000	30\$:	MOV	R4,TSD8(R5)		;SEND OUT PACKET ADDRESS	
1410	032604	004737	016146'		JSR	PC,CHKTSSR		;WAIT FOR SSR	
1411	032610	103405			BCS	33\$		;BR, IF NO PROBLEM	
1412	032612	010001			MOV	R0,R1		;SAVE TSSR	
1416	032614				ERRHRD	ERRNO,T15SSR,PKTSSR		;TSSR NOT CORRECT	
	032614	104456						TRAP	C\$ERHRD
	032616	000640						.WORD	416
	032620	033616'						.WORD	T15SSR
	032622	011656'						.WORD	PKTSSR
1417	032624			33\$:	CKLOOP			;SCOPE LOOP	
	032624	104406						TRAP	C\$CLP1
1418									
1419									
1420	032626	005203			INC	R3		;NEXT ADDRESS	
1421	032630	020327	010000		CMP	R3,#10000		;END OF RAM MEMORY CHECK	
1422	032634	001357			BNE	30\$		;BR, MORE RAM TO GO	
1423	032636	005303		35\$:	DEC	R3		;SET BACK TO 7777	
1424	032640	112702	000377	40\$:	MOVB	#377,R2		;SET TO ALL ONES	
1425	032644	112737	000001	033610'	MOVB	#1,T15BS0		;READ RAM COMMAND	
1426	032652	010337	033612'		MOV	R3,T15S2		;ADDRESS TO BE READ TO PACKET DATA	
1427	032656	0104 5	000000		MOV	R4,TSD8(R5)		;SEND OUT PACKET ADDRESS	
1428	032662	004737	016146'		JSR	PC,CHKTSSR		;WAIT FOR SSR TO SET	
1429	032666	103405			BCS	41\$		;BR, IF ALL IS WELL	
1430	032670	010001			MOV	R0,R1		;SAVE TSSR	
1434	032672				ERRHRD	ERRNO,T15SSR,PKTSSR		;TSSR NOT CORRECT	
	032672	104456						TRAP	C\$ERHRD
	032674	000641						.WORD	417
	032676	033616'						.WORD	T15SSR
	032700	011656'						.WORD	PKTSSR
1435	032702			41\$:	CKLOOP			;SCOPE LOOP	
	032702	104406						TRAP	C\$CLP1
1436	032704	013701	033142'		MOV	T15BFR+20,R1		;PICK UP READ DATA	
1437	032710	120102			CMPB	R1,R2		;BOTH SHOULD BE 11111111 BINARY	
1438	032712	001404			BEQ	42\$		;BR, IF DATA IS GOOD	
1442	032714				ERRHRD	ERRNO,T15AM3,EXPBREC		;CHARACTERISTICS DATA NOT CORRECT	
	032714	104456						TRAP	C\$ERHRD
	032716	000642						.WORD	418
	032720	033773'						.WORD	T15AM3
	032722	015312'						.WORD	EXPBREC
1443	032724	012702	000377	42\$:	MOV	#000377,R2		;SET ALL ONES WORD	
1444	032730	012737	000002	033610'	MOV	#2,T15BS0		;WRITE RAM COMMAND	
1445	032736	112737	000377	033614'	MOVB	#000377,T15S3		;ALL ONES PATTERN	

B10

TSVS - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 4: RAM EXERCISER TEST

SEQ 118

1446	032744	010465	000000		MOV	R4,TSDB(R5)		;PASS PACKET ADDRESS TO CONTR.	
1447	032750	004737	016146'		JSR	PC,CHKTSSR		;WAIT FOR SSR	
1448	032754	103405			BCS	43:		;BR, IF OK (NO ERROR)	
1449	032756	010001			MOV	RO,R1		;SAVE TSSR	
1453	032760				ERRHRD	ERRNO,T15SSR,PKTSSR		;TSSR NOT CORRECT	
	032760	104456						TRAP	C:ERHRD
	032762	000643						.WORD	419
	032764	033616'						.WORD	T15SSR
	032766	011656'						.WORD	PKTSSR
1454	032770			43:	CKLOOP			;SCOPE LOOP	
	032770	104406						TRAP	C:CLP1
1455	032772	112737	000001	033610'	MOVB	#1,T15B50		;SET UP FOR RAM READ	
1456	033000	010465	000000		MOV	R4,TSDB(R5)		;ISSUE RAM READ	
1457	033004	004737	016146'		JSR	PC,CHKTSSR		;WAIT FOR SSR TO SET	
1458	033010	103405			BCS	44:		;BR, IF OK (NO ERROR)	
1459	033012	010001			MOV	RO,R1		;SAVE TSSR	
1463	033014				ERRHRD	ERRNO,T15SSR,PKTSSR		;TSSR NOT CORRECT	
	033014	104456						TRAP	C:ERHRD
	033016	000644						.WORD	420
	033020	033616'						.WORD	T15SSR
	033022	011656'						.WORD	PKTSSR
1464	033024	013701	033142'	44:	MOV	T15BFR+20,R1		;PICK UP REC'D DATA	
1465	033030	120102			CMPB	R1,R2		;CHECK WITH DATA WRITTEN	
1466	033032	001404			BEQ	45:		;BR IF OK, DATA IN = DATA OUT	
1470	033034				ERRHRD	ERRNO,T15AM2,EXPBREC		;WRITTEN DATA NOT = TO READ	
	033034	104456						TRAP	C:ERHRD
	033036	000645						.WORD	421
	033040	033672'						.WORD	T15AM2
	033042	015312'						.WORD	EXPBREC
1471	033044			45:	CKLOOP			;SCOPE LOOP	
	033044	104406						TRAP	C:CLP1
1472	033046	005303			DEC	R3		;DROP RAM ADDRESS POINTER	
1473	033050	020327	000377		CIP	R3,#255.		;AT START YET	
1474	033054	001271			BNE	40:		;BR, IF MORE RAM TO CHECK	
1475									
1476	033056			50:					
1477	033056				ENDSUB			////////// END SUBTEST //////////	
	033056	104403						L10052:	
1478								TRAP	C:ESUB
1479									
1480	033060	004737	016270'		JSR	PC,TSTLOOP		;DO WE NEED TO ITERATE TEST ?	
1481	033064	103002			BCC	63:		;BRANCH IF NOT	
1482	033066	000137	031554'		JMP	T15LOOP		;EXECUTE AGAIN	
1483	033072			63:	EXIT	TST		;ALL DONE THIS TEST	
	033072	104432						TRAP	C:EXIT
	033074	001216						.WORD	L10047-
1484									
1485									
1486									
1487									
1489	033076				.BLKB	10-<.-TSV267>			
1491	033100				T15PACKET:			;COMMAND PACKET FOR TEST	
1492	033100	100204			.WORD	100204		;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK	
1493	033102	033110'			.WORD	T15DATA		;ADDRESS OF CHARACTERISTICS BLOCK	
1494	033104	000000			.WORD	0			
1495	033106	000010			.WORD	8.		;STARTING VALUE OF BLOCK SIZE	

C10

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 4: RAM EXERCISER TEST

SEQ 119

```

1496 033110
1497 033110 033122'
1498 033112 000000
1499 033114 000400
1500 033116 000000 000000
1501 033122
1502
1503
1504
1506 033576
1508 033600
1509 033600 100206
1510 033602 033610'
1511 033604 000000
1512 033606 000006
1513
1514
1515 033610
1516 033610 000
1517 033611 000
1518 033612 000000
1519 033614 000000
1520
1521
1522
1523
1524
1525
1526 033616 127 122 111
1527 033672 127 122 111
1528 033773 127 122 111
1529 034075 127 122 111
1530 034163 122 101 115
1531
1532
1533
1534
1535
1536
1537
1538
1539 034202
1540 034202
1541 034206 012701 033100'
1542 034212 012721 100204
1543 034216 012721 033110'
1544 034222 005021
1545 034224 012721 000010
1546 034230 012721 033122'
1547 034234 005021
1548 034236 012721 000400
1549 034242 005021
1550 034244 005011
1551 034246 005037 033122'
1552 034252 000207
1553
1554

```

```

T15DATA:
    .WORD T15BFR
    .WORD 0
    .WORD 256.
    .WORD 0,0
T15BFR: .BLKW 150.
;
;WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
    .BLKB 10-<.-TSV2&7>
T15PK2:
    .WORD 100206
    .WORD T15BF2
    .WORD 0
    .WORD 6.
;
    .EVEN
T15BF2:
T15BS0: .BYTE 0
T15BS1: .BYTE 0
T15S2: .WORD 0
T15S3: .WORD 0
;BSELO AREA
;BSEL1 AREA
;SEL 2 AREA
;DATA AREA

;
;LOCAL TEXT MESSAGES FOR TEST
;
111 T15SSR: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted'
111 T15AM2: .ASCIZ 'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Ones Word Read Back'
111 T15AM3: .ASCIZ 'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Zeros Word Read Back'
111 T15AM4: .ASCIZ 'WRITE SUBSYSTEM MEMORY COMMAND Failed On Address Test'
115 TST15ID: .ASCIZ 'RAM Exerciser'
;
;
;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
;WRITE SUBSYSTEM MEMORY COMMAND
;
T15REST:
    SAVREG
    MOV #T15PACKET,R1
    MOV #100204,(R1)+
    MOV #T15DATA,(R1)+
    CLR (R1)+
    MOV #8.,(R1)+
    MOV #T15BFR,(R1)+
    CLR (R1)+
    MOV #256.,(R1)+
    CLR (R1)+
    CLR (R1)
    CLR T15BFR
    RTS PC
;SAVE THE REGISTERS
;START OF THE PACKET
;WRITE SUBSYSTEM MEM. WITH ACK, IF
;ADDRESS OF CHARACTERISTICS DATA BLOCK
;EXTENDED ADDRESS
;SIZE OF DATA BLOCK IN BYTES
;ADDRESS OF MESSAGE BUFFER
;LENGTH OF MESSAGE BUFFER
;CLEAR 1ST LOC IN MESSAGE BUFFER
;RETURN

```

D10

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 4: RAM EXERCISER TEST

SEQ 120

```

1555 034254
1556 034254
1557 034260 012701 033600'
1558 034264 012721 100206
1559 034270 012721 033610'
1560 034274 005021
1561 034276 012721 000006
1562 034302 005021
1563 034304 005021
1564 034306 005011
1565 034310 000207
1566 034312
034312
034312 104401
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586 034314
034314
1591 034314 012700 036372'
1592 034320 004737 016322'
1593 034324 012737 000012 002216'
1594 034332
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612

T15RT2:
    SAVREG
    MOV     #T15PK2,R1
    MOV     #100206,(R1)+
    MOV     #T15BF2,(R1)+
    CLR     (R1)+
    MOV     #6,(R1)+
    CLR     (R1)+
    CLR     (R1)+
    CLR     (R1)
    RTS     PC
    ENDTST

;SAVE THE REGISTERS
;START OF THE PACKET
;WRITE SUBSYSTEM MEM. WITH ACK, IE
;ADDRESS OF DATA BLOCK
;EXTENDED ADDRESS
;SIZE OF DATA BLOCK IN BYTES

;RETURN

L10047: TRAP C#ETST

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

BGN1ST
T5::
    MOV     #TST16ID,R0
    JSR     PC,TSTSETUP
    MOV     #10,LOOPCNT
    ;ASCII MESSAGE TO IDENTIFY TEST
    ;DO INITIAL TEST SETUP
    ;PERFORM 10 ITERATIONS

T16LOOP:
.SBTTL TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST
;--
; TEST 5: SUBTEST 1:
; SUBTEST DESCRIPTION:
;
; This subtest verifies that the Invert Sense of Extended features
; Switch function (Write Subsystem Memory,Write Misc command)
; operates properly.
; First the state of the Extended Features switch is read in the
; message packet supplied by the write characteristics command.
; Then, the sense of the switch is logically inverted.
; A Write characteristics command is executed and it is verified
; that the Extended status register (XST4) is returned when
; in Extended mode, and not returned if not in extended mode.
; The subtest also verifies that specifying a Message Buffer

```

```

1613      address with any of bits 21:19 ,set will cause the command to
1614      be rejected.
1615
1616      TEST STEPS:
1617
1618
1619      BEGIN
1620      Write to TSSR register to soft initialize the controller
1621      Do WRITE CHARACTERISTICS to check for Extended Features Switch
1622      IF Extended Features Hardware Switch CLEAR
1623      THEN
1624      (* Verify Extended Features switch can be Inverted to SET *)
1625      Do Write Subsystem Write Miscellaneous to SET Extended Features.
1626      DO a WRITE CHARACTERISTICS with an extended characteristic word
1627      Compare the controller ram to the extended characteristic word
1628      If Data word in controller ram NOT= to word sent Then Print Error
1629      If Message Buffer Data Length NOT= 12. Then Print Error
1630      ELSE
1631      (* Verify Extended Features switch can be Inverted to CLEAR *)
1632      Do Write Subsystem Write Miscellaneous to CLEAR Extended Features.
1633      Do a WRITE CHARACTERISTICS without an extended characteristic word
1634      If Message Buffer Data Length NOT= 10. Then Print Error
1635      END-IF
1636      (* Verify Function Reject when Message Buffer 21:19 are non-zero *)
1637      Write to TSSR register to soft initialize the controller
1638      REPEAT FOR MESSAGE BUFFER ADDRESS bits <21:19> FROM 0 TO 7
1639      DO a WRITE CHARACTERISTICS with a message address bit<21:19> non-zero
1640      If TSSR termination code NOT= Function Reject Then Print Error
1641      END-REPEAT
1642      END
1643      --
1644      034332      BGNSUB      /////////////// BEGIN SUBTEST ///////////////
1645      034332      TS.1:      TRAP      C$BSUB
1646      034332      104402
1647
1648      5$:
1649      Write to TSSR register to soft initialize the controller
1650      JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
1651      BCS      10$      ;BR IF SOFT INIT OKAY
1652      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
1653      ERRDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
1654      034344      104455      TRAP      C$ERDF
1655      034346      000764      .WORD      500
1656      034350      003646      .WORD      SFIERR
1657      034352      011644      .WORD      SFIMSG
1658
1659      10$:
1660      Do WRITE CHARACTERISTICS to check for Extended Features Switch
1661      JSR      PC,T16REST      ;RESTORE PACKET DEFAULTS
1662      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
1663      MOV      @T16PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
1664      JSR      PC,WRTCHR      ;DO WRITE CHARACTERISTICS COMMAND
1665      FORCERROR      12$      ;DO FORCE ERROR IF FORCER=1
1666      BCS      15$      ;BR IF CARRY SET (GOOD RETURN)
1667      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
1668      NEXT,ERRNO
1669      12$:
1670      ERRDF      ERRNO,T16SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
1671      TRAP      C$ERDF

```

	034416	000765				.WORD	501
	034420	036442'				.WORD	T16SSR
	034422	011656'				.WORD	PKTSSR
1663	034424	005237	002222'		INC FATFLG	;SET FATAL ERROR FLAG	
1664	034430			15\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	034430	104406				TRAP	C\$CLP1
1665							
1666					If Extended Features Hardware Switch Clear then:		
1667					(* Verify Extended Features switch can be Inverted to SET *)		
1668					REPEAT FOR TEST PATTERNS IN TSTBLK TABLE		
1669	034432	012701	037742'		MOV #T16BFR,R1	;MESSAGE BUFFER ADDRESS	
1670	034436	032761	000200	000012	BIT #X2.EXTF,XST2(R1)	;EXTENDED FEATURES SWITCH CLEAR?	
1671	034444	001402			BEQ 20\$	;BR IF YES	
1672	034446	000137	035016'		JMP 200\$		
1673	034452	012703	002764'	20\$:	MOV #TSTBLK+10.,R3	;START OF TEST DATA	
1674					Do Write Subsystem Write Miscellaneous to SET Extended Features.		
1675							
1676	034456	004737	037700'		JSR PC,T16SEXT	;SETUP PACKET FOR WRITE MISC INVERT	
1677	034462	012704	040000'		MOV #T16PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
1678	034466	010465	000000		MOV R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
1679	034472	004737	016146'		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET	
1680	034476				FORCERROR 32\$	;DO FORCE ERROR IF FORCER=1	
1681	034512	103407			BCS 40\$	;BR IF CARRY SET (GOOD RETURN)	
1682	034514	010001			MOV RO,R1	;SAVE CONTENTS OF TSSR	
1683	034516				NEXT.ERRNO		
1684	034516			32\$:	ERRDF ERRNO,T162SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	
	034516	104455				TRAP	C\$ERDF
	034520	000766				.WORD	502
	034522	036477'				.WORD	T162SSR
	034524	011656'				.WORD	PKTSSR
1685	034526	005237	002222'		INC FATFLG	;SET FATAL ERROR FLAG	
1686	034532			40\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	034532	104406				TRAP	C\$CLP1
1687							
1688					DO a WRITE CHARACTERISTICS with an extended characteristic word		
1689	034534	012737	125252	002312'	MOV #125252,DATA	;SETUP TEST DATA FOR EXTENDED WORD	
1690	034542	012704	037720'		MOV #T16PACKET,R4	;GET THE ADDRESS OF COMMAND PACKET	
1691	034546	012764	000020	000006	MOV #16.,PKBCNT(R4)	;STORE MESSAGE PACKET SIZE	
1692	034554	013737	002312'	037740'	MOV DATA,T16DATA+10	;STORE TEST DATA IN EXTENDED WORD	
1693	034562	004737	010472'		JSR PC,WRTCHR	;DO WRITE CHARACTERISTICS COMMAND	
1694	034566				FORCERROR 42\$	;DO FORCE ERROR IF FORCER=1	
1695	034602	103407			BCS 50\$	;BR IF CARRY SET (GOOD RETURN)	
1696	034604	010001			MOV RO,R1	;SAVE CONTENTS OF TSSR	
1697	034606				NEXT.ERRNO		
1698	034606			42\$:	ERRDF ERRNO,T16SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	
	034606	104455				TRAP	C\$ERDF
	034610	000767				.WORD	503
	034612	036442'				.WORD	T16SSR
	034614	011656'				.WORD	PKTSSR
1699	034616	005237	002222'		INC FATFLG	;SET FATAL ERROR FLAG	
1700	034622			50\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	034622	104406				TRAP	C\$CLP1
1701					If the TSBA Address Register NOT= Expected Then Print Error		
1702	034624	016501	000000		MOV TSBA(R5),R1	;GET TSBA REGISTER CONTENTS	
1703	034630	012702	037742'		MOV #T16BFR,R2	;START OF THE DATA BLFFER	
1704	034634	062702	000020	62\$:	ADD #16.,R2	;EXPECTED CONTENTS OF TSBA	
1705	034640				FORCERROR 72\$,NOTSSR	;DO FORCE ERROR IF FORCER=1	

G10

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

SEQ 123

```

1706 034650 020102      CMP      R1,R2      ;COMPARE EXPECTED TO RECEIVED
1707 034652 001404      BEQ       80$      ;ERROR IF NOT EQUAL
1708 034654      NEXT.ERRNO
1709 034654      72$:  ERRHRD  ERRNO,T16TSBA,EXPREC  ;PRINT THE ERROR & EXPD/RECV
                        TRAP      C$ERHRD
                        .WORD      504
                        .WORD      T16TSBA
                        .WORD      EXPREC
                        034654 104456
                        034656 000770
                        034660 036610'
                        034662 015304'
1710 034664      80$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
1711      ;      Compare the controller ram to the extended characteristic word
1712      ;      If Data word in controller ram NOT= to word sent Then Print Error
1713 034666 012704 037730'      MOV      @T16DATA,R4      ;GET CHARACTERISTIC DATA ADDRESS
1714 034672 004737 011034'      JSR      PC,CKRAM2      ;DOES RAM DATA EQUAL DATA SENT?
1715 034676      FORCERROR 92$      ;SOFORCE ERROR IF FORCER=1
1716 034712 103404      BCS      100$      ;BR IF YES
1717 034714      NEXT.ERRNO
1718 034714      92$:  ERRHRD  ERRNO,PKTRAM,RAMERR  ;REPORT THE RAM ERROR(S)
                        TRAP      C$ERHRD
                        .WORD      505
                        .WORD      PKTRAM
                        .WORD      RAMERR
                        034714 104456
                        034716 000771
                        034720 004741'
                        034722 015320'
1719 034724      100$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
1720      ;      If Message Buffer Data Length NOT= 12. Then Print Error
1721 034726 012702 037742'      MOV      @T16BFR,R2      ;GET MESSAGE BUFFER ADDRESS
1722 034732 016201 000002      MOV      2(R2),R1      ;GET RECV DATA FIELD LENGTH
1723 034736 012702 000014      MOV      @12.,R2      ;GET EXPD DATA FIELD LENGTH
1724 034742      FORCERROR 112$,NOTSSR      ;SOFORCE ERROR IF FORCER=1
1725 034752 020102      CMP      R1,R2      ;COMPARE EXPECTED TO RECEIVED
1726 034754 001404      BEQ      120$      ;ERROR IF NOT EQUAL
1727 034756      NEXT.ERRNO
1728 034756      112$: ERRHRD  ERRNO,T16LEN,EXPREC  ;PRINT THE ERROR & EXPD/RECV
                        TRAP      C$ERHRD
                        .WORD      506
                        .WORD      T16LEN
                        .WORD      EXPREC
                        034756 104456
                        034760 000772
                        034762 036712'
                        034764 015304'
1729 034766      120$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                        TRAP      C$CLP1
1730      034766 104406
1731 034770 004737 015604'      JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
1732 034774 103405      BCS      125$      ;BR IF SOFT INIT OKAY
1733 034776 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
1734 035000      ERRDF  ERRNO,SFIERR,SFIMSG  ;DEVICE FATAL DURING INIT
                        TRAP      C$ERDF
                        .WORD      506
                        .WORD      SFIERR
                        .WORD      SFIMSG
                        035000 104455
                        035002 000772
                        035004 003646'
                        035006 011644'
1735 035010      125$: CKLOOP      ;LOOP IF SELECTED
                        TRAP      C$CLP1
1736 035012 000137 035176'      JMP      300$      ;
1737
1738      ;      (* Verify Extended Features switch can be Inverted to CLEAR *)
1739 035016      200$:
1740      ;      Do Write Subsystem Write Miscellaneous to CLEAR Extended Features.
1741 035016 004737 037700'      JSR      PC,T16SEXT      ;SETUP PACKET FOR WRITE MISC INVERT
1742 035022 012704 040000'      MOV      @T16PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET

```

H10

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

SEQ 124

```

1743 035026 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
1744 035032 004737 016146'    JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
1745 035036                      FORCERROR      232$      ;SSDFORCE ERROR IF FORCER=1
1746 035052 103407          BCS      240$      ;BR IF CARRY SET (GOOD RETURN)
1747 035054 010001          MOV      R0,R1      ;SAVE CONTENTS OF TSSR
1748 035056                      NEXT,ERRNO
1749 035056          232$:      ERRDF      ERRNO,T162SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      507
                                .WORD      T162SSR
                                .WORD      PKTSSR
                                035056 104455
                                035060 000773
                                035062 036477'
                                035064 011656'
1750 035066 005237 002222'    INC      FATFLG      ;SET FATAL ERROR FLAG
1751 035072          240$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                035072 104406
1752
1753          ;      DO a WRITE CHARACTERISTICS without an extended characteristic word
1754 035074 012704 037720'    MOV      #T16PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
1755 035100 012764 000016 000006  MOV      #14.,PKBCNT(R4)      ;STORE MESSAGE PACKET SIZE
1756 035106 004737 010472'    JSR      PC,WRTCHR      ;DO WRITE CHARACTERISTICS COMMAND
1757 035112          FORCERROR      242$      ;SSDFORCE ERROR IF FORCER=1
1758 035126 103407          BCS      250$      ;BR IF CARRY SET (GOOD RETURN)
1759 035130 010001          MOV      R0,R1      ;SAVE CONTENTS OF TSSR
1760 035132          NEXT,ERRNO
1761 035132          242$:      ERRDF      ERRNO,T16SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      508
                                .WORD      T16SSR
                                .WORD      PKTSSR
                                035132 104455
                                035134 000774
                                035136 036442'
                                035140 011656'
1762 035142 005237 002222'    INC      FATFLG      ;SET FATAL ERROR FLAG
1763 035146          250$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                035146 104406
1764          ;      If Message Buffer Data Length NOT= 10. Then Print Error
1765 035150 013701 037744'    MOV      T16BFR+2,R1      ;GET RECV DATA FIELD LENGTH
1766 035154 012702 000012      MOV      #10.,R2      ;GET EXPD DATA FIELD LENGTH
1767 035160 020102          CMP      R1,R2      ;COMPARE EXPECTED TO RECEIVED
1768 035162 001404          BEQ      270$      ;ERROR IF NOT EQUAL
1769 035164          NEXT,ERRNO
1770 035164          262$:      ERRHRD      ERRNO,T16LEN,EXPREC      ;PRINT THE ERROR & EXPD/RECV
                                TRAP      C$ERHRD
                                .WORD      509
                                .WORD      T16LEN
                                .WORD      EXPREC
                                035164 104456
                                035166 000775
                                035170 036712'
                                035172 015304'
1771 035174          270$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                035174 104406
1772
1773          ;      (* Verify Function Reject when Message Buffer 21-19 are non-zero *)
1774          ;      Write to TSSR register to soft initialize the controller
1775          ;
1776 035176          300$:      REPEAT FOR MESSAGE BUFFER ADDRESS bits <21:19> FROM 0 TO 7
1777          ;
1778 035176 012737 000001 002312' 320$:      MOV      #1,DATA      ;START AT BITS<21:19>=001
1779          ;
1780 035204          325$:      DO a WRITE CHARACTERISTICS with a message address bit<21:19> non-zero
1781 035204 012704 037720'    MOV      #T16PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
1782 035210 012764 000016 000006  MOV      #14.,PKBCNT(R4)      ;STORE MESSAGE PACKET SIZE
1783 035216 013700 002312'    MOV      DATA,R0      ;GET TEST DATA
1784          000003      .REPT      3

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

SEQ 125

```

1785                                ASL      R0                                ;SHIFT INTO BITS 21:19
1786                                .ENOR
1787 035230 010037 037732'          MOV      R0,T16DATA+2          ;STORE BUFFER ADDRESS BITS 21:19
1788 035234 010465 000000          MOV      R4,T508(R5)          ;SET THE PACKET ADDRESS TO EXECUTE
1789 035240 004737 016060'          JSR      PC,WAIF              ;WAIT FOR SSR
1790 035244                                FORCERROR 342$          ;DO FORCE ERROR IF FORCER=1
1791 035260 103407                                BCS      350$          ;BR IF CARRY SET (GOOD RETURN)
1792 035262 010001                                MOV      R0,R1          ;SAVE CONTENTS OF TSSR
1793 035264                                NEXT,ERRNO
1794 035264 342$: ERRDF ERRNO,T16SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     510
                                .WORD     T16SSR
                                .WORD     PKTSSR
                                035264 104455
                                035266 000776
                                035270 036442'
                                035272 011656'
1795 035274 005237 002222'          INC      FATFLG          ;SET FATAL ERROR FLAG
1796 035300 350$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                035300 104406
1797
1798                                ; If TSSR termination code NOT= Function Reject Then Print Error
1799 035302 016501 000002          MOV      TSSR(R5),R1          ;GET RECV TSSR
1800 035306 010102          MOV      R1,R2          ;COPY RECV TSSR
1801 035310 042702 000016          BIC      *TERCLS,R2          ;CLEAR TC<2:0> EXPD
1802 035314 052702 000006          BIS      *TSREJ,R2          ;SET EXPD TC<2:0>= FUNCTION REJECT
1803 035320          FORCERROR 352$,NOTSSR          ;DO FORCE ERROR IF FORCER=1
1804 035330 020102          CMP      R1,R2          ;EXPD EQUAL RECV?
1805 035332 001404          BEQ      360$          ;BR IF YES
1806 035334          NEXT,ERRNO
1807 035334 352$: ERRHRD ERRNO,T16REJ,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERHRD
                                .WORD     511
                                .WORD     T16REJ
                                .WORD     PKTSSR
                                035334 104456
                                035336 000777
                                035340 037024'
                                035342 011656'
1808 035344 360$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                035344 104406
1809 035346          FORCEXIT 370$
1810 035356 005237 002312'          INC      DATA          ;GET NEXT TST PATTERN
1811 035362 023727 002312' 000007          CMP      DATA,*7          ;DONE ALL DATA?
1812 035370 101002          BHI      370$          ;BR IF YES
1813 035372 000137 035204'          JMP      325$          ;DO ANOTHER TEST PATTERN
1814                                ;
1815 035376 370$: ENDSUB          ;///////////////// END SUBTEST ///////////////////
1816 035376                                L10054: TRAP      C$ESUB
                                035376 104403
1817
1818 035376 005737 002222'          TST      FATFLG          ;ANY FATAL ERRORS ?
1819 035380 001402          BEQ      460$          ;BRANCH IF NOT
1820 035384 004737 017014'          JSR      PC,CKDROP          ;TRY TO DROP THE UNIT
1821 035384 460$:
1822
1823                                .SBTTL TEST 5: SUBTEST 2: VERIFY TIMERS A,B
1824
1825                                ;**
1826                                ; TEST 5: SUBTEST 2:
1827                                ;
1828                                ; SUBTEST DESCRIPTION:
1829                                ;

```

```

1830      ; This subtest verifies that timers A,B can be reset
1831      ; and that Timer A is twice the frequency of Timer B.
1832      ; Timer A has a period of 25 microseconds and Timer B
1833      ; has a period of 50 microseconds. The timers are
1834      ; checked at 1, 28, 53, and 78 microseconds.
1835      ;
1836      ; TEST STEPS:
1837      ;
1838      ;
1839      ; Write to TSSR register to soft initialize the controller
1840      ; Do WRITE CHARACTERISTICS to setup a Message Buffer
1841      ; (* Verify Timers A,B after RESET TIMER with 0 microsecond delay *)
1842      ; Do a Write Control RESET TIMER with 1 microsecond delay
1843      ; Do a Write Subsystem READ STATUS
1844      ; If Timer A NOT= 0 Then Print Error
1845      ; If Timer B NOT= 0 Then Print Error
1846      ; (* Verify Timers A,B after RESET TIMER with 28 microsecond delay *)
1847      ; Do a Write Control RESET TIMER with 28 microsecond delay
1848      ; If Timer A NOT= 1 Then Print Error
1849      ; If Timer B NOT= 1 Then Print Error
1850      ; Do a Write Control RESET TIMER with 53 microsecond delay
1851      ; If Timer A NOT= 0 Then Print Error
1852      ; If Timer B NOT= 1 Then Print Error
1853      ; Do a Write Control RESET TIMER with 78 microsecond delay
1854      ; If Timer A NOT= 1 Then Print Error
1855      ; If Timer B NOT= 0 Then Print Error
1856      ;
1857      ;-- BGNSUB                      ;////////// BEGIN SUBTEST ////////////
1858      ;                                     T5.2:
1859      ;                                     TRAP C$BSUB
1860      ; Write to TSSR register to soft initialize the controller
1861      ; 5$:
1862      JSR PC,SOFINIT                ;WRITE TO TSSR TO SOFT INITIALIZE
1863      BCS 10$                      ;BR IF SOFT INIT OKAY
1864      MOV R0,R1                    ;SAVE CONTENTS OF TSSR
1865      ERDF ERRNO,SFIERR,SFIMSG    ;DEVICE FATAL DURING INIT
1866      ;                                     TRAP C$ERDF
1867      ;                                     .WORD 511
1868      ;                                     .WORD SFIERR
1869      ;                                     .WORD SFIMSG
1870      ; Do WRITE CHARACTERISTICS to setup a Message Buffer
1871      ; 10$:
1872      JSR PC,T16REST                ;RESTORE PACKET DEFAULTS
1873      CLR FATFLG                    ;CLEAR FATAL ERROR FLAG
1874      MOV #T16PACKET,R4             ;GET THE ADDRESS OF COMMAND PACKET
1875      MOV #8.,PKBCNT(R4)            ;MESSAGE PACKET SIZE NO EXTEND
1876      JSR PC,WRTCHR                 ;DO WRITE CHARACTERISTICS COMMAND
1877      FORCERROR 12$                 ;SSDFORCE ERROR IF FORCER=1
1878      BCS 15$                      ;BR IF CARRY SET (GOOD RETURN)
1879      MOV R0,R1                    ;SAVE CONTENTS OF TSSR
1880      NEXT ERRNO
1881      ; 12$:
1882      ERDF ERRNO,T16SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
1883      ;                                     TRAP C$ERDF
1884      ;                                     .WORD 512
1885      ;                                     .WORD T16SSR
1886      ;                                     .WORD PKTSSR
1887      ; 15$:
1888      INC FATFLG                    ;SET FATAL ERROR FLAG
1889      CKLOOP                        ;LOOP ON ERROR, IF FLAG SET

```

K10

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 5: SUBTEST 2: VERIFY TIMERS A,B

SEQ 127

```

035516 104406                                TRAP      C$CLP1
1877
1878 ;      (* Verify Timers A,B after RESET TIMER with 1 microsecond delay *)
1879 ;      Do a Write Control RESET TIMER with 1 microsecond delay
1880 035520 012700 000001      MOV      #MS.RSD,R0      ;RESET TIMER COMMAND
1881 035524 013701 036362'    MOV      T16D01,R1      ;1 MICROSECOND DELAY
1882 035530 004737 037652'    JSR      PC,T16WMISC      ;SETUP T16PK2 COMMAND PACKET
1883 035534 012704 040000'    MOV      #T16PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
1884 035540 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
1885 035544 004737 016146'    JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
1886 035550                      FORCERROR      32$      ;###FORCE ERROR IF FORCER=1
1887 035564 103407      BCS      40$      ;BR IF CARRY SET (GOOD RETURN)
1888 035566 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
1889 035570                      NEXT.ERRNO
1890 035570      32$:      ERRDF      ERRNO,T162SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      513
                                .WORD      T162SSR
                                .WORD      PKTSSR
                                TRAP      C$CLP1
                                104455
                                035572 001001
                                035574 036477'
                                035576 011656'
1891 035600 005237 002222'    INC      FATFLG      ;SET FATAL ERROR FLAG
1892 035604      40$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                104406
1893 ;      If Timer A NOT= 0 Then Print Error
1894 ;      If Timer B NOT= 0 Then Print Error
1895 035606 005002      CLR      R2      ;INIT EXPD
1896 035610 042702 000010      BIC      #S2.ATIM,R2      ;TIMER A EXPD=0
1897 035614 042702 000004      BIC      #S2.BTIM,R2      ;TIMER B EXPD=0
1898 035620 012700 037762'    MOV      #T16BFSTA,R0      ;GET RECV READ STATUS
1899 035624 016001 000002      MOV      2(R0),R1      ;GET RECV BYTE 2
1900 035630 042701 177763      BIC      #C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
1901 035634                      FORCERROR      72$,NOTSSR ;###
1902 035644 020201      CMP      R2,R1      ;EXPD EQUAL RECV?
1903 035646 001404      BEQ      80$      ;BR IF YES
1904 035650                      NEXT.ERRNO
1905 035650      72$:      ERHRD      ERRNO,T16T01,TIMEXP      ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD      514
                                .WORD      T16T01
                                .WORD      TIMEXP
                                TRAP      C$CLP1
                                104456
                                035652 001002
                                035654 037141'
                                035656 015362'
1906 035660      80$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                104406
1907
1908 ;      Do a Write Control RESET TIMER with 28 microsecond delay
1909 035662 012700 000001      MOV      #MS.RSD,R0      ;RESET TIMER COMMAND
1910 035666 013701 036364'    MOV      T16D28,R1      ;28 MICROSECOND DELAY
1911 035672 004737 037652'    JSR      PC,T16WMISC      ;SETUP T16PK2 COMMAND PACKET
1912 035676 012704 040000'    MOV      #T16PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
1913 035702 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
1914 035706 004737 016146'    JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
1915 035712                      FORCERROR      112$      ;###FORCE ERROR IF FORCER=1
1916 035726 103407      BCS      120$      ;BR IF CARRY SET (GOOD RETURN)
1917 035730 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
1918 035732                      NEXT.ERRNO
1919 035732      112$:      ERRDF      ERRNO,T162SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      515
                                .WORD      T162SSR
                                104455
                                035734 001003
                                035736 036477'

```

Line	Address	Hex	Label	Code	Comment	Trap	Word	PKTSSR
1920	035740	011656'		INC	FATFLG			
1921	035742	005237	002222'	CKLOOP				
1922	035746	104406						
1923				IF Timer A NOT= 1	Then Print Error			
1924	035750	005002		IF Timer B NOT= 1	Then Print Error			
1925	035752	052702	000010	CLR	R2			
1926	035756	052702	000004	BIS	#S2.ATIM,R2			
1927	035762	012700	037762'	BIS	#S2.BTIM,R2			
1928	035766	016001	000002	MOV	#T16BFSTA,R0			
1929	035772	042701	177763	MOV	2(R0),R1			
1930	035776			BIC	#C<S2.ATIM!S2.BTIM>,R1			
1931	036006	020201		FORCERROR	172\$,NOTSSR			
1932	036010	001404		CMP	R2,R1			
1933	036012			BEQ	180\$			
1934	036012			NEXT.ERRNO				
	036012	104456		ERRHRD	ERRNO,T16T28,TIMEXP			
	036014	001004						
	036016	037240'						
	036020	015362'						
1935	036022			180\$:	CKLOOP			
	036022	104406						
1936								
1937								
1938	036024	012700	000001		Do a Write Control RESET TIMER with 53 microsecond delay			
1939	036030	013701	036366'	MOV	#MS.RSD,R0			
1940	036034	004737	037652'	MOV	T16D53,R1			
1941	036040	012704	040000'	JSR	PC,T16WMISC			
1942	036044	010465	000000	MOV	#T16PK2,R4			
1943	036050	004737	016146'	MOV	R4,TSDB(R5)			
1944	036054			JSR	PC,CHKTSSR			
1945	036070	103407		FORCERROR	212\$			
1946	036072	010001		BCS	220\$			
1947	036074			MOV	R0,R1			
1948	036074			NEXT.ERRNO				
	036074	104455		ERRDF	ERRNO,T162SSR,PKTSSR			
	036076	001005						
	036100	036477'						
	036102	011656'						
1949	036104	005237	002222'	INC	FATFLG			
1950	036110			CKLOOP				
	036110	104406						
1951								
1952								
1953	036112	005002		IF Timer A NOT= 0	Then Print Error			
1954	036114	042702	000010	IF Timer B NOT= 1	Then Print Error			
1955	036120	052702	000004	CLR	R2			
1956	036124	012700	037762'	BIC	#S2.ATIM,R2			
1957	036130	016001	000002	BIS	#S2.BTIM,R2			
1958	036134	042701	177763	MOV	#T16BFSTA,R0			
1959	036140			MOV	2(R0),R1			
1960	036150	020201		BIC	#C<S2.ATIM!S2.BTIM>,R1			
1961	036152	001404		FORCERROR	272\$,NOTSSR			

N10

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 5: SUBTEST 2: VERIFY TIMERS A,B

SEQ 130

```

2005 036356          EXIT    TST          ;////////// EXIT TEST ////////////
      036356 104432          TRAP    C$EXIT
      036360 001534          .WORD    L10053-.

2006
2007
2008          ;+
2009          ;LOCAL STORAGE FOR THIS TEST
2010          ;-
2010 036362 000001      T16D01:          .WORD    1          ;1 MICROSECOND DELAY (ACTUALLY .8 MIC)
2011 036364 000040      T16D28:          .WORD    40         ;28 MICROSECOND DELAY (.8 MICROS PER)
2012 036366 000076      T16D53:          .WORD    76         ;53 MICROSECOND
2013 036370 000142      T16D78:          .WORD    142        ;78 MICROSECOND
2014
2015          ;+
2016          ;LOCAL TEXT MESSAGES FOR TEST
2017          ;-
2018 036372          105      170      164  TST16ID:          .ASCIIZ  'Extended Features Switch and Timers A,B'
2019 036442          127      122      111  T16SSR: .ASCIIZ  'WRITE CHARACTERISTICS Failed'
2020 036477          127      122      111  T162SSR: .ASCIIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
2021 036543          127      122      111  T163SSR: .ASCIIZ  'WRITE SUBSYSTEM (Read Status) Failed'
2022 036610          102      165      163  T16TSBA: .ASCIIZ  'Bus Address Register (TSBA) Incorrect after Write Characteristics'
2023 036712          104      141      164  T16LEN: .ASCIIZ  'Data Field Length in Message Buffer Incorrect after Write Characteristics'
2024 037024          124      123      123  T16REJ: .ASCIIZ  'TSSR Function Reject Not Returned When Non-Existent Buffer Address Specified'
d'
2025 037141          124      151      155  T16T01: .ASCIIZ  'Timer A,B Incorrect after Reset Timer with 1 microsecond Delay'
2026 037240          124      151      155  T16T28: .ASCIIZ  'Timer A,B Incorrect after Reset Timer with 28 microsecond Delay'
2027 037340          124      151      155  T16T53: .ASCIIZ  'Timer A,B Incorrect after Reset Timer with 53 microsecond Delay'
2028 037440          124      151      155  T16T78: .ASCIIZ  'Timer A,B Incorrect after Reset Timer with 78 microsecond Delay'
2029          .EVEN
2030
2031
2032          ;+
2033          ; SET DEFAULT PACKET
2034          ;-
2034 037540          T16REST:
2035 037540 012700 037720'      MOV      #T16PACKET,R0          ;PACKET ADDRESS
2036 037544 012720 100004'      MOV      #100004,(R0)+          ;WRITE CHARACTERISTICS WITH ACK
2037 037550 012720 037730'      MOV      #T16DATA,(R0)+          ;ADDRESS OF CHAR DATA BLOCK
2038 037554 005020          CLR      (R0)+          ;EXTENDED ADDRESS
2039 037556 012720 000012'      MOV      #10.,(R0)+          ;SIZE OF MESSAGE PACKET
2040 037562 012720 037742'      MOV      #T16BFR,(R0)+          ;MESSAGE BUFFER ADDRESS
2041 037566 005020          CLR      (R0)+          ;CLEAR EXTENDED BUFFER ADDRESS
2042 037570 012720 000024'      MOV      #20.,(R0)+          ;LENGTH OF MESSAGE BUFFER
2043 037574 005020          CLR      (R0)+          ;CLEAR ESS,ENB,EAI,ERI
2044 037576 005010          CLR      (R0)+          ;CLEAR EXTENDED FEATURES WORD
2045 037600 005037 037742'      CLR      T16BFR          ;CLEAR 1ST LOCATION IN MESSAGE BUFFER
2046 037604 000207          RTS      PC
2047
2048
2049          ;+
2050          ; CLEAR MESSAGE BUFFER
2051          ;-
2051 037606          T16CLRBUF:
2052 037606          SAVREG
2053 037612 012701 037742'      MOV      #T16BFR,R1          ;SAVE R1-R5 UNTIL NEXT RETURN
2054 037616 012702 000026'      MOV      #T16BEND-T16BFR,R2      ;GET MESSAGE BUFFER ADDRESS
2055 037622 105021          10$: CLRB      (R1)+          ;SIZE OF MESSAGE BUFFER IN BYTES
2056 037624 005302          DEC      R2          ;CLEAR A BYTE
2057 037626 003375          BGT      10$          ;DONE?
2058 037630 000207          RTS      PC          ;BR IF NO
2059          ;RETURN

```

```

2060
2061
2062
2063 037632
2064 037632 004737 037606'
2065 037636 012700 040010'
2066 037642 112720 000005
2067 037646 105010
2068 037650 000207
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078 037652
2079 037652
2080 037656 004737 037606'
2081 037662 012702 040010'
2082 037666 112722 000010
2083 037672 110022
2084 037674 110112
2085 037676 000207
2086
2087
2088
2089 037700
2090 037700 012700 040010'
2091 037704 112720 000010
2092 037710 112710 000200
2093 037714 000207
2094
2095
2096
2097
2099 037716
2101
2102
2103
2104 037720
2105 037720 100004
2106 037722 037730'
2107 037724 000000
2108 037726 000012
2109
2110 037730
2111 037730 037742'
2112 037732 000000
2113 037734 000024
2114 037736 000000
2115 037740 000000
2116
2117
2118

```

```

;
; SETUP T16PK2 PACKET FOR READ STATUS
;
T16SRD:
    JSR    P',T16CLRBUF        ;CLEAR MESSAGE BUFFER
    MOV    #T16DT2,R0          ;WRITE SUBSYSTEM DATA BUFFER
    MOVB   #PW,RDSTATUS,(R0),  ;STORE READ STATUS COMMAND IN BSEL0
    CLRB   (R0)                ;CLEAR BSEL1
    RTS     PC                  ;RETURN

;
; SETUP T16PK2 PACKET FOR WRITE MISC.
;
; INPUT:
; R0      CONTAINS WRITE MISC FUNCTION CODE (BSEL1)
; R1      CONTAINS DELAY (TIMES 800 NS) FOR BSEL2
;
T16WMISC:
    SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
    JSR    PC,T16CLRBUF        ;CLEAR MESSAGE BUFFER
    MOV    #T16DT2,R2          ;WRITE SUBSYSTEM DATA BUFFER
    MOVB   #PW,WMISC,(R2),     ;STORE WRITE MISCELLANEOUS IN BSEL0
    MOVB   R0,(R2),            ;STORE WRITE MISC CODE IN BSEL1
    MOVB   R1,(R2),            ;STORE DELAY (RESET TIMER) IN BSEL2
    RTS     PC                  ;RETURN

;
; SETUP T16PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
;
T16SEXT:
    MOV    #T16DT2,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

;
; .BLKB 10-<,-TSV2&7>
;
; WRITE CHARACTERISTICS COMMAND PACKET
;
T16PACKET:
    .WORD 100004                ;COMMAND PACKET FOR TEST
    .WORD T16DATA               ;WRITE CHARACTERISTICS COMMAND, WITH ACK
    .WORD 0                     ;ADDRESS OF CHARACTERISTICS BLOCK
    .WORD 10.                   ;MESSAGE PACKET SIZE

T16DATA:
    .WORD T16BFR                ;CHARACTERISTICS DATA BLOCK
    .WORD 0                     ;ADDRESS OF MESSAGE BUFFER
    .WORD 0                     ;LENGTH OF MESSAGE BUFFER
    .WORD 20.                   ;ESS,END,EAI,ERI
    .WORD 0                     ;EXTENDED FEATURES WORD

;MESSAGE BUFFER

```

C11

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 5: SUBTEST 2: VERIFY TIMERS A,B

SEQ 132

```

2119
2120 037742      T16BFR:      ;BEGIN MESSAGE BUFFER
2121 037742 000000      .WORD 0      ;MESSAGE TYPE
2122 037744 000000      .WORD 0      ;DATA FIELD LENGTH
2123 037746 000000      .WORD 0      ;RBPGR
2124 037750 000000      .WORD 0      ;XST0
2125 037752 000000      .WORD 0      ;XST1
2126 037754 000000      .WORD 0      ;XST2
2127 037756 000000      .WORD 0      ;XST3
2128 037760 000000      .WORD 0      ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
2129 037762      T16BFSTA: .BLKB 6.  ;READ STATUS AND WRITE FIFO BUFFER
2130 037770      T16BEND:      ;END OF MESSAGE BUFFER
2131
2132      ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
2133
2135 037770      .BLKB 10-<,-TSV2&7>
2137 040000      T16PK2:      ;WRITE SUBSYSTEM WITH ACK
2138 040000 100006      .WORD P.WRTSUB!P.ACK      ;LOW ADDRESS OF DATA BLOCK
2139 040002 040010'      .WORD T16DT2      ;HIGH ADDRESS OF DATA BLOCK
2140 040004 000000      .WORD 0      ;MINIMUM MESSAGE PACKET SIZE
2141 040006 000012      .WORD 10.
2142
2143 040010      T16DT2:      ;DATA BLOCK
2144 040010 000      .BYTE 0      ;BSELO
2145 040011 000      .BYTE 0      ;BSEL1
2146 040012 000000      .WORD 0      ;SEL2
2147 040014      .BLKB 64.      ;WRITE FIFO DATA OUTPUT BUFFER
2148
2149
2150 040114      ENDTST
2151 040114
2152 040114 104401      L10053: TRAP C!ETST
2153
2154      .SBTTL TEST 6: FIFO EXERCISER
2155
2156      ;**
2157      ;TEST DESCRIPTION:
2158      ;
2159      ;This test uses the Write Subsystem Memory command to
2160      ;verify the controller's FIFO and associated status and
2161      ;control logic.
2162
2163      ;TEST STEPS:
2164      ;
2165      ;REPEAT FOR LOOPCNT
2166      ;BEGIN
2167      ;Do Subtest 1 - FIFO Initialize status test
2168      ;Do Subtest 2 - FIFO Write Single Byte test
2169      ;Do Subtest 3 - FIFO Write Multiple Bytes test
2170      ;Do Subtest 4 - FIFO Verify ILW Status test
2171      ;Do Subtest 5 - FIFO Input Ready test
2172      ;Do Subtest 6 - FIFO Verify Reset FIFO test
2173      ;END
2174
2175      BGNTST
2176
2177      T6::

```

```

2179 040116 012700 046346'      MOV    #TST17ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
2180 040122 004737 016322'      JSR     PC,TSTSETUP      ;DO INITIAL TEST SETUP
2181 040126 012737 000012 002216'  MOV    #10.,LOOPCNT      ;PERFORM 10 ITERATIONS
2182 040134 004737 017106'      JSR     PC,KTOFF        ;SHUT OFF MEMORY MANAGEMENT
2183 040140 005037 003134'      CLR      KTENABLE        ;REALLY SHUT DOWN KT-11
2184 040144
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211 040144
      040144
      040144 104402
2212
2213
2214 040146
2215 040146 004737 015604'
2216 040152 103405
2217 040154 010001
2218 040156
      040156 104455
      040160 001130
      040162 003646'
      040164 011644'
2219
2220 040166 005037 002222'
2221 040172 012704 047740'
2222 040176 004737 010472'
2223 040202
2224 040216 103407
2225 040220 010001
2226 040222
2227 040222
      040222 104455
      040224 001131

T17LOOP:
      .SBTTL TEST 6: SUBTEST 1: FIFO INITIALIZE STATUS TEST
      **
      TEST 6: SUBTEST 1:
      SUBTEST DESCRIPTION:
      This test verifies, by using the Read Status select code,
      that the FIFO status is in the correct initial state after
      the controller is initialized (Input Ready TRUE,
      Output Ready and Data In Miss FALSE). These status
      signals are checked by the controller's self-test
      sequence, so this subtest is actually more of a partial
      check of the Read Status function than the FIFO status.

      TEST STEPS:
      BEGIN
      Write to TSSR to soft initialize
      Do a WRITE CHARACTERISTICS to setup a message buffer
      Do a WRITE SUBSYSTEM Read Status
      If Input Ready NOT=1 Then Print Error
      If Output Ready NOT=0 Then Print Error
      If Data In Miss NOT=0 Then Print Error
      END
      --
      BGNSUB
      ;;;;;;;;;;;;; BEGIN SUBTEST ;;;;;;;;;;;;;
      T6.1: TRAP C#BSUB

      Write to TSSR register to soft initialize the controller
      5$:
      JSR PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
      BCS 10$             ;BR IF SOFT INIT OKAY
      MOV R0,R1           ;SAVE CONTENTS OF TSSR
      ERDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
      TRAP C#ERDF
      .WORD 600
      .WORD SFIERR
      .WORD SFIMSG

      Do a WRITE CHARACTERISTICS to setup a message buffer
      10$:
      CLR FATFLG          ;CLEAR FATAL ERROR FLAG
      MOV #T17PACKET,R4   ;GET THE ADDRESS OF COMMAND PACKET
      JSR PC,WRTCHR        ;DO WRITE CHARACTERISTICS COMMAND
      FORCERROR 42$       ;FORCE ERROR IF FORCER=1
      BCS 50$             ;BR IF CARRY SET (GOOD RETURN)
      MOV R0,R1           ;SAVE CONTENTS OF TSSR
      NEXT,ERRNO
      42$:
      ERDF ERRNO,T17SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      TRAP C#ERDF
      .WORD 601

```

Address	Offset	Hex	Label	Operation	Comments	Trap	Word	PKTSSR
040226	046355'							T17SSR
040230	011656'							PKTSSR
2228	040232	005237	002222'	INC	FATFLG	;	SET FATAL ERROR FLAG	
2229	040236			50\$:	CKLOOP	;	LOOP ON ERROR, IF FLAG SET	
040236	104406						TRAP	C\$CLP1
2230								
2231								
2232	040240	004737	047524'		Do a Write Subsystem READ STATUS			
2233	040244	012704	050110'		JSR PC,T17SRD	;	SETUP PACKET FOR READ STATUS	
2234	040250	010465	000000		MOV #T17PK2,R4	;	GET WRITE SUBSYSTEM COMMAND PACKET	
2235	040254	004737	016146'		MOV R4,TSDB(R5)	;	SET THE PACKET ADDRESS TO EXECUTE	
2236	040260				JSR PC,CHKTSSR	;	WAIT FOR SSR TO SET	
2237	040274	103407			FORCERROR 62\$	;	FORCE ERROR IF FORCER=1	
2238	040276	010001			BCS 70\$	;	BR IF CARRY SET (GOOD RETURN)	
2239	040300				MOV R0,R1	;	SAVE CONTENTS OF TSSR	
2240	040300				NEXT,ERRNO			
040300	104455			62\$:	ERRDF ERRNO,T173SSR,PKTSSR	;	DEVICE FATAL SSR FAILED TO SET	
040302	001132						TRAP	C\$ERDF
040304	046466'						.WORD	602
040306	011656'						.WORD	T173SSR
2241	040310	005237	002222'		INC	FATFLG	;	SET FATAL ERROR FLAG
2242	040314			70\$:	CKLOOP	;	LOOP ON ERROR, IF FLAG SET	
040314	104406						TRAP	C\$CLP1
2243								
2244	040316	004737	047706'		Set WORDS 0-7 of expd message buffer = to recv since not testing			
2245	040322	012701	046142'		JSR PC,T17SETEXP	;	SET WORDS 0-7 EXPD=RECV	
2246	040326	012702	050002'		MOV #T17EXSTA,R1	;	GET EXPECTED READ STATUS	
2247	040332	012221			MOV #T17BFSTA,R2	;	GET RECV READ STATUS	
2248	040334	011211			MOV (R2)+,(R1)+	;	SET EXPD WORD #8 = RECV TEMP	
2249	040336	052711	000020		MOV (R2),(R1)	;	SET EXPD WORD #9 = RECV TEMP	
2250	040342	042711	000040		BIS #S2.INRDY,(R1)	;	SET EXP INPUT READY= TRUE	
2251	040346	042711	000200		BIC #S2.OUTRDY,(R1)	;	SET EXP OUTPUT READY= FALSE	
2252					BIC #S2.DIM,(R1)	;	SET EXP DATA IN MISS = FALSE	
2253					If Input Ready NOT=1 then Print Error			
2254	040352	005000			If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error			
2255	040354	012701	047762'		CLR R0	;	HIGH RECV ADDRESS FOR CKMSG2	
2256	040360	012702	046122'		MOV #T17BFR,R1	;	LOW RECV ADDRESS FOR CKMSG2	
2257	040364	012703	000024		MOV #T17EXP,R2	;	EXPD ADDRESS	
2258	040370	004737	011310'		MOV #20,R3	;	NUMBER OF BYTES TO COMPARE	
2259	040374				JSR PC,CKMSG2	;	EXPD EQUAL RECV?	
2260	040404	103404			FORCERROR 82\$,NOTSSR	;	GOOD	
2261	040406				BCS 90\$	;	BR IF YES	
2262	040406				NEXT,ERRNO			
040406	104456			82\$:	ERRHRD ERRNO,T171CMP,MSGSTAT	;	REPORT ERROR	
040410	001133						TRAP	C\$ERHRD
040412	046705'						.WORD	603
040414	012160'						.WORD	T171CMP
2263	040416						.WORD	MSGSTAT
040416	104406			90\$:	CKLOOP	;	LOOP ON ERROR, IF FLAG SET	
2264							TRAP	C\$CLP1
2265	040420				ENDSUB	;	////////// END SUBTEST //////////	
040420							L10057:	
040420	104403						TRAP	C\$ESUB
2266								
2267	040422	005737	002222'		TST	FATFLG	;	ANY FATAL ERRORS ?
2268	040426	001402			BEQ	160\$	;	BRANCH IF NOT
2269	040430	004737						

F11

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 6: SUBTEST 1: FIFO INITIALIZE STATUS TEST

SEQ 135

```

2270 040434          1604:
2271
2272
2273          .SBTTL TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST
2274
2275          ;++
2276          ; TEST 6: SUBTEST 2:
2277          ;
2278          ; SUBTEST DESCRIPTION:
2279          ;
2280          ; This subtest verifies the ability of the FIFO to correctly
2281          ; pass a single data byte from input to output. For each
2282          ; of 256 data values (0-377 octal) the following is done:
2283          ; 1. Initial FIFO status is checked
2284          ; 2. The Write FIFO function, specifying a count of
2285          ;    one byte to be written is executed.
2286          ; 3. Read Status is executed and FIFO status is checked.
2287          ; 4. Read FIFO is executed and the data and final status
2288          ;    is checked.
2289          ;
2290          ; TEST STEPS:
2291          ;
2292          ; BEGIN
2293          ;   Write to TSSR to soft initialize
2294          ;   Do a WRITE CHARACTERISTICS to setup a message buffer
2295          ;   Do a Write Subsystem READ STATUS
2296          ;   If Input Ready NOT=1 Then Print Error
2297          ;   If Output Ready NOT=0 Then Print Error
2298          ;   If Data In Miss NOT=0 Then Print Error
2299          ;
2300          ; REPEAT FOR DATA FROM 0 TO 377 OCTAL
2301          ; BEGIN
2302          ;   Do a Write Subsystem WRITE NPR to set tape direction out
2303          ;   Do a Write Subsystem WRITE FIFO with byte count equal to 1
2304          ;   Do a Write Subsystem READ STATUS
2305          ;   If Input Ready NOT=1 Then Print Error
2306          ;   If Output Ready NOT=1 Then Print Error
2307          ;   If Data In Miss NOT=0 Then Print Error
2308          ;   Do Write Subsystem READ FIFO with byte count equal to 1
2309          ;   If Data read from FIFO NOT= Data sent Then Print Error
2310          ;   Do a Write Subsystem READ STATUS
2311          ;   If Input Ready NOT=1 Then Print Error
2312          ;   If Output Ready NOT=0 Then Print Error
2313          ;   If Data In Miss NOT=0 Then Print Error
2314          ; END
2315          ; END
2316          ; --
2317          040434          BGNSUB          ;////////// BEGIN SUBTEST ///////////
          040434          T6.2:          TRAP C$BSUB
          040434 104402
2318
2319          ;
2320          040436          5$: Write to TSSR register to soft initialize the controller
2321          040436 004737 015604: JSR PC,SOFINIT ;WRITE TO TSSR TO SOFT INITIALIZE
2322          040442 103405 BCS 10$ ;BR IF SOFT INIT OKAY
2323          040444 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
2324          040446 ERRDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

SEQ 136

040446	104455						TRAP	C\$ERDF
040450	001133						.WORD	603
040452	003646'						.WORD	SFIERR
040454	011644'						.WORD	SFIMSG

2325				:	Do a WRITE CHARACTERISTICS to setup a message buffer			
2326	040456	005037	002222'	10\$:	CLR FATFLG			;CLEAR FATAL ERROR FLAG
2327	040462	012704	047740'		MOV #T17PACKET,R4			;GET THE ADDRESS OF COMMAND PACKET
2328	040466	004737	010472'		JSR PC,WRTCHR			;DO WRITE CHARACTERISTICS COMMAND
2329	040472				FORCERROR 42\$			;BDDFORCE ERROR IF FORCER=1
2330	040506	103407			BCS 50\$			;BR IF CARRY SET (GOOD RETURN)
2331	040510	010001			MOV R0,R1			;SAVE CONTENTS OF TSSR
2332	040512				NEXT,ERRNO			
2333	040512			42\$:	ERRDF ERRNO,T17SSR,PKTSSR			;DEVICE FATAL SSR FAILED TO SET
	040512	104455					TRAP	C\$ERDF
	040514	001134					.WORD	604
	040516	046365'					.WORD	T17SSR
	040520	011656'					.WORD	PKTSSR
2334	040522	005237	002222'		INC FATFLG			;SET FATAL ERROR FLAG
2335	040526			50\$:	CKLOOP			;LOOP ON ERROR, IF FLAG SET
	040526	104406					TRAP	C\$CLP1
2336				:	Do a Write Subsystem READ STATUS			
2337	040530	004737	047524'		JSR PC,T17SRD			;SETUP PACKET FOR READ STATUS
2338	040534	012704	050110'		MOV #T17PK2,R4			;GET WRITE SUBSYSTEM COMMAND PACKET
2339	040540	010465	000000		MOV R4,TSD8(R5)			;SET THE PACKET ADDRESS TO EXECUTE
2340	040544	004737	016146'		JSR PC,CHKTSSR			;WAIT FOR SSR TO SET
2341	040550				FORCERROR 62\$			;BDDFORCE ERROR IF FORCER=1
2342	040564	103407			BCS 70\$			;BR IF CARRY SET (GOOD RETURN)
2343	040566	010001			MOV R0,R1			;SAVE CONTENTS OF TSSR
2344	040570				NEXT,ERRNO			
2345	040570			62\$:	ERRDF ERRNO,T173SSR,PKTSSR			;DEVICE FATAL SSR FAILED TO SET
	040570	104455					TRAP	C\$ERDF
	040572	001135					.WORD	605
	040574	046466'					.WORD	T173SSR
	040576	011656'					.WORD	PKTSSR
2346	040600	005237	002222'		INC FATFLG			;SET FATAL ERROR FLAG
2347	040604			70\$:	CKLOOP			;LOOP ON ERROR, IF FLAG SET
	040604	104406					TRAP	C\$CLP1
2348				:	Set WORDS 0-7 of expd message buffer - to recv since not testing			
2349	040606	004737	047706'		JSR PC,T17SETEXP			;SET WORDS 0-7 EXPD=RECV
2350	040612	012701	046142'		MOV #T17EXSTA,R1			;GET EXPECTED READ STATUS
2351	040616	012702	050002'		MOV #T17BFSTA,R2			;GET RECV READ STATUS
2352	040622	012221			MOV (R2)+,(R1)+			;SET EXPD WORD #8 = RECV TEMP
2353	040624	011211			MOV (R2),(R1)			;SET EXPD WORD #9 = RECV TEMP
2354	040626	052711	000020		BIS #S2,INRDY,(R1)			;SET EXP INPUT READY= TRUE
2355	040632	042711	000040		BIC #S2,OUTRDY,(R1)			;SET EXP OUTPUT READY= FALSE
2356	040636	042711	000200		BIC #S2,DIM,(R1)			;SET EXP DATA IN MISS = FALSE
2357				:	If Input Ready NOT=1 then Print Error			
2358				:	If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error			
2359	040642	005000			CLR R0			;HIGH RECV ADDRESS FOR CKMSG2
2360	040644	012701	047762'		MOV #T17BFR,R1			;LOW RECV ADDRESS FOR CKMSG2
2361	040650	012702	046122'		MOV #T17EXP,R2			;EXPD ADDRESS
2362	040654	012703	000024		MOV #20,R3			;NUMBER OF BYTES TO COMPARE
2363	040660	004737	011310'		JSR PC,CKMSG2			;EXPD EQUAL RECV?
2364	040664				FORCERROR 82\$,NOTSSR			;BDD
2365	040674	103404			BCS 90\$			;BR IF YES
2366	040676				NEXT,ERRNO			
2367	040676			82\$:	ERRHRD ERRNO,T171CMP,MSGSTAT			;REPORT ERROR

Address	Offset	Hex Data	Assembly	Comments	Trap	Trap Offset	Trap Word	Trap Mask
040676		104456			TRAP	C\$ERHRO		
040700		001136			TRAP			
040702		046705'			TRAP			
040704		012160'			TRAP			
2368	040706		90\$:	CKLOOP				
	040706	104406						
2369								
2370								
2371	040710	012737 000000 002312'		; Repeat for DATA from 0 to 377				
2372	040716		100\$:	MOV #0,DATA				
2373								
2374	040716	012700 000100		; Do a Write Subsystem WRITE NPR to set tape direction out				
2375	040722	004737 047566'		MOV #NPR,OUT,R0				
2376	040726	012704 050110'		JSR PC,T17SNPR				
2377	040732	010465 000000		MOV #T17PK2,R4				
2378	040736	004737 016146'		MOV R4,TSDB(R5)				
2379	040742			JSR PC,CHKTSSR				
2380	040756	103407		FORCERROR 102\$				
2381	040760	010001		BCS 105\$				
2382	040762			MOV R0,R1				
2383	040762		102\$:	NEXT,ERRNO				
	040762	104455		ERRDF ERRNO,T174SSR,PKTSSR				
	040764	001137						
	040766	046533'						
	040770	011656'						
2384	040772	005237 002222'		INC FATFLG				
2385	040776		105\$:	CKLOOP				
	040776	104406						
2386								
2387	041000	012700 000001		; Do a Write Subsystem WRITE FIFO with byte count equal to 1				
2388	041004	012701 002312'		MOV #1,R0				
2389	041010	004737 047612'		MOV #DATA,R1				
2390	041014	012704 050110'		JSR PC,T17WFIF				
2391	041020	010465 000000		MOV #T17PK2,R4				
2392	041024	004737 016146'		MOV R4,TSDB(R5)				
2393	041030			JSR PC,CHKTSSR				
2394	041044	103407		FORCERROR 107\$				
2395	041046	010001		BCS 110\$				
2396	041050			MOV R0,R1				
2397	041050		107\$:	NEXT,ERRNO				
	041050	104455		ERRDF ERRNO,T175SSR,PKTSSR				
	041052	001140						
	041054	046576'						
	041056	011656'						
2398	041060	005237 002222'		INC FATFLG				
2399	041064		110\$:	CKLOOP				
	041064	104406						
2400								
2401								
2402	041066	004737 047524'		; Do a Write Subsystem READ STATUS				
2403	041072	012704 050110'		JSR PC,T17SRD				
2404	041076	010465 000000		MOV #T17PK2,R4				
2405	041102	004737 016146'		MOV R4,TSDB(R5)				
2406								

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

SEQ 138

```

2410 041126 104455 112$: ERRDF ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      041126 001141 TRAP C$ERDF
      041130 046466' .WORD 609
      041134 011656' .WORD T173SSR
      041134 005237 002222' .WORD PKTSSR
2411 041136 005237 002222'
2412 041142 104406 120$: INC FATFLG ;SET FATAL ERROR FLAG
      041142 104406 CKLOOP ;LOOP ON ERROR, IF FLAG SET
      ; TRAP C$CLP1
      ; Set WORDS 0-7 of expd message buffer = to recv since not testing
2413 041144 004737 047706' JSR PC,T17SETEXP ;SET WORDS 0-7 EXPD=RCV
2414 041150 012701 046142' MOV #T17EXSTA,R1 ;GET EXPECTED READ STATUS
2415 041154 012702 050002' MOV #T17BFSTA,R2 ;GET RCV READ STATUS
2416 041160 012221 MOV (R2),R1 ;SET EXPD WORD #8 = RCV TEMP
2417 041162 011211 MOV (R2),R1 ;SET EXPD WORD #9 = RCV TEMP
2418 041164 052711 000020 BIS #S2.INRDY,(R1) ;SET EXP INPUT READY= 1
2419 041170 052711 000040 BIS #S2.OTRDY,(R1) ;SET EXP OUTPUT READY= 1
2420 041174 042711 000200 BIC #S2.DIM,(R1) ;SET EXP DATA IN MISS = 0
2421 041174 042711 000200
2422 041174 042711 000200 ; If Input Ready NOT=1 then Print Error
2423 041174 042711 000200 ; If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
2424 041200 005000 CLR R0 ;HIGH RCV ADDRESS FOR CKMSG2
2425 041202 012701 047762' MOV #T17BFR,R1 ;LOW RCV ADDRESS FOR CKMSG2
2426 041206 012702 046122' MOV #T17EXP,R2 ;EXPD ADDRESS
2427 041212 012703 000024 MOV #20,R3 ;NUMBER OF BYTES TO COMPARE
2428 041216 004737 011310' JSR PC,CKMSG2 ;EXPD EQUAL RCV?
2429 041222 103404 FORCERROR 132$,NOTSSR ;GOOD
2430 041232 103404 BCS 140$ ;BR IF YES
2431 041234 NEXT.ERRNO
2432 041234 104456 132$: ERRHRD ERRNO,T173CMP,MSGSTAT ;REPORT ERROR
      041234 001142 TRAP C$ERHRD
      041236 047063' .WORD 610
      041242 012160' .WORD T173CMP
      041242 012160' .WORD MSGSTAT
2433 041244 104406 140$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
      041244 104406 TRAP C$CLP1
2434 041244 104406
2435 041244 104406 ; Do Write Subsystem READ FIFO with byte count equal to 1
2436 041246 012700 000001 MOV #1,R0 ;SET READ BYTE COUNT
2437 041252 004737 047646' JSR PC,T17RFIF ;SETUP T17PK2 FOR READ FIFO
2438 041256 012704 050110' MOV #T17PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
2439 041262 010465 000000 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
2440 041266 004737 016146' JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
2441 041272 103407 FORCERROR 142$ ;GOODFORCE ERROR IF FORCER=1
2442 041306 103407 BCS 150$ ;BR IF CARRY SET (GOOD RETURN)
2443 041310 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
2444 041312 NEXT.ERRNO
2445 041312 104455 142$: ERRDF ERRNO,T176SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      041312 001143 TRAP C$ERDF
      041316 046642' .WORD 611
      041320 011656' .WORD T176SSR
      041320 011656' .WORD PKTSSR
2446 041322 005237 002222'
2447 041326 104406 150$: INC FATFLG ;SET FATAL ERROR FLAG
      041326 104406 CKLOOP ;LOOP ON ERROR, IF FLAG SET
      ; TRAP C$CLP1
      ; Set WORDS 0-7 of expd message buffer = to recv since not testing
2448 041330 004737 047706' JSR PC,T17SETEXP ;SET WORDS 0-7 EXPD=RCV
2449 041334 012701 046142' MOV #T17EXSTA,R1 ;GET EXPECTED READ STATUS
2450 041334 012702 050002' MOV #T17BFSTA,R2 ;GET RCV READ STATUS
2451 041340 012702 050002'

```

2452	041344	013721	002312'	MOV DATA,(R1)+	;SET EXPD WORD #8 = COUNT DATA	
2453	041350	011211		MOV (R2),(R1)	;SET EXPD WORD #9 = RECV (NOT TESTING)	
2454				; If Data read from FIFO NOT= to Data sent Then Print Error		
2455				; The data is in WORD #8 of the message buffer		
2456	041352	005000		CLR R0	;HIGH RECV ADDRESS FOR CKMSG2	
2457	041354	012701	047762'	MOV #T17BFR,R1	;LOW RECV ADDRESS FOR CKMSG2	
2458	041360	012702	046122'	MOV #T17EXP,R2	;EXPD ADDRESS	
2459	041364	012703	000022	MOV #18.,R3	;NUMBER OF BYTES TO COMPARE	
2460	041370	004737	011310'	JSR PC,CKMSG2	;EXPD EQUAL RECV?	
2461	041374			FORCERROR 152\$,NOTSSR	;###	
2462	041404	103404		BCS 160\$	;BR IF YES	
2463	041406			NEXT,ERRNO		
2464	041406			152\$: ERRHRD ERRNO,T172CMP,MSGSUB	;REPORT ERROR	
	041406	104456				TRAP C\$ERHRD
	041410	001144				.WORD 612
	041412	046767'				.WORD T172CMP
	041414	013552'				.WORD MSGSUB
2465	041416			160\$: CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	041416	104406				TRAP C\$CLP1
2466				; Do a Write Subsystem READ STATUS		
2467				JSR PC,T17SRD	;SETUP PACKET FOR READ STATUS	
2468	041420	004737	047524'	MOV #T17PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
2469	041424	012704	050110'	MOV R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
2470	041430	010465	000000	JSR PC,CHKTSSR	;WAIT FOR SSR TO SET	
2471	041434	004737	016146'	FORCERROR 162\$	;###FORCE ERROR IF FORCER=1	
2472	041440			BCS 170\$	;BR IF CARRY SET (GOOD RETURN)	
2473	041454	103407		MOV R0,R1	;SAVE CONTENTS OF TSSR	
2474	041456	010001		NEXT,ERRNO		
2475	041460			162\$: ERRDF ERRNO,T173SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	
	041460	104455				TRAP C\$ERDF
	041462	001145				.WORD 613
	041464	046466'				.WORD T173SSR
	041466	011656'				.WORD PKTSSR
2477	041470	005237	002222'	INC FATFLG	;SET FATAL ERROR FLAG	
2478	041474			170\$: CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	041474	104406				TRAP C\$CLP1
2479				; Set WORDS 0-7 of expd message buffer = to recv since not testing		
2480	041476	004737	047706'	JSR PC,T17SETEXP	;SET WORDS 0-7 EXPD=RECV	
2481	041502	012701	046142'	MOV #T17EXSTA,R1	;GET EXPECTED READ STATUS	
2482	041506	012702	050002'	MOV #T17BFSTA,R2	;GET RECV READ STATUS	
2483	041512	012221		MOV (R2)+,(R1)+	;SET EXPD WORD #8 = RECV TEMP	
2484	041514	011211		MOV (R2),(R1)	;SET EXPD WORD #9 = RECV TEMP	
2485	041516	052711	000020	BIS #S2.INRDY,(R1)	;SET EXP INPUT READY= 1	
2486	041522	042711	000040	BIC #S2.OTRDY,(R1)	;SET EXP OUTPUT READY= 0	
2487	041526	042711	000200	BIC #S2.DIM,(R1)	;SET EXP DATA IN MISS = 0	
2488				; If Input Ready NOT=1 then Print Error		
2489				; If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error		
2490	041532	005000		CLR R0	;HIGH RECV ADDRESS FOR CKMSG2	
2491	041534	012701	047762'	MOV #T17BFR,R1	;LOW RECV ADDRESS FOR CKMSG2	
2492	041540	012702	046122'	MOV #T17EXP,R2	;EXPD ADDRESS	
2493	041544	012703	000024	MOV #20.,R3	;NUMBER OF BYTES TO COMPARE	
2494	041550	004737	011310'	JSR PC,CKMSG2	;EXPD EQUAL RECV?	
2495	041554			FORCERROR 172\$,NOTSSR	;###	
2496	041564	103404		BCS 180\$	;BR IF YES	
2497	041566			NEXT,ERRNO		
2498	041566			172\$: ERRHRD ERRNO,T174CMP,MSGSTAT	;REPORT ERROR	

K11

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

SEQ 140

```

041566 104456
041570 001146
041572 047147'
041574 012160'
2499 041576 180$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
041576 104406 ;TRAP C$ERHRD
;WORD 614
;WORD T174CMP
;WORD MSGSTAT
2500 041600 FORCEEXIT 205$ ;END
2501 041610 005237 002312' INC DATA ;GET NEXT TEST DATA
2502 041614 023727 002312' 000377 CMP DATA,0377 ;DONE 0 TO 377?
2503 041622 101002 BHI 205$ ;BR IF YES
2504 041624 000137 040716' JMP 100$ ;DO ANOTHER TEST PATTERN
2505 041630 205$:
2506
2507 041630 ENDSUB ;////////// END SUBTEST //////////
041630 L10060:
041630 104403 ;TRAP C$ESUB
2508
2509 041632 005737 002222' TST FATFLG ;ANY FATAL ERRORS ?
2510 041636 001402 BEQ 260$ ;BRANCH IF NO?
2511 041640 004737 017014' JSR PC,CKDROP ;TRY TO DROP THE UNIT
2512 041644 260$:
2513
2514 .SBTTL TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST
2515
2516 ;**
2517 ; TEST 6: SUBTEST 3:
2518 ;
2519 ; SUBTEST DESCRIPTION:
2520 ;
2521 ; This subtest verifies the ability of the FIFO to correctly
2522 ; pass a multiple data bytes from input to output.
2523 ; The following sequence is done with various data patterns
2524 ; and byte counts from 2 to 64.
2525 ; 1. Initial FIFO status is checked
2526 ; 2. The Write FIFO function.
2527 ; 3. Read Status is executed and FIFO status is checked.
2528 ; 4. Read FIFO is executed and the data and final status
2529 ; is checked.
2530 ;
2531 ; TEST STEPS:
2532 ;
2533 ; BEGIN
2534 ; Write to TSSR to soft initialize
2535 ; Do a WRITE CHARACTERISTICS to setup a message buffer
2536 ; Do a Write Subsystem READ STATUS
2537 ; If Input Ready NOT=1 Then Print Error
2538 ; If Output Ready NOT=0 Then Print Error
2539 ; If Data In Miss NOT=0 Then Print Error
2540 ; If Last Word NOT=0 Then Print Error
2541 ; REPEAT FOR DATA 0 TO 377, 377 TO 0, FLOATING 1'S,0'S AND ALL 1'S/0'S
2542 ; REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
2543 ; BEGIN
2544 ; Do a Write Subsystem WRITE NPR to set tape direction out
2545 ; Do a Write Subsystem WRITE FIFO
2546 ; Do a Write Subsystem READ STATUS
2547 ; If Input Ready NOT=1 Then Print Error
2548 ; If Output Ready NOT=1 Then Print Error

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

SEQ 141

```

2549      ;      If Data In Miss NOT=0      Then Print Error
2550      ;      If Last Word      NOT=0      Then Print Error
2551      ;      Do Write Subsystem READ FIFO
2552      ;      If Data read from FIFO NOT= to Data sent Then Print Error
2553      ;      Do a Write Subsystem READ STATUS
2554      ;      If Input Ready NOT=1      Then Print Error
2555      ;      If Output Ready NOT=0      Then Print Error
2556      ;      If Data In Miss NOT=0      Then Print Error
2557      ;      If Last Word      NOT=0      Then Print Error
2558      ;      END
2559      ;      END
2560      ;      ---
2561      041644      BGNSUB                      ;////////// BEGIN SUBTEST //////////
          041644                      T6.3:
          041644      104402                      TRAP      C$BSUB

2562      ;
2563      ;      Write to TSSR register to soft initialize the controller
2564      5$:
2565      041646      004737      015604'      JSR      FC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
2566      041652      103405      BCS      10$      ;BR IF SOFT INIT OKAY
2567      041654      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2568      041656      104455      ERDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
          041660      001146      TRAP      C$ERDF
          041662      003646'      .WORD      614
          041664      011644'      .WORD      SFIERR
          .WORD      SFIMSG

2569      ;      Do a WRITE CHARACTERISTICS to setup a message buffer
2570      10$:
2571      041666      005037      002222'      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
2572      041672      012704      047740'      MOV      @T17PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
2573      041702      JSR      PC,WRCHRR      ;DO WRITE CHARACTERISTICS COMMAND
2574      041716      103407      FORCERROR      42$      ;GOODFORCE ERROR IF FORCER=1
2575      041720      010001      BCS      50$      ;BR IF CARRY SET (GOOD RETURN)
2576      041722      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2577      42$:
          041722      104455      ERDF      ERRNO,T17SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
          041724      001147      TRAP      C$ERDF
          041726      046365'      .WORD      615
          041730      011656'      .WORD      T17SSR
          .WORD      PKTSSR

2578      041732      005237      002222'      INC      FATFLG      ;SET FATAL ERROR FLAG
2579      041736      104406      50$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
          TRAP      C$CLP1

2580      ;      Do a Write Subsystem READ STATUS
2581      041740      004737      047524'      JSR      PC,T17SRD      ;SETUP PACKET FOR READ STATUS
2582      041744      012704      050110'      MOV      @T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
2583      041750      010465      000000'      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
2584      041754      004737      016140'      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
2585      041760      FORCERROR      62$      ;GOODFORCE ERROR IF FORCER=1
2586      041774      103407      BCS      70$      ;BR IF CARRY SET (GOOD RETURN)
2587      041776      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2588      042000      NEXT,ERRNO
2589      62$:
          042000      104455      ERDF      ERRNO,T173SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
          042002      001150      TRAP      C$ERDF
          042004      046466'      .WORD      616
          042006      011556'      .WORD      T173SSR
          .WORD      PKTSSR

2590      042010      005237      002222'      INC      FATFLG      ;SET FATAL ERROR FLAG

```

M11

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

SEQ 142

```

2591 042014      70$: CKLOOP                                ;LOOP ON ERROR, IF FLAG SET
      042014 104406                                         TRAP C$CLP1
2592      ; Set WORDS 0-7 of expd message buffer = to recv since not testing
2593 042016 004737 047706' JSR PC,T17SETEXP      ;SET WORDS 0-7 EXPD=RECV
2594 042022 012701 046142' MOV #T17EXSTA,R1      ;GET EXPECTED READ STATUS
2595 042026 012702 050002' MOV #T17BFSTA,R2      ;GET RECV READ STATUS
2596 042032 012221      MOV (R2)+,(R1)+      ;SET EXPD WORD #8 = RECV TEMP
2597 042034 011211      MOV (R2),(R1)      ;SET EXPD WORD #9 = RECV TEMP
2598 042036 052711 000020 BIS #S2.INRDY,(R1)      ;SET EXP INPUT READY= 1
2599 042042 042711 000040 BIC #S2.OTRDY,(R1)      ;SET EXP OUTPUT READY= 0
2600 042046 042711 000200 BIC #S2.DIM,(R1)      ;SET EXP DATA IN MISS = 0
2601 042052 042711 000100 BIC #S2.ILW,(R1)      ;SET EXP LAST WORD (ILW)=0
2602      ; If Input Ready NOT=1 then Print Error
2603      ; If Output Ready NOT=0 or Data in Miss NOT=0 then Print Error
2604      ; If Last Word NOT=0 Then Print Error
2605 042056 005000      CLR R0      ;HIGH RECV ADDRESS FOR CKMSG2
2606 042060 012701 047762' MOV #T17BFR,R1      ;LOW RECV ADDRESS FOR CKMSG2
2607 042064 012702 046122' MOV #T17EXP,R2      ;EXPD ADDRESS
2608 042070 012703 000024 MOV #20,R3      ;NUMBER OF BYTES TO COMPARE
2609 042074 004737 011310' JSR PC,CKMSG2      ;EXPD EQUAL RECV?
2610 042100      FORCERROR 82$,NOTSSR      ;880
2611 042110 103404      BCS 90$      ;BR IF YES
2612 042112      NEXT.ERRNO
2613 042112 82$: ERRHRD ERRNO,T171CMP,MSGSTAT ;REPORT ERROR
      042112 104456                                         TRAP C$ERRHRD
      042114 001151                                         .WORD 617
      042116 046705'                                         .WORD T171CMP
      042120 012160'                                         .WORD MSGSTAT
2614 042122 90$: CKLOOP                                ;LOOP ON ERROR, IF FLAG SET
      042122 104406                                         TRAP C$CLP1
2615
2616
2617
2618      ; REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
2619      ; TSTFLAG =1 FOR INCREMENT TEST PATTERN
2620      ; =2 FOR DECREMENT TEST PATTERN
2621      ; =3 FOR TSTBLK TABLE PATTERN
2622 042124 012737 000001 002314' MOV #1,TSTFLAG      ;TEST PATTERN FLAG
2623 042132      95$:
2624 042132 012737 000002 002310' MOV #2,COUNT      ;GET FIRST BYTE COUNT
2625 042140      100$:
2626      ; Do a Write Subsystem WRITE NPR to set tape direction out
2627 042140 012700 000100 MOV #NP.OUT,R0      ;SET TAPE DIRECTION OUT
2628 042144 004737 047566' JSR PC,T17SNPR      ;SETUP T17PK2 FOR WRITE NPR
2629 042150 012704 050110' MOV #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
2630 042154 010465 000000 MOV R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
2631 042160 004737 016146' JSR PC,CHKTSSR      ;WAIT FOR SSR TO SET
2632 042164      FORCERROR 102$      ;880FORCE ERROR IF FORCER=1
2633 042200 103407      BCS 105$      ;BR IF CARRY SET (GOOD RETURN)
2634 042202 010001      MOV R0,R1      ;SAVE CONTENTS OF TSSR
2635 042204      NEXT.ERRNO
2636 042204 102$: ERRDF ERRNO,T174SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      042204 104455                                         TRAP C$ERRDF
      042206 001152                                         .WORD 618
      042210 046533'                                         .WORD T174SSR
      042212 011656'                                         .WORD PKTSSR
2637 042214 005237 002222' INC FATFLG      ;SET FATAL ERROR FLAG

```

N11

ISV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

SEQ 143

```

2638 042220      105$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      042220 104406      TRAP      C$CLP1
2639      ; Do a Write Subsystem WRITE FIFO
2640 042222 004737 047666' JSR      PC,T17CLEXP      ;CLEAR EXPD BUFFER
2641 042226 012701 046244' MOV      #T17WFDATA,R1      ;EXPD WRITE FIFO DATA BUFFER
2642 042232 013702 002310' MOV      COUNT,R2      ;TEST PATTERN SIZE
2643 042236 022737 000001 002314' CMP      #1,TSTFLAG      ;INCREMENT PATTERN THIS TIME THRU?
2644 042244 001005      BNE      115$      ;BR IF NO
2645 042246 005000      CLR      R0      ;INCREMENT TEST PATTERN
2646 042250 110021      110$: MOVB      R0,(R1)+      ;STORE INCREMENT TEST BYTE
2647 042252 005200      INC      R0      ;SET NEXT PATTERN
2648 042254 005302      DEC      R2      ;DONE?
2649 042256 003374      BGT      110$      ;BR IF NO
2650 042260 022737 000002 002314' 115$: CMP      #2,TSTFLAG      ;DECREMENT PATTERN THIS TIME THRU?
2651 042266 001006      BNE      125$      ;BR IF NO
2652 042270 012700 000377      MOV      #377,R0      ;DECREMENT TEST PATTERN
2653 042274 110021      120$: MOVB      R0,(R1)+      ;STORE DECREMENT TEST BYTE
2654 042276 005300      DEC      R0      ;SET NEXT PATTERN
2655 042300 005302      DEC      R2      ;DONE?
2656 042302 003374      BGT      120$      ;BR IF NO
2657 042304 022737 000003 002314' 125$: CMP      #3,TSTFLAG      ;TSTBLK PATTERNS THIS TIME THRU?
2658 042312 001005      BNE      135$      ;BR IF NO
2659 042314 012700 002752'      MOV      #TSTBLK,R0      ;FLOAT 1'S/0'S ETC. TEST TABLE
2660 042320 112021      130$: MOVB      (R0)+,(R1)+      ;STORE A TSTBLK BYTE
2661 042322 005302      DEC      R2      ;DONE?
2662 042324 003375      BGT      130$      ;BR IF NO
2663 042326      135$:      MOV      COUNT,R0      ;FIFO BYTE COUNT
2664 042326 013700 002310'      MOV      #T17WFDATA,R1      ;FIFO WRITE DATA ADDRESS
2665 042332 012701 046244'      JSR      PC,T17WFIF      ;SETUP T17PK2 FOR WRITE FIFO
2666 042336 004737 047612'      MOV      #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
2667 042342 012704 050110'      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
2668 042346 010465 000000      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
2669 042352 004737 016146'      FORCERROR      142$      ;GOODFORCE ERROR IF FORCER=1
2670 042356      BCS      150$      ;BR IF CARRY SET (GOOD RETURN)
2671 042372 103407      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2672 042374 010001      NEXT,ERRNO
2673 042376      142$: ERRDF      ERRNO,T175SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
2674 042376      104455      TRAP      C$ERDF
2675 042400 001153      .WORD      619
2676 042402 046576'      .WORD      T175SSR
2677 042404 011656'      .WORD      PKTSSR
2678 042406 005237 002222'      INC      FATFLG      ;SET FATAL ERROR FLAG
2679 042412 104406      150$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
2680      ; Do a Write Subsystem READ STATUS      TRAP      C$CLP1
2681 042414 004737 047524' JSR      PC,T17SRD      ;SETUP PACKET FOR READ STATUS
2682 042420 012704 050110' MOV      #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
2683 042424 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
2684 042430 004737 016146' JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
2685 042434      FORCERROR      157$      ;GOODFORCE ERROR IF FORCER=1
2686 042450 103407      BCS      160$      ;BR IF CARRY SET (GOOD RETURN)
2687 042452 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2688 042454      NEXT,ERRNO
2689 042454      157$: ERRDF      ERRNO,T173SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      104455      TRAP      C$ERDF

```

Address	Offset	Hex	Label	Operation	Comments	Trap	Trap Word	Trap Data
2688	042456	001154						620
2689	042460	046466						T173SSR
2690	042462	011656						PKTSSR
2691	042464	005237	002222	INC	FATFLG	SET FATAL ERROR FLAG		
2692	042470	104406		CKLOOP		LOOP ON ERROR, IF FLAG SET		
2693	042470	104406				TRAP		C#CLP1
2694	042472	004737	047706		Set WORDS 0-7 of expd message buffer = to rcv since not testing			
2695	042476	012701	046142	JSR	PC,T17SETEXP	SET WORDS 0-7 EXPD=RCV		
2696	042502	012702	050002	MOV	#T17EXSTA,R1	GET EXPECTED READ STATUS		
2697	042506	012221		MOV	#T17BFSTA,R2	GET RCV READ STATUS		
2698	042510	011211		MOV	(R2), (R1)	SET EXPD WORD #8 = RCV TEMP		
2699	042512	052711	000020	MOV	(R2), (R1)	SET EXPD WORD #9 = RCV TEMP		
2700	042516	052711	000040	BIS	#S2.INRDY, (R1)	SET EXP INPUT READY= 1		
2701	042522	042711	000200	BIS	#S2.OUTRDY, (R1)	SET EXP OUTPUT READY= 1		
2702	042526	042711	000100	BIC	#S2.DIM, (R1)	SET EXP DATA IN MISS = 0		
2703	042532	005000		BIC	#S2.ILW, (R1)	SET EXP LAST WORD (ILW)=0		
2704	042534	012701	047762		If Input Ready NOT=1 then Print Error			
2705	042540	012702	046122		If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error			
2706	042544	012703	000024	CLR	RO	HIGH RCV ADDRESS FOR CKMSG2		
2707	042550	004737	011310	MOV	#T17BFR,R1	LOW RCV ADDRESS FOR CKMSG2		
2708	042554			MOV	#T17EXP,R2	EXPD ADDRESS		
2709	042564	103404		MOV	#20, R3	NUMBER OF BYTES TO COMPARE		
2710	042566			JSR	PC,CKMSG2	EXPD EQUAL RCV?		
2711	042566			FORCERROR	162#,NOTSSR	SSR		
2712	042570	001155		BCS	170#	BR IF YES		
2713	042572	047063		NEXT,ERRNO				
2714	042574	012160		ERRHRD	ERRNO,T173CMP,MSGSTAT	REPORT ERROR		
2715	042576	104456				TRAP		C#ERHRD
2716	042576	104406				WORD		621
2717	042576	104406				WORD		T173CMP
2718	042576	104406				WORD		MSGSTAT
2719	042576	104406				TRAP		C#CLP1
2720	042600	013700	002310		Do Write Subsystem READ FIFO			
2721	042604	004737	047646	MOV	COUNT,RO	SET READ BYTE COUNT		
2722	042610	012704	050110	JSR	PC,T17RFIF	SETUP T17PK2 FOR READ FIFO		
2723	042614	010465	000000	MOV	#T17PK2,R4	GET WRITE SUBSYSTEM COMMAND PACKET		
2724	042620	004737	016146	MOV	R4,TSDB(R5)	SET THE PACKET ADDRESS TO EXECUTE		
2725	042624			JSR	PC,CHKTSSR	WAIT FOR SSR TO SET		
2726	042640	103407		FORCERROR	172#	FORCE ERROR IF FORCER=1		
2727	042642	010001		BCS	180#	BR IF CARRY SET (GOOD RETURN)		
2728	042644			MOV	RO,R1	SAVE CONTENTS OF TSSR		
2729	042644			NEXT,ERRNO				
2730	042644	104455		ERRDF	ERRNO,T176SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET		
2731	042646	001156				TRAP		C#ERDF
2732	042650	046642				WORD		622
2733	042652	011656				WORD		T176SSR
2734	042654	005237	002222			WORD		PKTSSR
2735	042660			INC	FATFLG	SET FATAL ERROR FLAG		
2736	042660	104406		CKLOOP		LOOP ON ERROR, IF FLAG SET		
2737	042660	104406				TRAP		C#CLP1
2738	042662	005000			If Data read from FIFO NOT= to Data sent Then Print Error			
2739	042664	012702	046244	CLR	RO	HIGH RCV ADDRESS FOR CKMSG2		
2740	042664			MOV	#T17WFDATA,R2	GET EXPECTED ADDRESS FOR CKMSG2		

C12

TSV5 - HARDWARE TESTS MAC80 M1113 01-FEB-84 17:02
TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

SEQ 145

2731	042670	012701	050002'	MOV	#T17BFSTA,R1	GET RECEIVED ADDRESS FOR CKMSG2	
2732	042674	013703	002310'	MOV	COUNT,R3	NUMBER OF BYTES TO COMPARE	
2733	042700	004737	011310'	JSR	PC,CKMSG2	EXPD EQUAL RECV?	
2734	042704			FORCERROR	192#,NOTSSR	ISDD	
2735	042714	103406		BCS	200#	IBR IF YES	
2736	042716			NEXT,ERRNO			
2737	042716	013701	002310'	192#:	MOV COUNT,R1	GET BYTE COUNT	
2738	042722			ERRHRD	ERRNO,T175CMP,FIFEXP	REPORT ERROR	
	042722	104456					TRAP C#ERRHRD
	042724	001157					.WORD 623
	042726	047232'					.WORD T175CMP
	042730	012000'					.WORD FIFEXP
2739	042732			200#:	CKLOOP	LOOP ON ERROR, IF FLAG SET	
	042732	104406					TRAP C#CLP1
2740							
2741					Do a Write Subsystem READ STATUS		
2742	042734	004737	047524'	JSR	PC,T17SRD	SETUP PACKET FOR READ STATUS	
2743	042740	012704	050110'	MOV	#T17PK2,R4	GET WRITE SUBSYSTEM COMMAND PACKET	
2744	042744	010465	000000'	MOV	R4,TSD8(R5)	SET THE PACKET ADDRESS TO EXECUTE	
2745	042750	004737	016146'	JSR	PC,CHKTSSR	WAIT FOR SSR TO SET	
2746	042754			FORCERROR	212#	ISDD FORCE ERROR IF FORCER=1	
2747	042770	103407		BCS	220#	IBR IF CARRY SET (GOOD RETURN)	
2748	042772	010001		MOV	R0,R1	SAVE CONTENTS OF TSSR	
2749	042774			NEXT,ERRNO			
2750	042774			212#:	ERRDF ERRNO,T173SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET	
	042774	104455					TRAP C#ERRDF
	042776	001160					.WORD 624
	043000	046466'					.WORD T173SSR
	043002	011656'					.WORD PKTSSR
2751	043004	005237	002222'	INC	FATFLG	SET FATAL ERROR FLAG	
2752	043010			220#:	CKLOOP	LOOP ON ERROR, IF FLAG SET	
	043010	104406					TRAP C#CLP1
2753					Set WORDS 0-7 of expd message buffer to recv since not testing		
2754	043012	004737	047706'	JSR	PC,T17SETEXP	SET WORDS 0-7 EXPD=RECV	
2755	043016	012701	046142'	MOV	#T17EXSTA,R1	GET EXPECTED READ STATUS	
2756	043022	012702	050002'	MOV	#T17BFSTA,R2	GET RECV READ STATUS	
2757	043026	012221		MOV	(R2), (R1)	SET EXPD WORD #8 = RECV TEMP	
2758	043030	011211		MOV	(R2), (R1)	SET EXPD WORD #9 = RECV TEMP	
2759	043032	052711	000020	BIS	#S2.INRDY,(R1)	SET EXP INPUT READY = 1	
2760	043036	042711	000040	BIC	#S2.OUTRDY,(R1)	SET EXP OUTPUT READY = 0	
2761	043042	042711	000200	BIC	#S2.DIM,(R1)	SET EXP DATA IN MISS = 0	
2762	043046	042711	000100	BIC	#S2.ILW,(R1)	SET EXP LAST WORD (ILW)=0	
2763					If Input Ready NOT=1 then Print Error		
2764					If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error		
2765	043052	005000		CLR	R0	HIGH RECV ADDRESS FOR CKMSG2	
2766	043054	012701	047762'	MOV	#T17BFR,R1	LOW RECV ADDRESS FOR CKMSG2	
2767	043060	012702	046122'	MOV	#T17EXP,R2	EXPD ADDRESS	
2768	043064	012703	000024	MOV	#20,R3	NUMBER OF BYTES TO COMPARE	
2769	043070	004737	011310'	JSR	PC,CKMSG2	EXPD EQUAL RECV?	
2770	043074			FORCERROR	232#,NOTSSR	ISDD	
2771	043104	103404		BCS	240#	IBR IF YES	
2772	043106			NEXT,ERRNO			
2773	043106			232#:	ERRHRD ERRNO,T174CMP,MSGSTAT	REPORT ERROR	
	043106	104456					TRAP C#ERRHRD
	043110	001161					.WORD 625
	043112	047147'					.WORD T174CMP
	043114	012160'					.WORD MSGSTAT

D12

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

SEQ 146

```

2774 043116          240$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      043116 104406          TRAP          C$CLP1
2775 043120          FORCEEXIT          250$ ;END
2776 043130 005237 002310'          INC      COUNT          ;GET NEXT BYTE COUNT
2777 043134 023727 002310' 000077          CMP      COUNT,077 ;DONE 0 TO 77
2778 043142 101002          BHI      250$ ;BR IF YES
2779 043144 000137 042140'          JMP      100$ ;DO ANOTHER BYTE COUNT
2780 043150 005237 002314'          250$: INC      TSTFLAG ;GET NEXT TEST PATTERN CODE
2781 043154 023727 002314' 000003          CMP      TSTFLAG,03 ;DONE INC,DEC,TSTBLK PATTERNS?
2782 043162 101002          BHI      255$ ;BR IF YES
2783 043164 000137 042132'          JMP      95$ ;DO ANOTHER TEST PATTERN
2784 043170          255$:          ;//////////////// END SUBTEST //////////////////
2785 043170          ENDSUB          L10061:          TRAP          C$ESUB
      043170 104403
2786
2787 043172 005737 002222'          TST      FATFLG          ;ANY FATAL ERRORS ?
2788 043176 001402          BEQ      260$ ;BRANCH IF NOT
2789 043200 004737 017014'          JSR      PC,CKDROP ;TRY TO DROP THE UNIT
2790 043204          260$:
2791
2792          .SBTIL TEST 6: SUBTEST 4: FIFO Verify ILW Status
2793
2794
2795          ;
2796          ; TEST 6: SUBTEST 4:
2797          ; SUBTEST DESCRIPTION:
2798          ;
2799          ; This subtest verifies that reading the FIFO when it is
2800          ; empty causes the Last Word (ILW) status to assert.
2801          ;
2802          ; TEST STEPS:
2803          ;
2804          ; BEGIN
2805          ; Write to TSSR to soft initialize
2806          ; Do Write Subsystem READ FIFO with byte count equal to 1
2807          ; Do a Write Subsystem READ STATUS
2808          ; If Input Ready NOT=1 Then Print Error
2809          ; If Output Ready NOT=0 Then Print Error
2810          ; If Data In Miss NOT=0 Then Print Error
2811          ; If Last Word (ILW) NOT=1 Then Print Error
2812          ; END
2813          ; --
2814 043204          BGNSUB          ;//////////////// BEGIN SUBTEST //////////////////
      043204          T6.4:          TRAP          C$BSUB
      043204 104402
2815
2816          ; Write to TSSR register to soft initialize the controller
2817 043206          5$:          JSR      PC,SOFINIT          ;WRITE TO TSSR TO SOFT INITIALIZE
2818 043206 004737 015604'          BCS      10$ ;BR IF SOFT INIT OKAY
2819 043212 103405          MOV      R0,R1 ;SAVE CONTENTS OF TSSR
2820 043214 010001          ERDIF      ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
2821 043216          TRAP          C$ERDF
      043216 104455          .WORD      625
      043220 001161          .WORD      SFIERR
      043222 003646'          .WORD      SFIMSG
      043224 011644'

```

```

2822      ; Do a WRITE CHARACTERISTICS to setup a message buffer
2823 043226 005037 002222' 10$: CLR FATFLG ;CLEAR FATAL ERROR FLAG
2824 043232 012704 047740' MOV #T17PK2,R4 ;GET THE ADDRESS OF COMMAND PACKET
2825 043236 004737 010472' JSR PC,WRTCHR ;DO WRITE CHARACTERISTICS COMMAND
2826 043242 FORCERROR 42$ ;DO FORCE ERROR IF FORCER=1
2827 043256 103407 BCS 50$ ;BR IF CARRY SET (GOOD RETURN)
2828 043260 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
2829 043262 NEXT.ERRNO
2830 043262 42$: ERRDF ERRNO,T17SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      TRAP C$ERDF
      .WORD 626
      .WORD T17SSR
      .WORD PKTSSR
2831 043272 005237 002222' 50$: INC FATFLG ;SET FATAL ERROR FLAG
2832 043276 CKLOOP ;LOOP ON ERROR, IF FLAG SET
      TRAP C$CLP1
2833      ; Do Write Subsystem READ FIFO with byte count equal to 1
2834      MOV #1,R0 ;SET READ BYTE COUNT
2835 043300 012700 000001' JSR PC,T17RFIF ;SETUP T17PK2 FOR READ FIFO
2836 043304 004737 047646' MOV #T17PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
2837 043310 012704 050110' MOV R4,TSD8(R5) ;SET THE PACKET ADDRESS TO EXECUTE
2838 043314 010465 000000' JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
2839 043320 004737 016146' FORCERROR 142$ ;DO FORCE ERROR IF FORCER=1
2840 043324 BCS 150$ ;BR IF CARRY SET (GOOD RETURN)
2841 043340 103407 MOV R0,R1 ;SAVE CONTENTS OF TSSR
2842 043342 010001 NEXT.ERRNO
2843 043344 142$: ERRDF ERRNO,T176SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      TRAP C$ERDF
      .WORD 627
      .WORD T176SSR
      .WORD PKTSSR
2845 043354 005237 002222' 150$: INC FATFLG ;SET FATAL ERROR FLAG
2846 043360 CKLOOP ;LOOP ON ERROR, IF FLAG SET
      TRAP C$CLP1
2847      ; Do a Write Subsystem READ STATUS
2848      JSR PC,T17SRD ;SETUP PACKET FOR READ STATUS
2849 043362 004737 047524' MOV #T17PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
2850 043366 012704 050110' MOV R4,TSD8(R5) ;SET THE PACKET ADDRESS TO EXECUTE
2851 043372 010465 000000' JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
2852 043376 004737 016146' FORCERROR 162$ ;DO FORCE ERROR IF FORCER=1
2853 043402 BCS 170$ ;BR IF CARRY SET (GOOD RETURN)
2854 043416 103407 MOV R0,R1 ;SAVE CONTENTS OF TSSR
2855 043420 010001 NEXT.ERRNO
2856 043422 162$: ERRDF ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      TRAP C$ERDF
      .WORD 628
      .WORD T173SSR
      .WORD PKTSSR
2858 043432 005237 002222' 170$: INC FATFLG ;SET FATAL ERROR FLAG
2859 043436 CKLOOP ;LOOP ON ERROR, IF FLAG SET
      TRAP C$CLP1
2860      ; Set WORDS 0-7 of expd message buffer = to recv since not testing
2861 043440 004737 047706' JSR PC,T17SETEXP ;SET WORDS 0-7 EXPD=RCV
2862 043444 012701 046142' MOV #T17EXSTA,R1 ;GET EXPECTED READ STATUS
2863 043450 012702 050002' MOV #T17BFSTA,R2 ;GET RCV READ STATUS

```

```

2864 043454 012221      MOV      (R2), (R1)+      ;SET EXPD WORD #8 = RECV TEMP
2865 043456 011211      MOV      (R2), (R1)      ;SET EXPD WORD #9 = RECV TEMP
2866 043460 052711 000020  BIS      #S2.INRDY, (R1)  ;SET EXP INPUT READY= 1
2867 043464 042711 000040  BIC      #S2.OTRDY, (R1)  ;SET EXP OUTPUT READY= 0
2868 043470 042711 000200  BIC      #S2.DIM, (R1)    ;SET EXP DATA IN MISS = 0
2869 043474 052711 000100  BIS      #S2.ILW, (R1)    ;SET EXP LAST WORD (ILW)=1
2870                                     ; If Input Ready NOT=1 then Print Error
2871                                     ; If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
2872                                     ; If Last Word (ILW) NOT=1 Then Print Error
2873 043500 005000      CLR      R0              ;HIGH RECV ADDRESS FOR CKMSG2
2874 043502 012700 41702'  MOV      #T178FR, R1      ;LOW RECV ADDRESS FOR CKMSG2
2875 043506 012700 41702'  MOV      #T17EXP, R2     ;EXPD ADDRESS
2876 043512 012700 000024  MOV      #20, R3        ;NUMBER OF BYTES TO COMPARE
2877 043516 004737 011310' JSR      PC, CKMSG2      ;EXPD EQUAL RECV?
2878 043522                                     FORCERROR 172$, NOTSSR ;000
2879 043532 103404      BCS      180$          ;BR IF YES
2880 043534                                     NEXT, ERRNO
2881 043534 172$: ERRHRD  ERRNO, T176CMP, MSGSTAT ;REPORT ERROR
                                     TRAP      C$ERHRD
                                     .WORD     629
2882 043544 180$: CKLOOP                                     .WORD     T176CMP
                                     .WORD     MSGSTAT
2883 043544 104406      SET      TRAP      C$CLP1
2884 043546                                     END SUB
2885 043546 104403      L10062: TRAP      C$ESUB
2886 043550 005737 002222' TST      FATFLG        ;ANY FATAL ERRORS ?
2887 043554 001402 260$   BEQ      260$          ;BRANCH IF NOT
2888 043556 004737 017014' JSR      PC, CKDROP     ;TRY TO DROP THE UNIT
2889 043562 260$:
2890                                     .SBTTL  TEST 6: SUBTEST 5: FIFO Verify Input Ready
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913

```

TEST 6: SUBTEST 5:
SUBTEST DESCRIPTION:
This subtest verifies that writing 64. bytes into the FIFO without reading any out causes the Input Ready status to negate. The Subtest then verifies that writing a 65th byte into the FIFO causes the Data In Miss status to assert. Next it is verified that the original 64 bytes can be read out correctly and that the data has not been corrupted.
TEST STEPS:
BEGIN
Write to TSSR to soft initialize
Do a WRITE CHARACTERISTICS to setup a message buffer
Do a Write Subsystem WRITE NPR to set tape direction out
Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
Do a Write Subsystem READ STATUS
If Input Ready NOT=0 Then Print Error

```

2914      ;      If Output Ready NOT=1      Then Print Error
2915      ;      If Data In Miss NOT=0      Then Print Error
2916      ;      Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
2917      ;      Do a Write Subsystem READ STATUS
2918      ;      If Input Ready NOT=0      Then Print Error
2919      ;      If Output Ready NOT=1      Then Print Error
2920      ;      If Data In Miss NOT=1      Then Print Error
2921      ;      Do Write Subsystem READ FIFO
2922      ;      If Data read from FIFO NOT= to Data sent Then Print Error
2923      ;      Do a Write Subsystem READ STATUS
2924      ;      If Input Ready NOT=1      Then Print Error
2925      ;      If Output Ready NOT=0      Then Print Error
2926      ;      If Data In Miss NOT=1      Then Print Error
2927      ;      END
2928      ;      --
2929      043562      BGNSUB                      ;//////////////// BEGIN SUBTEST //////////////////
                043562                      T6.5:
                043562      104402                      TRAP      C#BSUB

2930      ;
2931      ;      Write to TSSR register to soft initialize the controller
2932      5$:      043564                      JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
                043564      004737      015604'      BCS      10$      ;BR IF SOFT INIT OKAY
2933      043570      103405                      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2934      043572      010001                      ERDF      ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
2935      043574                      ;
                043574      104455                      TRAP      C#ERDF
                043576      001165                      .WORD      629
                043600      003646'                      .WORD      SFIERR
                043602      011644'                      .WORD      SFIMSG

2937      ;      Do a WRITE CHARACTERISTICS to setup a message buffer
2938      10$:      043604      005037      002222'      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
2939      043610      012704      047740'      MOV      @T17PK2,R4      ;GET THE ADDRESS OF COMMAND PACKET
2940      043614      004737      010472'      JSR      PC,WRITCHR      ;DO WRITE CHARACTERISTICS COMMAND
2941      043620                      FORCERROR      42$      ;SSDFORCE ERROR IF FORCER=1
2942      043634      103407                      BCS      50$      ;BR IF CARRY SET (GOOD RETURN)
2943      043636      010001                      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2944      043640                      NEXT.ERRNO
2945      42$:      043640                      ERDF      ERRNO,T17SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                043640      104455                      TRAP      C#ERDF
                043642      001166                      .WORD      630
                043644      046365'                      .WORD      T17SSR
                043646      011656'                      .WORD      PKTSSR

2946      043650      005237      002222'      INC      FATFLG      ;SET FATAL ERROR FLAG
2947      043654      104406      50$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                043654                      TRAP      C#CLP1

2948      ;
2949      ;      Do a Write Subsystem WRITE NPR to set tape direction out
2950      100$:      043656      012700      000100      MOV      @NP.OUT,R0      ;SET TAPE DIRECTION OUT
2951      043662      004737      047566'      JSR      PC,T17SNPR      ;SETUP T17PK2 FOR WRITE NPR
2952      043666      012704      050110'      MOV      @T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
2953      043672      010465      000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
2954      043676      004737      016146'      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
2955      043702                      FORCERROR      102$      ;SSDFORCE ERROR IF FORCER=1
2956      043716      103407                      BCS      105$      ;BR IF CARRY SET (GOOD RETURN)
2957      043720      010001                      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2958      043722                      NEXT.ERRNO
2959      102$:      043722                      ERDF      ERRNO,T174SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET

```

Address	Offset	Hex Data	Assembly	Comments	Symbolic Data
043722	104455				TRAP C\$ERDF
043724	001167				.WORD 631
043726	046533				.WORD T174SSR
043730	011656				.WORD PKTSSR
2960	043732	005237 002222	INC FATFLG	;SET FATAL ERROR FLAG	
2961	043736	105\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	TRAP C\$CLP1
2962	043736	104406			
2963					
2964	043740	012737 000100 002310	Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern		
2965	043746	012701 046244	MOV #64.,COUNT	;WRITE 64 BYTES	
2966	043752	012702 000100	MOV #T17WFDATA,R1	;EXPD WRITE FIFO DATA BUFFER	
2967	043756	005000	MOV #64.,R2	;TEST PATTERN SIZE	
2968	043760	110021	CLR R0	;INCREMENT TEST PATTERN	
2969	043762	005200	110\$: MOV# R0,(R1)+	;STORE INCREMENT TEST BYTE	
2970	043764	005302	INC R0	;SET NEXT PATTERN	
2971	043766	003374	DEC R2	;DONE?	
2972	043770	013700 002310	BGT 110\$	;BR IF NO	
2973	043774	012701 046244	MOV COUNT,R0	;FIFO BYTE COUNT	
2974	044000	004737 047612	MOV #T17WFDATA,R1	;FIFO WRITE DATA ADDRESS	
2975	044004	012704 050110	JSR PC,T17WFI	;SETUP T17PK2 FOR WRITE FIFO	
2976	044010	010465 000000	MOV #T17PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
2977	044014	004737 016146	MOV R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
2978	044020		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET	
2979	044034	103407	FORCERROR 142\$	;GOODFORCE ERROR IF FORCER=1	
2980	044036	010001	BCS 150\$	;BR IF CARRY SET (GOOD RETURN)	
2981	044040		MOV R0,R1	;SAVE CONTENTS OF TSSR	
2982	044040		NEXT,ERRNO		
	044040	104455	142\$: ERRDF ERRNO,T175SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	TRAP C\$ERDF
	044042	001170			.WORD 632
	044044	046576			.WORD T175SSR
	044046	011656			.WORD PKTSSR
2983	044050	005237 002222	INC FATFLG	;SET FATAL ERROR FLAG	
2984	044054	150\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	TRAP C\$CLP1
2985	044054	104406			
2986					
2987			Do a Write Subsystem READ STATUS		
2988			If Input Ready NOT=0 Then Print Error		
2989			If Output Ready NOT=1 Then Print Error		
2990	044056	004737 047524	If Data In Miss NOT=0 Then Print Error		
2991	044062	012704 050110	JSR PC,T17SRD	;SETUP PACKET FOR READ STATUS	
2992	044066	010465 000000	MOV #T17PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
2993	044072	004737 016146	MOV R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
2994	044076		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET	
2995	044112	103407	FORCERROR 157\$	;GOODFORCE ERROR IF FORCER=1	
2996	044114	010001	BCS 160\$	;BR IF CARRY SET (GOOD RETURN)	
2997	044116		MOV R0,R1	;SAVE CONTENTS OF TSSR	
2998	044116		NEXT,ERRNO		
	044116	104455	157\$: ERRDF ERRNO,T173SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	TRAP C\$ERDF
	044120	001171			.WORD 633
	044122	046466			.WORD T173SSR
	044124	011656			.WORD PKTSSR
2999	044126	005237 002222	INC FATFLG	;SET FATAL ERROR FLAG	
3000	044132	160\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	TRAP C\$CLP1
3001	044132	104406			
			Set WORDS 0-7 of expd message buffer = to resv since not testing		

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 6: SUBTEST 5: FIFO VERIFY INPUT READY

SEQ 151

3002	044134	004737	047706'	JSR	PC,T17SETEXP	;SET WORDS 0-7 EXPD=RECV	
3003	044140	012701	046142'	MOV	#T17EXSTA,R1	;GET EXPECTED READ STATUS	
3004	044144	012702	050002'	MOV	#T17BFSTA,R2	;GET RECV READ STATUS	
3005	044150	012221		MOV	(R2)+,(R1)+	;SET EXPD WORD #8 = RECV TEMP	
3006	044152	011211		MOV	(R2),(R1)	;SET EXPD WORD #9 = RECV TEMP	
3007	044154	042711	000020	BIC	#S2.INRDY,(R1)	;SET EXP INPUT READY= 0	
3008	044160	052711	000040	BIS	#S2.OTRDY,(R1)	;SET EXP OUTPUT READY= 1	
3009	044164	042711	000200	BIC	#S2.DIM,(R1)	;SET EXP DATA IN MISS = 0	
3010	044170	005000		CLR	R0	;HIGH RECV ADDRESS FOR CKMSG2	
3011	044172	012701	047762'	MOV	#T17BFR,R1	;LOW RECV ADDRESS FOR CKMSG2	
3012	044176	012702	046122'	MOV	#T17EXP,R2	;EXPD ADDRESS	
3013	044202	012703	000024	MOV	#20,R3	;NUMBER OF BYTES TO COMPARE	
3014	044206	004737	011310'	JSR	PC,CKMSG2	;EXPD EQUAL RECV?	
3015	044212			FORCERROR	162\$,NOTSSR	;000	
3016	044222	103404		BCS	170\$	;BR IF YES	
3017	044224			NEXT.ERRNO			
3018	044224			162\$:	ERRHRD ERRNO,T173CMP,MSGSTAT	;REPORT ERROR	
	044224	104456					TRAP C\$ERHRD
	044226	001172					.WORD 634
	044230	047063'					.WORD T173CMP
	044232	012160'					.WORD MSGSTAT
3019	044234			170\$:	CKLOOP	;LOOP ON ERROR. IF FLAG	SET
	044234	104406					TRAP C\$CLP1
3020							
3021							
3022							
3023	044236	012700	000001		Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written		
3024	044242	012701	046244'	MOV	#1,R0	;FIFO RYTE COUNT	
3025	044246	004737	047612'	MOV	#T17WFDATA,R1	;FIFO WRITE DATA ADDRESS	
3026	044252	012704	050110'	JSR	PC,T17WIF	;SETUP T17PK2 FOR WRITE FIFO	
3027	044256	010465	000000	MOV	#T17PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
3028	044262	004737	016146'	MOV	R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
3029	044266			JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET	
3030	044302	103407		FORCERROR	172\$	;000FORCE ERROR IF FORCER=1	
3031	044304	010001		BCS	180\$	;BR IF CARRY SET (GOOD RETURN)	
3032	044306			MOV	R0,R1	;SAVE CONTENTS OF TSSR	
3033	044306			NEXT.ERRNO			
	044306	104455		172\$:	ERRDF ERRNO,T175SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	
	044310	001173					TRAP C\$ERDF
	044312	046576'					.WORD 635
	044314	011656'					.WORD T175SSR
	044314						.WORD PKTSSR
3034	044316	005237	002222'	INC	FATFLG	;SET FATAL ERROR FLAG	
3035	044322			180\$:	CKLOOP	;LOOP ON ERROR. IF FLAG	SET
	044322	104406					TRAP C\$CLP1
3036							
3037							
3038					Do a Write Subsystem READ STATUS		
3039					If Input Ready NOT=0 Then Print Error		
3040					If Output Ready NOT=1 Then Print Error		
					If Data In Miss NOT=1 Then Print Error		
3041	044324	004737	047524'	JSR	PC,T17SRD	;SETUP PACKET FOR READ STATUS	
3042	044330	012704	050110'	MOV	#T17PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
3043	044334	010465	000000	MOV	R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
3044	044340	004737	016146'	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET	
3045	044344			FORCERROR	187\$	;000FORCE ERROR IF FORCER=1	
3046	044360	103407		BCS	190\$	;BR IF CARRY SET (GOOD RETURN)	
3047	044362	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR	
3048	044364			NEXT.ERRNO			

J12

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 6: SUBTEST 5: FIFO VERIFY INPUT READY

SEQ 152

```

3049 044364      187$:  ERRDF  ERRNO,T173SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
      044364 104455                                     TRAP  C$ERDF
      044366 001174                                     .WORD  636
      044370 046466'                                     .WORD  T173SSR
      044372 011656'                                     .WORD  PKTSSR
3050 044374 005237 002222'
3051 044400      190$:  INC    FATFLG  ;SET FATAL ERROR FLAG
      044400 104406      CKLOOP  ;LOOP ON ERROR, IF FLAG SET
                                     TRAP  C$CLP1
3052 ;      Set WORDS 0-7 of expd message buffer = to recv since not testing
3053 044402 004737 047706'  JSR    PC,T17SETEXP  ;SET WORDS 0-7 EXPD-RECV
3054 044406 012701 046142'  MOV    #T17EXSTA,R1  ;GET EXPECTED READ STATUS
3055 044412 012702 050002'  MOV    #T17BFSTA,R2  ;GET RECV READ STATUS
3056 044416 012221          MOV    (R2)+,(R1)+  ;SET EXPD WORD #8 = RECV TEMP
3057 044420 011211          MOV    (R2),(R1)    ;SET EXPD WORD #9 = RECV TEMP
3058 044422 042711 000020  BIC    #S2.INRDY,(R1)  ;SET EXP INPUT READY= 0
3059 044426 052711 000040  BIS    #S2.OUTRDY,(R1)  ;SET EXP OUTPUT READY= 1
3060 044432 052711 000200  BIS    #S2.DIM,(R1)    ;SET EXP DATA IN MISS = 1
3061 044436 005000          CLR    R0          ;HIGH RECV ADDRESS FOR CKMSG2
3062 044440 012701 047762'  MOV    #T17BFR,R1    ;LOW RECV ADDRESS FOR CKMSG2
3063 044444 012702 046122'  MOV    #T17EXP,R2    ;EXPD ADDRESS
3064 044450 012703 000024  MOV    #20,R3          ;NUMBER OF BYTES TO COMPARE
3065 044454 004737 011310'  JSR    PC,CKMSG2     ;EXPD EQUAL RECV?
3066 044460          FORCERROR 192$,NOTSSR  ;BAD
3067 044470 103404          BCS    200$      ;BR IF YES
3068 044472          NEXT.ERRNO
3069 044472      192$:  ERRHRD  ERRNO,T173CMP,MSGSTAT ;REPORT ERROR
      044472 104456                                     TRAP  C$ERHRD
      044474 001175                                     .WORD  637
      044476 047063'                                     .WORD  T173CMP
      044500 012160'                                     .WORD  MSGSTAT
3070 044502      200$:  CKLOOP  ;LOOP ON ERROR, IF FLAG SET
      044502 104406      TRAP  C$CLP1
3071 ;      Do Write Subsystem READ FIFO
3072 044504 013700 002310'  MOV    COUNT,R0      ;SET READ BYTE COUNT
3073 044510 004737 047646'  JSR    PC,T17RFIF    ;SETUP T17PK2 FOR READ FIFO
3074 044514 012704 050110'  MOV    #T17PK2,R4    ;GET WRITE SUBSYSTEM COMMAND PACKET
3075 044520 010465 000000  MOV    R4,TSD8(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
3076 044524 004737 016146'  JSR    PC,CHKTSSR    ;WAIT FOR SSR TO SET
3077 044530          FORCERROR 212$      ;BAD/FORCE ERROR IF FORCER=1
3078 044544 103407          BCS    220$      ;BR IF CARRY SET (GOOD RETURN)
3079 044546 010001          MOV    R0,R1      ;SAVE CONTENT OF TSSR
3080 044550          NEXT.ERRNO
3081 044550      212$:  ERRDF  ERRNO,T176SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      044550 104455                                     TRAP  C$ERDF
      044552 001176                                     .WORD  639
      044554 046642'                                     .WORD  T176SSR
      044556 011656'                                     .WORD  PKTSSR
3082 044560 005237 002222'
3083 044564      220$:  INC    FATFLG  ;SET FATAL ERROR FLAG
      044564 104406      CKLOOP  ;LOOP ON ERROR, IF FLAG SET
                                     TRAP  C$CLP1
3084 ;
3085 ;      If Data read from FIFO NOT= to Data sent then Print Error
3086 044566 005000          CLR    R0          ;HIGH RECV ADDRESS FOR CKMSG2
3087 044570 012702 046244'  MOV    #T17WFDATA,R2  ;GET EXPECTED ADDRESS FOR CKMSG2
3088 044574 012701 050002'  MOV    #T17BFSTA,R1  ;GET RECEIVED ADDRESS FOR CKMSG2
3089 044600 013703 002310'  MOV    COUNT,R3      ;NUMBER OF BYTES TO COMPARE
3090 044604 004737 011310'  JSR    PC,CKMSG2     ;EXPD EQUAL RECV?

```

K12

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 6: SUBTEST 5: FIFO VERIFY INPUT READY

SEQ 153

```

3091 044610          FORCERROR      232$,NOTSSR      ;BDD
3092 044620 103406    BCS          240$             ;BR IF YES
3093 044622          NEXT.ERRNO
3094 044622 013701 002310' 232$: MOV      COUNT,R1      ;GET BYTE COUNT
3095 044626          ERRHRD  ERRNO,T175CMP,FIFEXP    ;REPORT ERROR
          044626 104456          TRAP      C$ERHRD
          044630 001177          .WORD    639
          044632 047232'          .WORD    T175CMP
          044634 012000'          .WORD    FIFEXP
3096 044636          240$: CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
          044636 104406          TRAP      C$CLP1
3097
3098          ; Do a Write Subsystem READ STATUS
3099          ; If Input Ready NOT=1 Then Print Error
3100          ; If Output Ready NOT=0 Then Print Error
3101          ; If Data In Miss NOT=1 Then Print Error
3102 044640 004737 047524' JSR      PC,T17SRD      ;SETUP PACKET FOR READ STATUS
3103 044644 012704 050110' MOV      @T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
3104 044650 010465 000000' MOV      R4,TSD8(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
3105 044654 004737 016146' JSR      PC,CHKTSSR    ;WAIT FOR SSR TO SET
3106 044660          FORCERROR      252$             ;BDDFORCE ERROR IF FORCER=1
3107 044674 103407    BCS          260$             ;BR IF CARRY SET (GOOD RETURN)
3108 044676 010001    MOV      R0,R1                ;SAVE CONTENTS OF TSSR
3109 044700          NEXT.ERRNO
3110 044700          252$: ERRDF  ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
          044700 104455          TRAP      C$ERDF
          044702 001200          .WORD    640
          044704 046466'          .WORD    T173SSR
          044706 011656'          .WORD    PKTSSR
3111 044710 005237 002222' 260$: INC      FATFLG      ;SET FATAL ERROR FLAG
3112 044714          260$: CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
          044714 104406          TRAP      C$CLP1
3113          ; Set WORDS 0-7 of expd message buffer = to recv since not testing
3114 044716 004737 047706' JSR      PC,T17SETEXP ;SET WORDS 0-7 EXPD-RECV
3115 044722 012701 046142' MOV      @T17EXSTA,R1    ;SET EXPECTED READ STATUS
3116 044726 012702 050002' MOV      @T17BFSTA,R2    ;GET RECV READ STATUS
3117 044732 012221    MOV      (R2)+,(R1)+          ;SET EXPD WORD #8 = RECV TEMP
3118 044734 011211    MOV      (R2),(R1)            ;SET EXPD WORD #9 = RECV TEMP
3119 044736 052711 000020  BIS      @S2.INRDY,(R1)    ;SET EXP INPUT READY= 1
3120 044742 042711 000040  BIC      @S2.OUTPUT,(R1)   ;SET EXP OUTPUT READY= 0
3121 044746 052711 000200  BIS      @S2.DIM,(R1)      ;SET EXP DATA IN MISS = 1
3122 044752 005000    CLR      R0                  ;HIGH RECV ADDRESS FOR CKMSG2
3123 044754 012701 047762' MOV      @T17BFR,R1      ;LOW RECV ADDRESS FOR CKMSG2
3124 044760 012702 046122' MOV      @T17EXP,R2      ;EXPD ADDRESS
3125 044764 012703 000024  MOV      @20,R3          ;NUMBER OF BYTES TO COMPARE
3126 044770 004737 011310' JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
3127 044774          FORCERROR      272$,NOTSSR      ;BDD
3128 045004 103404    BCS          280$             ;BR IF YES
3129 045006          NEXT.ERRNO
3130 045006          272$: ERRHRD  ERRNO,T174CMP,MSGSTAT ;REPORT ERROR
          045006 104456          TRAP      C$ERHRD
          045010 001201          .WORD    641
          045012 047147'          .WORD    T174CMP
          045014 012160'          .WORD    MSGSTAT
3131 045016          280$: CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
          045016 104406          TRAP      C$CLP1
3132

```

L12

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 6: SUBTEST 5: FIFO VERIFY INPUT READY

SEQ 154

```

3133 045020                                ENDSUB                                ;////////// END SUBTEST //////////
                                           L10063:                                TRAP      C$ESUB
                                           045020 104403
3134
3135 045022 005737 002222'                TST      FATFLG                ;ANY FATAL ERRORS ?
3136 045026 001402                        BEQ      300$                ;BRANCH IF NOT
3137 045030 004737 017014'                JSR      PC,CKDROP            ;TRY TO DROP THE UNIT
3138 045034                                300$:
3139
3140                                .SBTTL TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test
3141
3142                                ;**
3143                                ; TEST 6: SUBTEST 6:
3144                                ;
3145                                ; SUBTEST DESCRIPTION:
3146                                ;
3147                                ; This subtest verifies that the Reset FIFO function within
3148                                ; the Write Miscellaneous Control 1 function initializes
3149                                ; the FIFO to correct initial status. The following steps
3150                                ; are performed:
3151                                ; 1. Reset an already initialized FIFO and check for
3152                                ; proper status.
3153                                ; 2. Write a varying number of bytes (1-65.) into the
3154                                ; FIFO and verify that after each block of bytes is
3155                                ; written the FIFO can be reset to it's initial
3156                                ; state.
3157                                ;
3158                                ; TEST STEPS:
3159                                ;
3160                                ; BEGIN
3161                                ; Write to TSSR to soft initialize
3162                                ; Do a WRITE CHARACTERISTICS to setup a message buffer
3163                                ; Do a Write Subsystem Write Misc to Reset FIFO
3164                                ; Do a Write Subsystem READ STATUS
3165                                ; If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
3166                                ; signals NOT=0 Then Print Error
3167                                ; Do a Write Subsystem WRITE NPR to set tape direction out
3168                                ;
3169                                ; REPEAT FOR BYTE COUNT 1 TO 65.
3170                                ; BEGIN
3171                                ; Do a Write Subsystem WRITE FIFO with the current byte count
3172                                ; Do a Write Subsystem Write Misc to Reset FIFO
3173                                ; Do a Write Subsystem READ STATUS
3174                                ; If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
3175                                ; signals NOT=0 Then Print Error
3176                                ;
3177                                ; END
3178 045034                                ;--                                ;////////// BEGIN SUBTEST //////////
                                           045034                                T6.6:                                TRAP      C$BSUB
                                           045034 104402
3179
3180                                ;
3181 045036                                5$:                                Write to TSSR register to soft initialize the controller
3182 045036 004737 015604'                JSR      PC,SOFINIT            ;WRITE TO TSSR TO SOFT INITIALIZE
3183 045042 103405                        BCS      10$                ;BR IF SOFT INIT OKAY
3184 045044 010001                        MOV      R0,R1                ;SAVE CONTENTS OF TSSR
3185 045046                                ERRDF   ERRNO,SF1ERR,SF1MSG        ;DEVICE FATAL DURING INIT

```

M12

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 6: SUBTEST 6: FIFO VERIFY RESET FIFO TEST

SEQ 155

045046	104455					TRAP	C\$ERDF
045050	001201					.WORD	641
045052	003646'					.WORD	SFIERR
045054	011644'					.WORD	SFIMSG
3186							
3187	045056	005037	002222'	10\$:	Do a WRITE CHARACTERISTICS to setup a message buffer		
3188	045062	012704	047740'		CLR FATFLG		;CLEAR FATAL ERROR FLAG
3189	045066	004737	010472'		MOV #T17PACKET,R4		;GET THE ADDRESS OF COMMAND PACKET
3190	045072				JSR PC,WR1CHR		;DO WRITE CHARACTERISTICS COMMAND
3191	045106	103407			FORCERROR 42\$		;FORCE ERROR IF FORCER=1
3192	045110	010001			BCS 50\$		;BR IF CARRY SET (GOOD RETURN)
3193	045112				MOV R0,R1		;SAVE CONTENTS OF TSSR
3194	045112			42\$:	NEXT,ERRNO		
	045112	104455			ERRDF ERRNO,T17SSR,PKTSSR		;DEVICE FATAL SSR FAILED TO SET
	045114	001202				TRAP	C\$ERDF
	045116	046365'				.WORD	642
	045120	011656'				.WORD	T17SSR
						.WORD	PKTSSR
3195	045122	005237	002222'		INC FATFLG		;SET FATAL ERROR FLAG
3196	045126			50\$:	CKLOOP		;LOOP ON ERROR, IF FLAG SET
	045126	104406				TRAP	C\$CLP1
3197					Do a Write Subsystem Write Misc to Reset FIFO		
3198	045130	004737	047544'		JSR PC,T17RSFIF		;SETUP PKT FOR WRITE MISC RESET FIFO
3199	045134	012704	050110'		MOV #T17PK2,R4		;GET WRITE SUBSYSTEM COMMAND PACKET
3200	045140	010465	000000		MOV R4,TSDB(R5)		;SET THE PACKET ADDRESS TO EXECUTE
3201	045144	004737	016146'		JSR PC,CHKTSSR		;WAIT FOR SSR TO SET
3202	045150				FORCERROR 62\$		;FORCE ERROR IF FORCER=1
3203	045164	103407			BCS 70\$		;BR IF CARRY SET (GOOD RETURN)
3204	045166	010001			MOV R0,R1		;SAVE CONTENTS OF TSSR
3205	045170				NEXT,ERRNO		
3206	045170			62\$:	ERRDF ERRNO,T172SSR,PKTSSR		;DEVICE FATAL SSR FAILED TO SET
	045170	104455				TRAP	C\$ERDF
	045172	001203				.WORD	643
	045174	046422'				.WORD	T172SSR
	045176	011656'				.WORD	PKTSSR
3207	045200	005237	002222'		INC FATFLG		;SET FATAL ERROR FLAG
3208	045204			70\$:	CKLOOP		;LOOP ON ERROR, IF FLAG SET
	045204	104406				TRAP	C\$CLP1
3209					Do a Write Subsystem READ STATUS		
3210					If all Tape Status 2 (ICER,IFMK,IHER) flip-flop		
3211					signals NOT=0 Then Print Error		
3212							
3213	045206	004737	047524'		JSR PC,T17SRD		;SETUP PACKET FOR READ STATUS
3214	045212	012704	050110'		MOV #T17PK2,R4		;GET WRITE SUBSYSTEM COMMAND PACKET
3215	045216	010465	000000		MOV R4,TSDB(R5)		;SET THE PACKET ADDRESS TO EXECUTE
3216	045222	004737	016146'		JSR PC,CHKTSSR		;WAIT FOR SSR TO SET
3217	045226				FORCERROR 77\$		;FORCE ERROR IF FORCER=1
3218	045242	103407			BCS 80\$		;BR IF CARRY SET (GOOD RETURN)
3219	045244	010001			MOV R0,R1		;SAVE CONTENTS OF TSSR
3220	045246				NEXT,ERRNO		
3221	045246			77\$:	ERRDF ERRNO,T173SSR,PKTSSR		;DEVICE FATAL SSR FAILED TO SET
	045246	104455				TRAP	C\$ERDF
	045250	001204				.WORD	644
	045252	046466'				.WORD	T173SSR
	045254	011656'				.WORD	PKTSSR
3222	045256	005237	002222'		INC FATFLG		;SET FATAL ERROR FLAG
3223	045262			80\$:	CKLOOP		;LOOP ON ERROR, IF FLAG SET
	045262	104406				TRAP	C\$CLP1

N12

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 6: SUBTEST 6: FIFO VERIFY RESET FIFO TEST

SEQ 156

```

3224 045264 004737 047706' JSR PC,T17SETEXP ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
3225 045270 012701 046142' MOV #T17EXSTA,R1 ;GET EXPECTED READ STATUS
3226 045274 012702 050002' MOV #T17BFSTA,R2 ;GET RECV READ STATUS
3227 045300 011211 (R2),(R1) ;SET EXPD WORD #8 = RECV TEMP
3228 045302 042711 002000 BIC #S1.ICER,(R1) ;SET EXPD ICER =0
3229 045306 042711 001000 BIC #S1.IFMK,(R1) ;SET EXPD IFMK =0
3230 045312 042711 000400 BIC #S1.IHER,(R1) ;SET EXPD IHER =0
3231 045316 016261 000002 000002 MOV 2(R2),2(R1) ;SET EXPD WORD #9 = RECV (NOT TESTING)
3232 045324 005000 CLR R0 ;HIGH RECV ADDRESS FOR CKMSG2
3233 045326 012701 047762' MOV #T17BFR,R1 ;LOW RECV ADDRESS FOR CKMSG2
3234 045332 012702 046122' MOV #T17EXP,R2 ;EXPD ADDRESS
3235 045336 012703 000024 MOV #20.,R3 ;NUMBER OF BYTES TO COMPARE
3236 045342 004737 011310' JSR PC,CKMSG2 ;EXPD EQUAL RECV?
3237 045346 FORCERROR 92$,NOTSSR ;000
3238 045356 103404 BCS 100$ ;BR IF YES
3239 045360 NEXT.ERRNO
3240 045360 92$: ERRHRD ERRNO,T177CMP,MSGSTAT ;REPORT ERROR
                                TRAP C$ERHRD
                                .WORD 645
                                .WORD T177CMP
                                .WORD MSGSTAT
                                TRAP C$CLP1
                                3241 045370 104406 100$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                3242 045370 104406 104406 TRAP C$CLP1
                                3243 ; Do a Write Subsystem WRITE NPR to set tape direction out
                                3244 045372 012700 000100 MOV #NPR,OUT,R0 ;SET TAPE DIRECTION OUT
                                3245 045376 004737 047566' JSR PC,T17SNPR ;SETUP T17PK2 FOR WRITE NPR
                                3246 045402 012704 050110' MOV #T17PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
                                3247 045406 010465 000000 MOV R4,TSD8(R5) ;SET THE PACKET ADDRESS TO EXECUTE
                                3248 045412 004737 016146 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
                                3249 045416 FORCERROR 112$ ;000FORCE ERROR IF FORCER=1
                                3250 045432 103407 BCS 120$ ;BR IF CARRY SET (GOOD RETURN)
                                3251 045434 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
                                3252 045436 NEXT.ERRNO
                                3253 045436 112$: ERRDF ERRNO,T174SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP C$ERDF
                                .WORD 646
                                .WORD T174SSR
                                .WORD PKTSSR
                                3254 045446 005237 002222' 120$: INC FATFLG ;SET FATAL ERROR FLAG
                                3255 045452 104406 120$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                TRAP C$CLP1
                                3256 ; Setup incrementing pattern in FIFO data buffer
                                3257 ;
                                3258 045454 012701 046142' MOV #T17EXSTA,R1 ;EXPD WRITE FIFO DATA BUFFER
                                3259 045460 012702 000100 MOV #64.,R2 ;TEST PATTERN SIZE
                                3260 045464 005000 CLR R0 ;INCREMENT TEST PATTERN
                                3261 045466 110021 130$: MOV8 R0,(R1)+ ;STORE INCREMENT TEST BYTE
                                3262 045470 005200 INC R0 ;SET NEXT PATTERN
                                3263 045472 005302 DEC R0 ;DONE?
                                3264 045474 003374 BGT 130$ ;BR IF NO
                                3265 ;
                                3266 ; REPEAT FOR BYTE COUNT 1 TO 65.
                                3267 045476 012737 000001 002310' MOV #1,COUNT ;GET FIRST BYTE COUNT
                                3268 ; Do a Write Subsystem WRITE FIFO with the current byte count
                                3269 045504 150$: MOV COUNT,R0 ;REPEAT LOOP LABEL
                                3270 045504 013700 002310' MOV COUNT,R0 ;FIFO BYTE COUNT

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 6: SUBTEST 6: FIFO VERIFY RESET FIFO TEST

SEQ 157

3271	045510	012701	046142'	MOV	#T17EXSTA,R1	;FIFO WRITE DATA ADDRESS
3272	045514	004737	047612'	JSR	PC,T17WFIF	;SETUP T17PK2 FOR WRITE FIFO
3273	045520	012704	050110'	MOV	#T17PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
3274	045524	010465	000000	MOV	R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE
3275	045530	004737	016146'	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
3276	045534			FORCERROR	152#	;GOODFORCE ERROR IF FORCER=1
3277	045550	103407		BCS	160#	;BR IF CARRY SET (GOOD RETURN)
3278	045552	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR
3279	045554			NEXT,ERRNO		
3280	045554			ERRDF	ERRNO,T175SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	045554	104455				TRAP C#ERDF
	045556	001207				.WORD 647
	045560	046576'				.WORD T175SSR
	045562	011656'				.WORD PKTSSR
3281	045564	005237	002222'	INC	FATFLG	;SET FATAL ERROR FLAG
3282	045570			CKLOOP		;LOOP ON ERROR, IF FLAG SET
	045570	104406				TRAP C#CLP1
3283						
3284					Do a Write Subsystem Write Misc to Reset FIFO	
3285	045572	004737	047544'	JSR	PC,T17RSI1'	;SETUP PKT FOR WRITE MISC RESET FIFO
3286	045576	012704	050110'	MOV	#T17PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
3287	045602	010465	000000	MOV	R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE
3288	045606	004737	016146'	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
3289	045612			FORCERROR	162#	;GOODFORCE ERROR IF FORCER=1
3290	045626	103407		BCS	170#	;BR IF CARRY SET (GOOD RETURN)
3291	045630	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR
3292	045632			NEXT,ERRNO		
3293	045632			ERRDF	ERRNO,T172SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	045632	104455				TRAP C#ERDF
	045634	001210				.WORD 648
	045636	046422'				.WORD T172SSR
	045640	011656'				.WORD PKTSSR
3294	045642	005237	002222'	INC	FATFLG	;SET FATAL ERROR FLAG
3295	045646			CKLOOP		;LOOP ON ERROR, IF FLAG SET
	045646	104406				TRAP C#CLP1
3296						
3297					Do a Write Subsystem READ STATUS	
3298					If all Tape Status 2 (ICER,IFMK,INER) flip-flop	
3299					signals NOT=0 Then Print Error	
3300	045650	004737	047524'	JSR	PC,T17SRD	;SETUP PACKET FOR READ STATUS
3301	045654	012704	050110'	MOV	#T17PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
3302	045660	010465	000000	MOV	R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE
3303	045664	004737	016146'	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
3304	045670			FORCERROR	177#	;GOODFORCE ERROR IF FORCER=1
3305	045704	103407		BCS	180#	;BR IF CARRY SET (GOOD RETURN)
3306	045706	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR
3307	045710			NEXT,ERRNO		
3308	045710			ERRDF	ERRNO,T173SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	045710	104455				TRAP C#ERDF
	045712	001211				.WORD 649
	045714	046466'				.WORD T173SSR
	045716	011656'				.WORD PKTSSR
3309	045720	005237	002222'	INC	FATFLG	;SET FATAL ERROR FLAG
3310	045724			CKLOOP		;LOOP ON ERROR, IF FLAG SET
	045724	104406				TRAP C#CLP1
3311	045726	004737	047706'	JSR	PC,T17SEIEXP	;SET WORDS 0-7 EXPQ-RECV (NOT TESTING)
3312	045732	012701	046142'	MOV	#T17EXSTA,R1	;GET EXPECTED READ STATUS

C13

TSV5 - HARDWARE TESTS MACRO M1113 01 FEB 84 17102
TEST 6: SUBTEST 6: FIFO VERIFY RESET FIFO TEST

SEQ 158

3313	045736	012702	050002'	MOV	#T17BFSTA,R2	GET RECV READ STATUS		
3314	045742	011211		MOV	(R2),(R1)	SET EXPD WORD #8 - RECV TEMP		
3315	045744	042711	002000	BIC	#S1,ICER,(R1)	SET EXPD ICER =0		
3316	045750	042711	001000	BIC	#S1,IFMK,(R1)	SET EXPD IFMK =0		
3317	045754	042711	000400	BIC	#S1,IHER,(R1)	SET EXPD IHER =0		
3318	045760	016261	000002 000002	MOV	2(R2),2(R1)	SET EXPD WORD #9 - RECV (NOT TESTING)		
3319	045766	005000		CLR	R0	HIGH RECV ADDRESS FOR CKMSG2		
3320	045770	012701	047762'	MOV	#T17BFR,R1	LOW RECV ADDRESS FOR CKMSG2		
3321	045774	012702	046122'	MOV	#T17EXP,R2	EXPD ADDRESS		
3322	046000	012703	000024	MOV	#20,,R3	NUMBER OF BYTES TO COMPARE		
3323	046004	004737	011310'	JSR	PC,CKMSG2	EXPD EQUAL RECV?		
3324	046010			FORCERROR	192\$,NOTSSR	0000		
3325	046020	103404		BCS	200\$	BR IF YES		
3326	046022			NEXT.ERRNO				
3327	046022			ERRHRD	ERRNO,T177CMP,MSGSTAT	REPORT ERROR		
	046022	104456					TRAP	C\$ERRHRD
	046024	001212					.WORD	650
	046026	047414'					.WORD	T177CMP
	046030	012160'					.WORD	MSGSTAT
3328	046032			200\$:	CKLOOP	LOOP ON ERROR, IF FLAG SET	SET	
	046032	104406					TRAP	C\$CLP1
3329								
3330								
3331	046034			250\$:	FORCEXIT			
3332	046034				260\$			
3333	046044	005237	002310'	INC	COUNT	GET NEXT BYTE COUNT		
3334	046050	023727	002310' 000101	CMP	COUNT,065,	DONE ALL BYTES?		
3335	046056	101002		BHI	260\$	BR IF YES		
3336	046060	000137	045504'	JMP	150\$	DO ANOTHER BYTE COUNT		
3337	046064			260\$:				
3338								
3339	046064			ENDSUB		////////// END SUBTEST //////////		
	046064					L10064;		
	046064	104403					TRAP	C\$ESUB
3340								
3341	046066	005737	002222'	TST	FATFLG	ANY FATAL ERRORS ?		
3342	046072	001402		BEQ	300\$	BRANCH IF NOT		
3343	046074	004737	017014'	JSR	PC,CKDROP	TRY TO DROP THE UNIT		
3344	046100	004737	016270'	300\$:	JSR	PC,TSTLOOP		
3345	046104	103002		BCC	305\$	DO ITERATIONS?		
3346	046106	000137	040144'	JMP	T17LOOP	BR IF NO		
3347	046112			305\$:		LOOP UNTIL ITERATIONS DONE		
3348								
3349	046112			EXIT	TST	////////// EXIT TEST //////////		
	046112	104432					TRAP	C\$EXIT
	046114	002112					.WORD	L10056-
3350								
3351								
3352								
3353								
3354								
3355								
3356	046116			T17MSK:		NUMBER OF UNTESTED BITS IN READ STATUS BYTES		
3357						UNTESTED BITS ARE SET TO 1		
3358	046116	377		.BYTE	%C<000>	BYTE 0 MASK		
3359	046117	037		.BYTE	%C<340>	BYTE 1 MASK (PARERR,IRESV2,IRESV1)		
3360	046120	360		.BYTE	%C<017>	BYTE 2 (TIMER A,TIMER B,UNDEFINED<1;0>)		

```

3361 046121      000          .BYTE      0          ;MAKE IT EVEN
3362
3363 046122          T17EXP:          ;BEGIN EXPECTED DATA BUFFER
3364 046122      000000          .WORD      0          ;MESSAGE TYPE
3365 046124      000000          .WORD      0          ;DATA FIELD LENGTH
3366 046126      000000          .WORD      0          ;RBPGR
3367 046130      000000          .WORD      0          ;XST0
3368 046132      000000          .WORD      0          ;XST1
3369 046134      000000          .WORD      0          ;XST2
3370 046136      000000          .WORD      0          ;XST3
3371 046140      000000          .WORD      0          ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
3372 046142          T17EXSTA: .BLKB 66.          ;EXPECTED READ STATUS AND WRITE FIFO DATA
3373 046244          T17EXEND:          ;END EXPECTED DATA BUFFER
3374
3375 046244          T17WFDATA: .BLKB 66.          ;WRITE FIFO EXPECTED DATA BUFFER
3376
3377
3378          ;
3379          ;LOCAL TEXT MESSAGES FOR TEST
3380          ;
3381 046346      106      111      106      TST17ID:          .ASCIZ      'FIFO Exerciser'
3382 046365      127      122      111      T17SSR: .ASCIZ      'WRITE CHARACTERISTICS Failed'
3383 046422      127      122      111      T172SSR: .ASCIZ      'WRITE SUBSYSTEM (Write Misc) Failed'
3384 046466      127      122      111      T173SSR: .ASCIZ      'WRITE SUBSYSTEM (Read Status) Failed'
3385 046533      127      122      111      T174SSR: .ASCIZ      'WRITE SUBSYSTEM (Write Npr) Failed'
3386 046576      127      122      111      T175SSR: .ASCIZ      'WRITE SUBSYSTEM (Write FIFO) Failed'
3387 046642      127      122      111      T176SSR: .ASCIZ      'WRITE SUBSYSTEM (Read FIFO) Failed'
3388 046705      106      111      106      T171CMP: .ASCIZ      'FIFO Status in WORD #9 Incorrect after Initialize'
3389 046767      122      145      141      T172CMP: .ASCIZ      'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
3390 047063      106      111      106      T173CMP: .ASCIZ      'FIFO Status (In WORD #9) Incorrect after WRITE FIFO'
3391 047147      106      111      106      T174CMP: .ASCIZ      'FIFO Status (In WORD #9) Incorrect after READ FIFO'
3392 047232      122      145      141      T175CMP: .ASCIZ      'Read FIFO Data not equal to Write FIFO Data'
3393 047306      106      111      106      T176CMP: .ASCIZ      'FIFO Status (In WORD #9) Incorrect after READ FIFO from an Empty FIFO'
3394 047414      106      111      106      T177CMP: .ASCIZ      'FIFO Status (In WORD #9) Incorrect after RESET FIFO'
3395          .EVEN
3396
3397
3398          ;
3399          ; CLEAR MESSAGE BUFFER
3400
3401 047500          T17CLRBUF:
3402 047500          SAVREG
3403 047504      012701      047762'      MOV      #T17BFR,R1          ;SAVE R1-R5 UNTIL NEXT RETURN
3404 047510      012702      000120'      MOV      #T17BEND-T17BFR,R2      ;GET MESSAGE BUFFER ADDRESS
3405 047514      105021          10#:      CLRB      (R1),          ;SIZE OF MESSAGE BUFFER IN BYTES
3406 047516      005302          DEC      R2          ;CLEAR A BYTE
3407 047520      003375          BGT      10#          ;DONE?
3408 047522      000207          RTS      PC          ;BR IF NO
3409          ;
3410          ; SETUP T17PK2 PACKET FOR READ STATUS
3411          ;
3412 047524      004737      047500'      T17SRD:      JSR      PC,T17CLRBUF          ;CLEAR MESSAGE BUFFER
3413 047530      012700      050120'      MOV      #T17DT2,R0          ;WRITE SUBSYSTEM DATA BUFFER
3414 047534      112720      000005'      MOVB     #PW_RDSTATUS,(R0).      ;STORE READ STATUS COMMAND IN BSEL0
3415 047540      105010          CLRB      (R0)          ;CLEAR BSEL1
3416 047542      000207          RTS      PC          ;RETURN

```

```

3418
3419
3420      ;*
3421      ; SETUP T17PK2 PACKET FOR WRITE MISC RESET FIFO
3422      ;*
3422 047544      T17RSFIF:
3423 047544      JSR      PC,T17CLRBUF      ;CLEAR MESSAGE BUFFER
3424 047550      MOV      @T17DT2,R0      ;WRITE SUBSYSTEM DATA BUFFER
3425 047554      MOV      @PW.WMISC,(R0)+  ;STORE WRITE MISCELLANEOUS IN BSEL0
3426 047560      MOV      @MS.RSFIF!MS.RSTAP,(R0) ;STORE BSEL1 CLEAR FIFO CODES
3427 047564      RTS      PC              ;RETURN
3428
3429
3430      ;*
3431      ; SETUP T17PK2 PACKET FOR WRITE NPR
3432      ;*
3432      ; INPUT:
3433      ;      RO CONTAINS BSEL1 NPR DATA
3434      ;
3435      ;      SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
3436      ;*
3437 047566      T17SNPR:
3438 047566      JSR      PC,T17CLRBUF      ;CLEAR MESSAGE BUFFER
3439 047572      MOV      @T17DT2,R1      ;WRITE SUBSYSTEM DATA BUFFER
3440 047576      MOV      @PW.WNPR,(R1)+  ;STORE WRITE NPR IN BSEL0
3441 047602      BIS      @NP.WRP,R0      ;DON'T WRITE WRONG PARITY
3442 047606      MOV      RO,(R1)          ;STORE NPR DATA IN BSEL1
3443 047610      RTS      PC              ;RETURN
3444
3445
3446      ;*
3447      ; SETUP T17PK2 PACKET FOR WRITE FIFO
3448      ;*
3448      ; INPUT:
3449      ;      RO CONTAINS BYTE COUNT
3450      ;      R1 CONTAINS DATA PATTERN BLOCK ADDRESS
3451      ;*
3452 047612      T17WFIF:
3453 047612      SAVREG
3454 047616      JSR      PC,T17CLRBUF      ;SAVE R1-R5 UNTIL NEXT RETURN
3455 047622      MOV      @T17DT2,R2      ;CLEAR MESSAGE BUFFER
3456 047626      MOV      @PW.WFIFO,(R2)+  ;WRITE SUBSYSTEM DATA BUFFER
3457 047632      MOV      RO,(R2)+  ;STORE WRITE FIFO IN BSEL0
3458 047634      CLR      (R2)+  ;STORE BYTE COUNT IN BSEL1
3459 047636      MOV      (R1)+,(R2)+  ;CLEAR SEL2 (UNUSED)
3460 047640      DEC      RO              ;STORE DATA PATTERN BYTE
3461 047642      BGT      10$            ;DONE ALL BYTES?
3462 047644      BGT      10$            ;BR IF NO
3463      RTS      PC                    ;RETURN
3464
3465      ;*
3466      ; SETUP T17PK2 PACKET FOR READ FIFO
3467      ;*
3467      ; INPUT:
3468      ;      RO CONTAINS SEL2 BYTE COUNT
3469      ;*
3470 047646      T17RFIF:
3471 047646      JSR      PC,T17CLRBUF      ;CLEAR MESSAGE BUFFER
3472 047652      MOV      @T17DT2,R1      ;WRITE SUBSYSTEM DATA BUFFER
3473 047656      MOV      @PW.RFIFO,(R1)+  ;STORE READ FIFO IN BSEL0
3474 047662      MOV      RO,(R1)+  ;STORE BYTE COUNT IN BSEL1

```


G13

TSV5 - HARDWARE TESTS MACRO M113 01-FEB-84 17:02
TEST 6: SUBTEST 6: FIFO VERIFY RESET FIFO TEST

SEQ 162

```

3534
3536 050102
3538 050110
3539 050110 100006
3540 050112 050120'
3541 050114 000000
3542 050116 000012
3543
3544 050120
3545 050120 000
3546 050121 000
3547 050122 000000
3548 050124
3549
3550 050226
    050226
    050226 104401.
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572 050230
    050230
3577 050230 012700 050736'
3578 050234 004737 016322'
3579 050240 012737 000012 002216'
3580 050246
3581
3582 050246
3583 050246 004737 015604'
3584 050252 103405
3585 050254 010001
3586 050256
    050256 104455
    050260 001274
    050262 003646
    050264 011644'
3587
3588
3589 050266 005037 002222'

;
; .BLKB 10-<.-TSV2&7>
T17PK2:
; .WORD P.WRTSUB!P.ACK ;WRITE SUBSYSTEM WITH ACK
; .WORD T17DT2 ;LOW ADDRESS OF DATA BLOCK
; .WORD 0 ;HIGH ADDRESS OF DATA BLOCK
; .WORD 10. ;MINIMUM MESSAGE PACKET SIZE

T17DT2:
; .BYTE 0 ;DATA BLOCK
; .BYTE 0 ;BSELO
; .WORD 0 ;BSEL1
; .BLKB 66. ;SEL2
;WRITE FIFO DATA OUTPUT BUFFER

ENDTST

L10056: TRAP C$ETST

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

BGNTST

T7::
MOV #TST18ID,R0 ;ASCII MESSAGE TO IDENTIFY TEST
JSR PC,TSTSETUP ;DO INITIAL TEST SETUP
MOV #10.,LOOPCNT ;PERFORM 10 ITERATIONS

T18LOOP:
; Write to TSSR register to soft initialize the controller
5$:
JSR PC,SOFINIT ;WRITE TO TSSR TO SOFT INITIALIZE
BCS 10$ ;BR IF SOFT INIT OK
MOV R0,R1 ;SAVE CONTENTS OF TSSR
ERRDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
TRAP C$ERDF
; .WORD 700
; .WORD SFIERR
; .WORD SFIMSG

10$:
; Do WRITE CHARACTERISTICS to check for Extended Features Switch
CLR FATFLG ;CLEAR FATAL ERROR FLAG

```

H13

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

SEQ 163

```

3590 050272 012704 051420'      MOV    #T18PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
3591 050276 004737 010472'      JSR     PC,WRTCHR      ;DO WRITE CHARACTERISTICS COMMAND
3592 050302                FORCERROR 12$      ;BDDFORCE ERROR IF FORCER=1
3593 050316 103407                BCS     15$      ;BR IF CARRY SET (GOOD RETURN)
3594 050320 010001                MOV     RO,R1      ;SAVE CONTENTS OF TSSR
3595 050322                NEXT,ERRNO
3596 050322                12$: ERRDF  ERRNO,T18SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     701
                                .WORD     T18SSR
                                .WORD     PKTSSR
                                3597 050332 005237 002222'      INC     FATFLG      ;SET FATAL ERROR FLAG
                                3598 050336                15$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
3599                                ;
3600                                ; If Extended Features Hardware Switch Clear then:
3601                                ; Do Write Subsystem Write Miscellaneous to Set Extended Features.
3602 050340 012701 051442'      MOV     #T18BFR,R1      ;MESSAGE BUFFER ADDRESS
3603 050344 032761 000200 000012 BIT     #X2.EXTF,XST2(R1) ;EXTENDED FEATURES SWITCH SET?
3604 050352 001026                BNE     30$      ;BR IF YES
3605 050354 004737 051266'      JSR     PC,T18SMISC      ;SETUP PACKET FOR WRITE MISCELLANEOUS
3606 050360 012704 051470'      MOV     #T18PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
3607 050364 010465 000000      MOV     R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
3608 050370 004737 016146'      JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
3609 050374                FORCERROR 22$      ;BDDFORCE ERROR IF FORCER=1
3610 050410 103407                BCS     30$      ;BR IF CARRY SET (GOOD RETURN)
3611 050412 010001                MOV     RO,R1      ;SAVE CONTENTS OF TSSR
3612 050414                NEXT,ERRNO
3613 050414                22$: ERRDF  ERRNO,T182SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     702
                                .WORD     T182SSR
                                .WORD     PKTSSR
                                3614 050424 005237 002222'      INC     FATFLG      ;SET FATAL ERROR FLAG
                                3615 050430                30$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
3616                                ;
3617                                ; Do WRITE CHARACTERISTICS to select reserved unit 7
3618                                ;
3619 050432 005037 002222'      CLR     FATFLG      ;CLEAR FATAL ERROR FLAG
3620 050436 012704 051420'      MOV     #T18PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
3621 050442 004737 010472'      JSR     PC,WRTCHR      ;DO WRITE CHARACTERISTICS COMMAND
3622 050446                FORCERROR 42$      ;BDDFORCE ERROR IF FORCER=1
3623 050462 103407                BCS     50$      ;BR IF CARRY SET (GOOD RETURN)
3624 050464 010001                MOV     RO,R1      ;SAVE CONTENTS OF TSSR
3625 050466                NEXT,ERRNO
3626 050466                42$: ERRDF  ERRNO,T18SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     703
                                .WORD     T18SSR
                                .WORD     PKTSSR
                                3627 050476 005237 002222'      INC     FATFLG      ;SET FATAL ERROR FLAG
                                3628 050502                50$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
3629                                ;
3630                                ; Clear message buffer
3631 050504 012701 051442'      MOV     #T18BFR,R1      ;GET MESSAGE BUFFER ADDRESS

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

SEQ 164

```

3632 050510 013700 051434'      MOV      T18DATA+4,R0      ;SIZE OF MESSAGE BUFFER IN BYTES
3633 050514 105021      60$: CLRB      (R1)+      ;CLEAR A BYTE
3634 050516 005300      DEC      R0      ;DONE?
3635 050520 003375      BGT      60$      ;BR IF NO
3636      ; Do a Write Subsystem READ STATUS
3637 050522 004737 051246'      JSR      PC,T18SRD      ;SETUP PACKET FOR READ STATUS
3638 050526 012704 051470'      MOV      @T18PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
3639 050532 010465 000000      MOV      R4,(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
3640 050536 004737 016146'      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
3641 050542      FORCERROR      62$      ;GOOD FORCE ERROR IF FORCER=1
3642 050556 103407      BCS      70$      ;BR IF CARRY SET (GOOD RETURN)
3643 050560 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
3644 050562      NEXT.ERRNO
3645 050562      62$: ERRDF      ERRNO,T183SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      704
                                .WORD      T183SSR
                                .WORD      PKTSSR
3646 050572 005237 002222'      INC      FATFLG      ;SET FATAL ERROR FLAG
3647 050576      70$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
3648
3649
3650      ; Set first 8 words of expd message buffer = to recv since not testing
3651      ; Set unused bits in Read Status expd equal rcvd
3652 050600 004737 051310'      JSR      PC,T18SETEXP      ;SET SOME EXPD TO RECV
3653      ; If any transport interface signals are asserted then Print Error
3654 050604 005000      CLR      R0      ;HIGH RECV ADDRESS FOR CKMSG2
3655 050606 012701 051442'      MOV      @T18BFR,R1      ;LOW RECV ADDRESS FOR CKMSG2
3656 050612 012702 050706'      MOV      @T18EXP,R2      ;EXPD ADDRESS
3657 050616 012703 000012      MOV      @10,R3      ;NUMBER OF WORDS TO COMPARE
3658 050622 004737 011310'      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
3659 050626      FORCERROR      82$,NOTSSR      ;GOOD
3660 050636 103404      BCS      90$      ;BR IF YES
3661 050640      NEXT.ERRNO
3662 050640      82$: ERRHRD      ERRNO,T18CMP,MSGSTAT      ;REPORT ERROR
                                TRAP      C$ERRRD
                                .WORD      705
                                .WORD      T18CMP
                                .WORD      MSGSTAT
3663 050650      90$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
3664 050650 104406
3665 050652 005737 002222'      TST      FATFLG      ;ANY FATAL ERRORS ?
3666 050656 001402      BEQ      160$      ;BRANCH IF NOT
3667 050660 004737 017014'      JSR      PC,CKDROP      ;TRY TO DROP THE UNIT
3668 050664 004737 016270'      160$: JSR      PC,TSTLOOP      ;DO ITERATIONS?
3669 050670 103072      BCC      165$      ;BR IF NO
3670 050672 000137 050246'      JMP      T18LOOP      ;LOOP UNTIL ITERATIONS DONE
3671 050676      165$:
3672 050676      EXIT      TST
                                TRAP      C$EXIT
                                .WORD      L10065
3673
3674      ; LOCAL STORAGE FOR THIS TEST
3675
3676

```

```

3677
3678 050702          T18MSK:          ;MASK OF UNUSED BITS IN READ STATUS BYTES
3679 050702          .BYTE 377        ;BYTE 0 MASK
3680 050703          .BYTE 037        ;BYTE 1
3681 050704          .BYTE 100        ;BYTE 2
3682 050705          .BYTE 000        ;MAKE IT EVEN
3683
3684 050706          T18EXP:          ;EXPECTED DATA BUFFER
3685 050706 000000    .WORD 0          ;MESSAGE TYPE
3686 050710 000000    .WORD 0          ;DATA FIELD LENGTH
3687 050712 000000    .WORD 0          ;RBPGR
3688 050714 000000    .WORD 0          ;XST0
3689 050716 000000    .WORD 0          ;XST1
3690 050720 000000    .WORD 0          ;XST2
3691 050722 000000    .WORD 0          ;XST3
3692 050724 000000    .WORD 0          ;XST4 (ALWAYS PRESENT FOR WRITE SUB)
3693 050726 000000    .WORD 0          ;READ STATUS BYTE 1/0
3694 050730 000000    .WORD 0          ;READ STATUS BYTE 2
3695
3696 050732          T18XS: .BYTE 377,020 ;READ STATUS BYTE 0/1 EXPECTED BASE
3697 050734 000000    .WORD 0          ;READ STATUS BYTE 2 EXPECTED BASE
3698
3699          ;*
3700          ;LOCAL TEXT MESSAGES FOR TEST
3701          ;-
3702
3703 050736          123      164      141 TST18ID: .ASCIZ 'Static Transport Bus Interface'
3704 050737          127      122      111 T18SSR: .ASCIZ 'WRITE CHARACTERISTICS Failed'
3705 051032          127      122      111 T182SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Misc) Failed'
3706 051076          127      122      111 T183SSR: .ASCIZ 'WRITE SUBSYSTEM (Read Status) Failed'
3707 051143          124      162      141 T18CMP: .ASCIZ 'Transport Bus Interface Signals NOT Negated After Unit 7 Selected'
3708          .EVEN
3709
3710          ;*
3711          ; SETUP T18PK2 PACKET FOR READ STATUS
3712          ;-
3713 051246          T18SRD:
3714 051246          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
3715 051252 012700 051500'          MOV 4T18DT2,R0          ;WRITE SUBSYSTEM DATA BUFFER
3716 051256 112720 000005          MOVB 4PW.RDSTATUS,(R0)+ ;STORE READ STATUS COMMAND IN BSEL0
3717 051262 105010          CLRB (R0)          ;CLEAR BSEL1
3718 051264 000207          RTS PC          ;RETURN
3719
3720          ;*
3721          ; SETUP T18PK2 PACKET FOR WRITE MISC.
3722          ;-
3723 051266          T18SMISC:
3724 051266          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
3725 051272 012700 051500'          MOV 4T18DT2,R0          ;WRITE SUBSYSTEM DATA BUFFER
3726 051276 112720 000005          MOVB 4PW.WMISC,(R0)+ ;STORE WRITE MISCELLANEOUS IN BSEL0
3727 051302 112710 000200          MOVB 4MS.EXT,(R0)+ ;STORE INVERT EXTENDED FEATURES IN BSEL1
3728 051306 000207          RTS PC          ;RETURN
3729
3730          ;*
3731          ;Set first 8 words of expd message buffer - to recv since not testing
3732          ; Set unused bits in Read Status expd equal recvd
3733          ;-

```

K13

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

SEQ 166

```

3734 051310          T18SETEXP:
3735 051310 012702 050706'      MOV      #T18EXP,R2          ;GET EXPD
3736 051314 012703 051442'      MOV      #T18BFR,R3         ;GET READ STATUS RECV BUFFER
3737 051320 012700 000010      MOV      #8.,R0              ;SET FIRST 8 WORDS EXP=RECV
3738 051324 012322          5$:  MOV      (R3)+,(R2)+         ;SET EXPD=RECV
3739 051326 005300          DEC      R0                      ;DONE FIRST 8 WORDS?
3740 051330 003375          BGT      5$                      ;BR IF NO
3741 051332 012701 050702'      MOV      #T18MSK,R1         ;GET UNUSED BIT MASK
3742 051336 013712 050732'      MOV      T18XS,(R2)         ;SETUP BASE EXPECTED BYTE 1/0
3743 051342 013762 050734' 000002 MOV      T18XS+2,2(R2)    ;SETUP BASE EXPECTED BYTE 2
3744 051350 011300          MOV      (R3),R0                ;GET RECV BYTE 1 AND BYTE 0
3745 051352 041100          BIC      (R1),R0                ;CLEAR ALL BUT UNUSED
3746 051354 040012          BIC      R0,(R2)                ;CLEAR UNUSED IN EXP
3747 051356 050012          BIS      R0,(R2)                ;SET UNUSED EXPD=RECV FOR COMPARE
3748 051360 016300 000002      MOV      2(R3),R0            ;GET RECV BYTE 2
3749 051364 046100 000002      BIC      2(R1),R0            ;CLEAR ALL BUT UNUSED
3750 051370 040062 000002      BIC      R0,2(R2)            ;CLEAR UNUSED IN EXPD
3751 051374 050062 000002      BIS      R0,2(R2)            ;SET UNUSED EXPD=RECV FOR COMPARE
3752 051400 105062 000003      CLRB    3(R2)                ;CLEAR EXPD BYTE 3 (UNUSED)
3753 051404 105063 000003      CLRB    3(R3)                ;CLEAR RECV BYTE 3 (UNUSED)
3754 051410 000207          RTS      PC                      ;RETURN
3755
3757 051412          .BLKB    10-<.-TSV267>
3759
3760          ;WRITE CHARACTERISTICS COMMAND PACKET
3761
3762 051420          T18PACKET:
3763 051420 100004          .WORD    100004                ;COMMAND PACKET FOR TEST
3764 051422 051430'      .WORD    T18DATA                ;WRITE CHARACTERISTICS COMMAND, WITH ACK
3765 051424 000000          .WORD    0                    ;ADDRESS OF CHARACTERISTICS BLOCK
3766 051426 000012          .WORD    10.                   ;MESSAGE PACKET MINIMUM SIZE
3767
3768 051430          T18DATA:
3769 051430 051442'      .WORD    T18BFR                  ;CHARACTERISTICS DATA BLOCK
3770 051432 000000          .WORD    0                    ;ADDRESS OF MESSAGE BUFFER
3771 051434 000024          .WORD    20.                   ;LENGTH OF MESSAGE BUFFER
3772 051436 000000          .WORD    0                    ;ESS,ENB,EAI,ERI
3773 051440 000007          .WORD    7                    ;SELECT RESERVED UNIT 7
3774
3775
3776 051442          T18BFR:
3777 051442 000000          .WORD    0                    ;MESSAGE BUFFER
3778 051444 000000          .WORD    0                    ;MESSAGE TYPE
3779 051446 000000          .WORD    0                    ;DATA FIELD LENGTH
3780 051450 000000          .WORD    0                    ;RBPOR
3781 051452 000000          .WORD    0                    ;XST0
3782 051454 000000          .WORD    0                    ;XST1
3783 051456 000000          .WORD    0                    ;XST2
3784 051460 000000          .WORD    0                    ;XST3
3785 051462 000000          .WORD    0                    ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM)
3786 051464 000000          .WORD    0                    ;READ STATUS BYTE 1/0 RETURNED
3787
3788          ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
3789
3791 051466          .BLKB    10-<.-TSV267>
3793 051470          T18PK2:
3794 051470 100006          .WORD    P,WRTSUB!P,ACK          ;WRITE SUBSYSTEM WITH ACK

```

L13

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

SEQ 167

```

3795 051472 051500'      .WORD    T18DT2      ;LOW ADDRESS OF DATA BLOCK
3796 051473 000000      .WORD    0          ;HIGH ADDRESS OF DATA BLOCK
3797 051476 000010      .WORD    8          ;BUFFER EXTENT
3798
3799 051500      T18DT2:      ;DATA BLOCK
3800 051500      .BYTE    0          ;BSEL0
3801 051501      .BYTE    0          ;BSEL1
3802 051502 000000      .WORD    0          ;SEL2
3803 051504 000000      .WORD    0          ;DATA
3804
3805
3806 051506      ENDTST
3807 051506 104401      L10065: TRAP    C$ETST
3808
3809      .SBTTL  TEST  8: TRANSPORT BUS INTERFACE LOOPBACK TEST
3810
3811      ;++
3812      ; TEST DESCRIPTION:
3813      ;
3814      ; This test verifies the controller's Transport Bus
3815      ; drivers, receivers, and signal loopback logic. Note
3816      ; that the Static Transport Bus test must have run
3817      ; correctly for this test to provide meaningful results.
3818
3819      ; TEST STEPS:
3820      ;
3821      ; REPEAT FOR LOOPCNT
3822      ; BEGIN
3823      ; Do Subtest 1 - Loopback Control signals test
3824      ; Do Subtest 2 - Loopback Read/Write signals test
3825      ; Do Subtest 3 - Loopback Write Strobe test
3826      ; Do Subtest 4 - Loopback Read Strobe test
3827      ; END
3828      ;--
3829 051510      BCONTST
3830 051510
3834 051510 012700 000000      MOV     @TST19ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
3835 051514 004737 000000      JSR     PC,TSTSETUP      ;DO INITIAL TEST SETUP
3836 051520 012737 000000 002216'      MOV     @10,LOOPCNT      ;PERFORM 10 ITERATIONS
3837 051526      T18LOOP:
3838
3839      .SBTTL  TEST  8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST
3840
3841      ;++
3842      ; TEST 8: SUBTEST 1:
3843      ;
3844      ; SUBTEST DESCRIPTION:
3845      ;
3846      ; This subtest verifies the Transport Control loopback
3847      ; path can transmit and receive correctly. The
3848      ; control signals are all loopback signals other
3849      ; than the read/write data (IW<7:0> and IR<7:0>).
3850
3851      ; TEST STEPS:
3852

```

M13

TSV5 - HARDWARE TESTS MACRO M1113 01 FEB 84 17:02
TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

SEQ 168

```

3853 ; The loopback signals IFAD,ITAD0,ITAD1 are the tape unit select
3854 ; lines. Since reserved unit 7 must remain selected these signals
3855 ; are always set low. This further means the signals they drive
3856 ; (ISPEED,IRDY,IONL) are only tested in the low state.
3857 ;
3858 ; BEGIN
3859 ; Write to TSSR register to soft initialize the controller
3860 ; Do WRITE CHARACTERISTICS to check for Extended Features Switch
3861 ; If Extended Features Hardware Switch Clear then:
3862 ;   Do Write Subsystem Write Miscellaneous to Set Extended Features.
3863 ; Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
3864 ; Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
3865 ; Do Write Subsystem Write Control to CLEAR loopback signals group 1.
3866 ; Do Write Subsystem Write Format to CLEAR loopback signals group 2.
3867 ;   (the loopback signals have to be cleared here due to the flip-flops
3868 ;   that are set on a 1 to 0 transition (IHER,IFMK,ICER))
3869 ; Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
3870 ; Do a Write Subsystem READ STATUS
3871 ; If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
3872 ; signals NOT=0 Then Print Error
3873 ;
3874 ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
3875 ; BEGIN
3876 ;   Do Write Subsystem Write Control to Drive loopback signals group 1.
3877 ;   Do Write Subsystem Write Format to Drive loopback signals group 2.
3878 ;   Do a Write Subsystem READ STATUS
3879 ;   If loopback data NOT= data sent Then Print Error
3880 ;   Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
3881 ;   Do a Write Subsystem READ STATUS
3882 ;   If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
3883 ;   signals NOT=0 Then Print Error
3884 ; END
3885 ;
3886 051526 BGNSUB ;////////// BEGIN SUBTEST ///////////
      051526 T8.1: TRAP C$BSUB
      051526 104402
3887 ;
3888 ; Write to TSSR register to soft initialize the controller
3889 051530 5$: JSR PC,SOFINIT ;WRITE TO TSSR TO SOFT INITIALIZE
3890 051530 004737 015604' BCS 10$ ;BR IF SOFT INIT OKAY
3891 051534 103405 MOV R0,R1 ;SAVE CONTENTS OF TSSR
3892 051536 010001 ERDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
3893 051540 TRAP C$ERDF
      051540 104455 .WORD 800
      051542 001440 .WORD SFIERR
      051544 003646' .WORD SFIMSG
      051546 011644'
3894 ; Do WRITE CHARACTERISTICS to check for Extended Features Switch
3895 051550 10$: CLR FATELG ;CLEAR FATAL ERROR FLAG
3896 051554 012704 062050' MOV #T19PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
3897 051560 004737 010472' JSR PC,WRTCHR ;DO WRITE CHARACTERISTICS COMMAND
3898 051564 FORCERROR 12$ ;DO FORCE ERROR IF FORCER=1
3899 051600 103407 BCS 15$ ;BR IF CARRY SET (GOOD RETURN)
3900 051602 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
3901 051604 NEXT,ERRNO
3902 051604 12$: ERDF ERRNO,T19SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      051604 104455 TRAP C$ERDF

```

Address	Offset	Hex	Label	Operation	Comments	Register/Value	SSR
051606	001441						801
051610	057763'						T19SSR
051612	011656'						PKTSSR
3903	051614	005237	002222'	INC	FATFLG		
3904	051620			CKLOOP			
051620	104406						
3905							
3906							
3907	051622	012701	062072'	MOV	#T19BFR,R1		
3908	051626	032761	000200 000012	BIT	#X2.EXTF,XST2(R1)		
3909	051634	001026		BNE	30\$		
3910	051636	004737	061722'	JSR	PC,T19SEXT		
3911	051642	012704	062220'	MOV	#T19PK2,R4		
3912	051646	010465	000000	MOV	R4,TSDB(R5)		
3913	051652	004737	016146'	JSR	PC,CHKTSSR		
3914	051656			FORCERROR	22\$		
3915	051672	103407		BCS	30\$		
3916	051674	010001		MOV	R0,R1		
3917	051676			NEXT.ERRNO			
3918	051676			ERRDF	ERRNO,T192SSR,PKTSSR		
051676	104455						
051700	001442						
051702	060020'						
051704	011656'						
3919	051706	005237	002222'	INC	FATFLG		
3920	051712			CKLOOP			
051712	104406						
3921							
3922	051714	005037	002222'	CLR	FATFLG		
3923	051720	012704	062050'	MOV	#T19PACKET,R4		
3924	051724	004737	010472'	JSR	PC,WRTCHR		
3925	051730			FORCERROR	42\$		
3926	051744	103407		BCS	50\$		
3927	051746	010001		MOV	R0,R1		
3928	051750			NEXT.ERRNO			
3929	051750			ERRDF	ERRNO,T19SSR,PKTSSR		
051750	104455						
051752	001443						
051754	057763'						
051756	011656'						
3930	051760	005237	002222'	INC	FATFLG		
3931	051764			CKLOOP			
051764	104406						
3932							
3933	051766	012700	000100	MOV	#NP.OUT,R0		
3934	051772	052700	000040	BIS	#NP.LOOP,R0		
3935	051776	004737	061562'	JSR	PC,T19SNPR		
3936	052002	012704	062220'	MOV	#T19PK2,R4		
3937	052006	010465	000000	MOV	R4,TSDB(R5)		
3938	052012	004737	016146'	JSR	PC,CHKTSSR		
3939	052016			FORCERROR	62\$		
3940	052032	103407		BCS	70\$		
3941	052034	010001		MOV	R0,R1		
3942	052036			NEXT.ERRNO			
3943	052036			ERRDF	ERRNO,T194SSR,PKTSSR		
052036	104455						
052040	001444						

Address	Offset	Hex Data	Label	Operation	Comments
052042	060131				.WORD T194SSR
052044	011656				.WORD PKTSSR
3944	052046	005237	002222	INC FATFLG	;SET FATAL ERROR FLAG
3945	052052		701:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
052052	104406				TRAP C1CLP1
3946					Do Write Subsystem Write Control to CLEAR loopback signals group 1.
3947					Do Write Subsystem Write Format to CLEAR loopback signals group 2.
3948					(the loopback signals have to be cleared here due to the flip-flops
3949					that are set on a 1 to 0 transition (IHER,IFMK,ICER))
3950	052054	005000		CLR R0	;WRITE 0'S
3951	052056	042700	000200	BIC #WC,IFAD,R0	;IFAD MUST ALWAYS =0
3952	052062	042700	000100	BIC #WC,IOTAD,R0	;ITADO MUST ALWAYS =0
3953	052066	042700	000040	BIC #WC,IITAD,R0	;ITADI MUST ALWAYS =0
3954	052072	004737	061662	JSR PC,T19WCTL	;SETUP PACKET FOR WRITE CONTROL
3955	052076	012704	062220	MOV #T19PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
3956	052102	010465	000000	MOV R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE
3957	052106	004737	016146	JSR PC,CHKTSSR	;WAIT FOR SSR TO SET
3958	052112			FORCERROR 821	;GOODFORCE ERROR IF FORCER=1
3959	052126	103407		BCS 901	;BR IF CARRY SET (GOOD RETURN)
3960	052130	010001		MOV R0,R1	;SAVE CONTENTS OF TSSR
3961	052132			NEXT,ERRNO	
3962	052132		821:	ERRDF ERRNO,T197SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
052132	104455				TRAP C1ERDF
052134	001145				.WORD 805
052136	060303				.WORD T197SSR
052140	011656				.WORD PKTSSR
3963	052142	005237	002222	INC FATFLG	;SET FATAL ERROR FLAG
3964	052146		901:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
052146	104406				TRAP C1CLP1
3965	052150	005000		CLR R0	;SET FORMAT DRIVE DATA=0
3966	052152	004737	061702	JSR PC,T19WFMF	;SETUP PACKET FOR WRITE FORMAT
3967	052156	012704	062220	MOV #T19PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
3968	052162	010465	000000	MOV R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE
3969	052166	004737	016146	JSR PC,CHKTSSR	;WAIT FOR SSR TO SET
3970	052172			FORCERROR 1021	;GOODFORCE ERROR IF FORCER=1
3971	052206	103407		BCS 1101	;BR IF CARRY SET (GOOD RETURN)
3972	052210	010001		MOV R0,R1	;SAVE CONTENTS OF TSSR
3973	052212			NEXT,ERRNO	
3974	052212		1021:	ERRDF ERRNO,T198SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
052212	104455				TRAP C1ERDF
052214	001446				.WORD 806
052216	060352				.WORD T198SSR
052220	011656				.WORD PKTSSR
3975	052222	005237	002222	INC FATFLG	;SET FATAL ERROR FLAG
3976	052226		1101:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
052226	104406				TRAP C1CLP1
3977					Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
3978	052230	004737	061540	JSR PC,T19R5FIF	;SETUP PKT FOR WRITE MISC Reset Tape Status F-FLOPS
3979	052234	012704	062220	MOV #T19PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
3980	052240	010465	000000	MOV R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE
3981	052244	004737	016146	JSR PC,CHKTSSR	;WAIT FOR SSR TO SET
3982	052250			FORCERROR 1221	;GOODFORCE ERROR IF FORCER=1
3983	052264	103407		BCS 1301	;BR IF CARRY SET (GOOD RETURN)
3984	052266	010001		MOV R0,R1	;SAVE CONTENTS OF TSSR
3985	052270			NEXT,ERRNO	
3986	052270		1221:	ERRDF ERRNO,T192SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
052270	104455				TRAP C1ERDF

	052272	001447					.WORD	807
	052274	060020					.WORD	T192SSR
	052276	011656					.WORD	PKTSSR
3987	052300	005237	002222'		INC	FATFLG		;SET FATAL ERROR FLAG
3988	052304			130:	CKLOOP			;LOOP ON ERROR, IF FLAG SET TRAP C:CLP1
3989		104406						
3990						Do a Write Subsystem READ STATUS		
3991						If all Tape Status 2 (ICER,IFMK,IHER) Flip-Flop signals NOT=0 Then Print Error		
3992	052306	004737	061520'		JSR	PC,T19SRD		;SETUP PACKET FOR READ STATUS
3993	052312	012704	062220'		MOV	#T19PK2,R4		;GET WRITE SUBSYSTEM COMMAND PACKET
3994	052316	010465	000000		MOV	R4,TSD8(R5)		;SET THE PACKET ADDRESS TO EXECUTE
3995	052322	004737	016146'		JSR	PC,CHKTSSR		;WAIT FOR SSR TO SET
3996	052326				FORCERROR	132:		;GOODFORCE ERROR IF FORCER=1
3997	052342	103407			BCS	140:		;BR IF CARRY SET (GOOD RETURN)
3998	052344	010001			MOV	RO,R1		;SAVE CONTENTS OF TSSR
3999	052346				NEXT,ERRNO			
4000	052346			132:	ERRDF	ERRNO,T193SSR,PKTSSR		;DEVICE FATAL SSR FAILED TO SET TRAP C:ERDF
	052346	104455					.WORD	808
	052350	001450					.WORD	T193SSR
	052352	060064					.WORD	PKTSSR
	052354	011656						
4001	052356	005237	002222'		INC	FATFLG		;SET FATAL ERROR FLAG
4002	052362			140:	CKLOOP			;LOOP ON ERROR, IF FLAG SET TRAP C:CLP1
	052362	104406						
4003	052364	004737	061760'		JSR	PC,T19SETEXP		;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
4004	052370	012701	057622'		MOV	#T19EXSTA,R1		;GET EXPECTED READ STATUS
4005	052374	012702	062112'		MOV	#T19BFSTA,R2		;GET RCV READ STATUS
4006	052400	011211			MOV	(R2),(R1)		;SET EXPD WORD #8 = RCV TEMP
4007	052402	042711	002000		BIC	#S1.ICER,(R1)		;SET EXPD ICER =0
4008	052406	042711	001000		BIC	#S1.IFMK,(R1)		;SET EXPD IFMK =0
4009	052412	042711	000400		BIC	#S1.IHER,(R1)		;SET EXPD IHER =0
4010	052416	016261	000002	000002	MOV	2(R2),2(R1)		;SET EXPD WORD #9 = RCV (NOT TESTING)
4011	052424	005000			CLR	RO		;HIGH RCV ADDRESS FOR CKMSG2
4012	052426	012701	062072'		MOV	#T19BFR,R1		;LOW RCV ADDRESS FOR CKMSG2
4013	052432	012702	057602'		MOV	#T19EXP,R2		;EXPD ADDRESS
4014	052436	012703	000024		MOV	#20,,R3		;NUMBER OF BYTES TO COMPARE
4015	052442	004737	011310'		JSR	PC,CKMSG2		;EXPD EQUAL RCV?
4016	052446				FORCERROR	152:,NOTSSR		;GOOD
4017	052456	103404			BCS	160:		;BR IF YES
4018	052460				NEXT,ERRNO			
4019	052460			152:	ERRHRD	ERRNO,T197CMP,MSGLoop		;REPORT ERROR TRAP C:ERHRD
	052460	104456					.WORD	809
	052462	001451					.WORD	T197CMP
	052464	061023					.WORD	MSGLoop
	052466	012674						
4020	052470			160:	CKLOOP			;LOOP ON ERROR, IF FLAG SET TRAP C:CLP1
	052470	104406						
4021						REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE		
4022	052472	005037	057534'		CLR	T19PREV		;INIT 1-0 TRANSITION FLAG
4023	052476	012703	002752'		MOV	#TSTBLK,R3		;GET FIRST PATTERN ADDRESS
4024	052502	012300		200:	MOV	(R3),,RO		;GET A TEST PATTERN
4025	052504	010337	002316'		MOV	R3,TSTPIR		;SAVE POINTER INTO TSTBLK
4026	052510	042700	000200		BIC	#WC.IFAD,RO		;IFAD MUST ALWAYS =0
4027	052514	042700	000100					

```

4030
4031
4032 052530 013700 002312'
4033 052534 004737 062004'
4034
4035 052540 004737 061662'
4036 052544 012704 062220'
4037 052550 010465 000000'
4038 052554 004737 016146'
4039 052560
4040 052574 103407
4041 052576 010001
4042 052600
4043 052600
      052600 104455
      052602 001452
      052604 060303'
      052606 011656'
4044 052610 005237 002222'
4045 052614
      052614 104406
4046
4047
4048
4049 052616 013700 002312'
4050 052622 004737 062004'
4051 052626 000300
4052 052630 004737 061702'
4053 052634 012704 062220'
4054 052640 010465 000000'
4055 052644 004737 016146'
4056 052650
4057 052664 103407
4058 052666 010001
4059 052670
4060 052670
      052670 104455
      052672 001453
      052674 060352'
      052676 011656'
4061 052700 005237 002222'
4062 052704
      052704 104406
4063
4064 052706 004737 061520'
4065 052712 012704 062220'
4066 052716 010465 000000'
4067 052722 004737 016146'
4068 052726
4069 052742 103407
4070 052744 010001
4071 052746
4072 052746
      052746 104455
      052750 001454
      052752 060064'
      052754 011656'

; Do Write Subsystem Write Control to Drive loopback signals group 1.
; 000 CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
MOV DATA,R0 ;GET TEST PATTERN
JSR PC,T19CNVT ;CONVERT PATTERN TO CONTROL DRIVE MASK
;R0 CONTAINS WRITE CONTROL DATA HERE
JSR PC,T19WCTL ;SETUP PACKET FOR WRITE CONTROL
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 212; ;000FORCE ERROR IF FORCER=1
BCS 220; ;BR IF CARRY SET (GOOD RETURN)
MOV R0,R1 ;SAVE CONTENTS OF TSSR
NEXT,ERRNO
212;: ERDF ERRNO,T197SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      TRAP C:ERDF
      .WORD 810
      .WORD T197SSR
      .WORD PKTSSR
INC FATFLG ;SET FATAL ERROR FLAG
220;: CKLOOP ;LOOP ON ERROR, IF FLAG SET
      TRAP C:CLP1

; 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
SWAB R0 ;WRITE FORMAT DATA RETURNED IN HIGH BYTE
JSR PC,T19WFMF ;SETUP PACKET FOR WRITE FORMAT
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 232; ;000FORCE ERROR IF FORCER=1
BCS 240; ;BR IF CARRY SET (GOOD RETURN)
MOV R0,R1 ;SAVE CONTENTS OF TSSR
NEXT,ERRNO
232;: ERDF ERRNO,T198SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      TRAP C:ERDF
      .WORD 811
      .WORD T198SSR
      .WORD PKTSSR
INC FATFLG ;SET FATAL ERROR FLAG
240;: CKLOOP ;LOOP ON ERROR, IF FLAG SET
      TRAP C:CLP1

; Do Write Subsystem READ STATUS
JSR PC,T19SRD ;SETUP PACKET FOR READ STATUS
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 252; ;000FORCE ERROR IF FORCER=1
BCS 260; ;BR IF CARRY SET (GOOD RETURN)
MOV R0,R1 ;SAVE CONTENTS OF TSSR
NEXT,ERRNO
252;: ERDF ERRNO,T193SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      TRAP C:ERDF
      .WORD 812
      .WORD T193SSR
      .WORD PKTSSR

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

SEQ 173

```

4073 052756 005237 002222'      INC      FATFLG      ;SET FATAL ERROR FLAG
4074 052762      260$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      052762 104406      TRAP      C$CLP1
4075      ;      If loopback data NOT= data sent Then Print Error
4076 052764 004737 061760'      JSR      PC,T19SETEXP ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
4077 052770 012701 057622'      MOV      @T19EXSTA,R1 ;GET EXPECTED READ STATUS
4078 052774 012702 062112'      MOV      @T198FSTA,R2 ;GET RECV READ STATUS
4079 053000 013711 002312'      MOV      DATA,(R1) ;SET EXPD WORD #8 TO TEST DATA FIRST
4080 053004 013700 057534'      MOV      T19PREV,R0 ;GET PREVIOUS DATA PATTERN
4081 053010 013703 002312'      MOV      DATA,R3 ;GET CURRENT PATTERN
4082 053014 012704 000400      MOV      @S1.IHER,R4 ;SETUP IHER EXPECTED
4083 053020 040411      BIC      R4,(R1) ;SET EXPD IHER =0
4084 053022 030400      BIT      R4,R0 ;PREVIOUS =1?
4085 053024 001403      BEQ      275$ ;BR IF NO
4086 053026 030403      BIT      R4,R3 ;CURRENT =0?
4087 053030 001001      BNE      275$ ;BR IF NO
4088 053032 050411      BIS      R4,(R1) ;SET EXPD IHER =1
4089 053034 012704 001000      275$: MOV      @S1.IFMK,R4 ;SETUP IFMK EXPECTED
4090 053040 040411      BIC      R4,(R1) ;SET EXPD IFMK =0
4091 053042 030400      BIT      R4,R0 ;PREVIOUS =1?
4092 053044 001403      BEQ      280$ ;BR IF NO
4093 053046 030403      BIT      R4,R3 ;CURRENT =0?
4094 053050 001001      BNE      280$ ;BR IF NO
4095 053052 050411      BIS      R4,(R1) ;SET EXPD IFMK =1
4096 053054 012704 002000      280$: MOV      @S1.ICER,R4 ;SETUP ICER EXPECTED
4097 053060 040411      BIC      R4,(R1) ;SET EXPD ICER =0
4098 053062 030400      BIT      R4,R0 ;PREVIOUS =1?
4099 053064 001403      BEQ      285$ ;BR IF NO
4100 053066 030403      BIT      R4,R3 ;CURRENT =0?
4101 053070 001001      BNE      285$ ;BR IF NO
4102 053072 050411      BIS      R4,(R1) ;SET EXPD ICER =1
4103 053074 011100      285$: MOV      (R1),R0 ;GET EXPD WORD
4104      ;      If previous IIDENT=1 and current is IIDENT=1 then EXPD= 0 else 1
4105 053076 012704 004000      MOV      @S1.IIDENT,R4 ;IIDENT
4106 053102 050400      BIS      R4,R0 ;ASSUME EXPD=1
4107 053104 030437 057534'      BIT      R4,T19PREV ;PREVIOUS IIDENT=1?
4108 053110 001403      BEQ      288$ ;BR IF NO
4109 053112 030403      BIT      R4,R3 ;IS CURRENT IIDENT=1?
4110 053114 001401      BEQ      288$ ;BR IF NO
4111 053116 040400      BIC      R4,R0 ;SET EXPD=0
4112 053120 052700 040000      288$: BIS      @S1.I2RES,R0 ;IRESV2 EXPD ALWAYS=1
4113 053124 052700 020000      BIS      @S1.I1RES,R0 ;IRESV1 EXPD ALWAYS=1
4114 053130 042700 100000      BIC      @S1.PARERR,R0 ;IGNORE PARERR
4115 053134 032712 100000      BIT      @S1.PARERR,(R2) ;IS PARERR SET IN RECV?
4116 053140 001402      BEQ      290$ ;BR IF NO
4117 053142 052700 100000      BIS      @S1.PARERR,R0 ;SET IN EXPD
4118 053146 010011      290$: MOV      R0,(R1) ;SETUP FINAL EXPD IN WORD #8
4119 053150 016261 000002 000002      MOV      2(R2),2(R1) ;SET EXPD WORD #9 = RECV (NOT TESTING)
4120 053156 005000      CLR      R0 ;HIGH RECV ADDRESS FOR CKMSG2
4121 053160 012701 062072'      MOV      @T198FR,R1 ;LOW RECV ADDRESS FOR CKMSG2
4122 053164 012702 057602'      MOV      @T19EXP,R2 ;EXPD ADDRESS
4123 053170 012703 000024      MOV      @20.,R3 ;NUMBER OF BYTES TO COMPARE
4124 053174 004737 011310'      JSR      PC,CKMSG2 ;EXPD EQUAL RECV?
4125 053200      FORCERRR      302$,NOTSSR ;NO
4126 053210 103404      BCS      310$ ;BR IF YES
4127 053212      NEXT.ERRNO
4128 053212      302$: ERRHRD ERRNO,T198CHP,MSGLOOP ;REPORT ERROR

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

SEQ 174

053212	104456				TRAP	C#ERHRD
053214	001455				.WORD	813
053216	061111'				.WORD	T198CMP
053220	012674'				.WORD	MSGLOOP
4129	053222			310\$:	CKLOOP	!LOOP ON ERROR, IF FLAG SET
	053222	104406				TRAP C#CLP1
4130					Do a Write Subsystem Write Misc	to Reset Tape Status F-FLOPS
4131	053224	004737	061540'		JSR PC,T19RSFIF	!SETUP PKT FOR WRITE MISC Reset STATUS
4132	053230	012704	062220'		MOV #T19PK2,R4	!GET WRITE SUBSYSTEM COMMAND PACKET
4133	053234	010465	000000		MOV R4,TSD8(R5)	!SET THE PACKET ADDRESS TO EXECUTE
4134	053240	004737	016146'		JSR PC,CHKTSSR	!WAIT FOR SSR TO SET
4135	053244				FORCERROR 322\$	!DO FORCE ERROR IF FORCER=1
4136	053260	103407			BCS 330\$	!BR IF CARRY SET (GOOD RETURN)
4137	053262	010001			MOV R0,R1	!SAVE CONTENTS OF TSSR
4138	053264				NEXT,ERRNO	
4139	053264			322\$:	ERRDF ERRNO,T192SSR,PKTSSR	!DEVICE FATAL SSR FAILED TO SET
	053264	104455				TRAP C#ERDF
	053266	001456				.WORD 814
	053270	060020'				.WORD T192SSR
	053272	011656'				.WORD PKTSSR
4140	053274	005237	002222'		INC FATFLG	!SET FATAL ERROR FLAG
4141	053300			330\$:	CKLOOP	!LOOP ON ERROR, IF FLAG SET
	053300	104406				TRAP C#CLP1
4142					Do a Write Subsystem READ STATUS	
4143	053302	004737	061520'		JSR PC,T19SRD	!SETUP PACKET FOR READ STATUS
4144	053306	012704	062220'		MOV #T19PK2,R4	!GET WRITE SUBSYSTEM COMMAND PACKET
4145	053312	010465	000000		MOV R4,TSD8(R5)	!SET THE PACKET ADDRESS TO EXECUTE
4146	053316	004737	016146'		JSR PC,CHKTSSR	!WAIT FOR SSR TO SET
4147	053322				FORCERROR 342\$	!DO FORCE ERROR IF FORCER=1
4148	053336	103407			BCS 350\$	!BR IF CARRY SET (GOOD RETURN)
4149	053340	010001			MOV R0,R1	!SAVE CONTENTS OF TSSR
4150	053342				NEXT,ERRNO	
4151	053342			342\$:	ERRDF ERRNO,T193SSR,PKTSSR	!DEVICE FATAL SSR FAILED TO SET
	053342	104455				TRAP C#ERDF
	053344	001457				.WORD 815
	053346	060064'				.WORD T193SSR
	053350	011656'				.WORD PKTSSR
4152	053352	005237	002222'		INC FATFLG	!SET FATAL ERROR FLAG
4153	053356			350\$:	CKLOOP	!LOOP ON ERROR, IF FLAG SET
	053356	104406				TRAP C#CLP1
4154	053360	004737	061760'		JSR PC,T19SETEXP	!SET WORDS 0-7 EXPD=RCV (NOT TESTING)
4155	053364	012701	057622'		MOV #T19EXSTA,R1	!GET EXPECTED READ STATUS
4156	053370	012702	062112'		MOV #T198FSTA,R2	!GET RCV READ STATUS
4157	053374	011211			MOV (R2),(R1)	!SET EXPD WORD #8 = RCV TEMP
4158	053376	042711	002000		BTC #S1,ICER,(R1)	!SET EXPD ICER =0
4159	053402	042711	001000		BIC #S1,IFMK,(R1)	!SET EXPD IFMK =0
4160	053406	042711	000400		BIC #S1,IHER,(R1)	!SET EXPD IHER =0
4161	053412	016261	000002 000002		MOV 2(R2),2(R1)	!SET EXPD WORD #9 = RCV (NOT TESTING)
4162	053420	005000			CLR R0	!HIGH RCV ADDRESS FOR CKMSG2
4163	053422	012701	062072'		MOV #T198FR,R1	!LOW RCV ADDRESS FOR CKMSG2
4164	053426	012702	057602'		MOV #T19EXP,R2	!EXPD ADDRESS
4165	053432	012703	000024		MOV #20,R3	!NUMBER OF BYTES TO COMPARE
4166	053436	004737	011310'		JSR PC,CKMSG2	!EXPD EQUAL RCV?
4167	053442				FORCERROR 362\$,NOTSSR	!DO
4168	053452	103404			BCS 370\$	!BR IF YES
4169	053454				NEXT,ERRNO	
4170	053454			362\$:	ERRHRD ERRNO,T197CMP,MSGSTAT	!REPORT ERROR

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

SEQ 175

```

    053454 104456
    053456 001460
    053460 061023'
    053462 012160'
4171 053464 104406
    053464 104406
4172
4173 053466 013737 002312' 057534'
4174 053474 013703 002316'
4175 053500 020327 003062'
4176 053504 103002
4177 053506 000137 052502'
4178 053512
4179
4180 053512
    053512
    053512 104403
4181
4182 053520 002222'
4183 053522
4184 053522 017014'
4185 053526
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220

370$: CKLDROP
    MOV DATA,T19PREV
    MOV TSTPTR,R3
    CMP R3,0TBLEND
    BHS 400$
    JMP 200$

400$:
    ENDSUB

460$:
    TST FATELG
    BEQ 460$
    JSR PC,CKDROP

;LOOP ON ERROR, IF FLAG SET
;SETUP PREVIOUS DATA FOR EXPD CALC.
;RESTORE CURRENT TSTBLK POINTER
;END OF TSTBLK?
;BR IF YES
;DO NEXT TSTBLK PATTERN

;//////////////// END SUBTEST //////////////////
L10067:
    TRAP C$ESUB

;ANY FATAL ERRORS ?
;BRANCH IF NOT
;TRY TO DROP THE UNIT

.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

```

```

4221      END
4222      ---
4223 053526      BGNSUB                      ;////////// BEGIN SUBTEST //////////
      053526      T8.2:
      053526 104402      TRAP      C$BSUB
4224      ;      Write to TSSR register to soft initialize the controller
4225 053530      5$:
4226 053530 004737 015604'      JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
4227 053534 103405      BCS      10$      ;BR IF SOFT INIT OKAY
4228 053536 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4229 053540      ERRDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
      053540 104455      TRAP      C$ERDF
      053542 001460      .WORD      816
      053544 003646'      .WORD      SFIERR
      053546 011644'      .WORD      SFIMSG
4230      ;      Do WRITE CHARACTERISTICS to check for Extended Features Switch
4231 053550 005037 002222'      10$:      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
4232 053554 012704 062050'      MOV      @T19PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
4233 053560 004737 010472'      JSR      PC,WRTCHR      ;DO WRITE CHARACTERISTICS COMMAND
4234 053564      FORCERROR      12$      ;DO FORCE ERROR IF FORCER=1
4235 053600 103407      BCS      15$      ;BR IF CARRY SET (GOOD RETURN)
4236 053602 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4237 053604      NEXT,ERRNO
4238 053604      12$:      ERRDF      ERRNO,T19SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      053604 104455      TRAP      C$ERDF
      053606 001461      .WORD      817
      053610 057763'      .WORD      T19SSR
      053612 011656'      .WORD      PKTSSR
4239 053614 005237 002222'      15$:      INC      FATFLG      ;SET FATAL ERROR FLAG
4240 053620      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      053620 104406      TRAP      C$CLP1
4241      ;      If Extended Features Hardware Switch Clear then:
4242      ;      Do Write Subsystem Write Miscellaneous to Set Extended Features.
4243 053622 012701 062072'      MOV      @T19BFR,R1      ;MESSAGE BUFFER ADDRESS
4244 053626 032761 000200 000012      BIT      @X2.EXTF,XST2(R1)      ;EXTENDED FEATURES SWITCH SET?
4245 053634 001026      BNE      30$      ;BR IF YES
4246 053636 004737 061722'      JSR      PC,T19SEXT      ;SETUP PACKET FOR WRITE MISC INVERT
4247 053642 012704 062220'      MOV      @T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
4248 053646 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
4249 053652 004737 016146'      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
4250 053656      FORCERROR      22$      ;DO FORCE ERROR IF FORCER=1
4251 053672 103407      BCS      30$      ;BR IF CARRY SET (GOOD RETURN)
4252 053674 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4253 053676      NEXT,ERRNO
4254 053676      22$:      ERRDF      ERRNO,T192SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      053676 104455      TRAP      C$ERDF
      053700 001462      .WORD      818
      053702 060020'      .WORD      T192SSR
      053704 011656'      .WORD      PKTSSR
4255 053706 005237 002222'      30$:      INC      FATFLG      ;SET FATAL ERROR FLAG
4256 053712      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      053712 104406      TRAP      C$CLP1
4257      ;      Do WRITE CHARACTERISTICS to select reserved unit 7
4258 053714 012704 062050'      MOV      @T19PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
4259 053720 004737 010472'      JSR      PC,WRTCHR      ;DO WRITE CHARACTERISTICS COMMAND
4260 053724      FORCERROR      42$      ;DO FORCE ERROR IF FORCER=1
4261 053740 103407      BCS      50$      ;BR IF CARRY SET (GOOD RETURN)

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 8: SUBTEST 2: LOOPBACK READ/WRITE SIGNALS TEST

SEQ 177

```

4262 053742 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4263 053744      NEXT,ERRNO
4264 053744      42$:  ERRDF  ERRNO,T19SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     819
                                .WORD     T19SSR
                                .WORD     PKTSSR
                                053744 104455
                                053746 001463
                                053750 057763'
                                053752 011656'
4265 053754 005237 002222'      INC      FATFLG      ;SET FATAL ERROR FLAG
4266 053760      50$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                053760 104406      TRAP      C$CLP1
4267
4268
4269      ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4270 053762 012703 002752'      MOV      @TSTBLK,R3      ;GET FIRST PATTERN ADDRESS
4271 053766 012337 002312'      100$:  MOV      (R3)+,DATA      ;GET A TEST PATTERN
4272 053772 042737 177400 002312'      BIC      @C<377>,DATA      ;DATA IS BYTE
4273 054000 010337 002316'      MOV      R3,TSTPTR      ;SETUP CURRENT TSTBLK POINTER
4274      ; Do a WRITE NPR to set loopback and tape direction OUT
4275 054004 012700 000100      MOV      @NPR.OUT,R0      ;SET TAPE DIRECTION OUT
4276 054010 052700 000040      BIS      @NPR.LOOP,R0      ;SET LOOPBACK
4277 054014 004737 061562'      JSR      PC,T19SNPR      ;SETUP T19PK2 FOR WRITE NPR
4278 054020 012704 062220'      MOV      @T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
4279 054024 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
4280 054030 004737 016146'      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
4281 054034      FORCERROR      102$      ;SSO FORCE ERROR IF FORCER=1
4282 054050 103407      BCS      105$      ;BR IF CARRY SET (GOOD RETURN)
4283 054052 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4284 054054      NEXT,ERRNO
4285 054054      102$:  ERRDF  ERRNO,T194SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     820
                                .WORD     T194SSR
                                .WORD     PKTSSR
                                054054 104455
                                054056 001464
                                054060 060131'
                                054062 011656'
4286 054064 005237 002222'      INC      FATFLG      ;SET FATAL ERROR FLAG
4287 054070      105$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                054070 104406      TRAP      C$CLP1
4288      ; Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4289 054072 012700 000001      MOV      @1,R0      ;WRITE 1 BYTE
4290 054076 012701 002312'      MOV      @DATA,R1      ;FIFO WRITE DATA ADDRESS
4291 054102 004737 061626'      JSR      PC,T19WFIF      ;SETUP T19PK2 FOR WRITE FIFO
4292 054106 012704 062220'      MOV      @T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
4293 054112 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
4294 054116 004737 016146'      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
4295 054122      FORCERROR      107$      ;SSO FORCE ERROR IF FORCER=1
4296 054136 103407      BCS      110$      ;BR IF CARRY SET (GOOD RETURN)
4297 054140 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4298 054142      NEXT,ERRNO
4299 054142      107$:  ERRDF  ERRNO,T195SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     821
                                .WORD     T195SSR
                                .WORD     PKTSSR
                                054142 104455
                                054144 001465
                                054146 060174'
                                054150 011656'
4300 054152 005237 002222'      INC      FATFLG      ;SET FATAL ERROR FLAG
4301 054156      110$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                054156 104406      TRAP      C$CLP1
4302      ; Do a READ FIFO with tape direction OUT to load tape out write latch
4303 054160 012700 000001      MOV      @1,R0      ;SET READ BYTE COUNT

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 8: SUBTEST 2: LOOPBACK READ/WRITE SIGNALS TEST

SEQ 178

4304	054164	004737	061606'	JSR	PC,T19RFIF	;SETUP T19PK2 FOR READ FIFO	
4305	054170	012704	062220'	MOV	#T19PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
4306	054174	010465	000000	MOV	R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
4307	054200	004737	016146'	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET	
4308	054204			FORCERROR	122\$	;GOODFORCE ERROR IF FORCER=1	
4309	054220	103407		BCS	130\$	;BR IF CARRY SET (GOOD RETURN)	
4310	054222	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR	
4311	054224			NEXT,ERRNO			
4312	054224			122\$:	ERRDF ERRNO,T196SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	
	054224	104455				TRAP	C\$ERDF
	054226	001466				.WORD	822
	054230	060240'				.WORD	T196SSR
	054232	011656'				.WORD	PKTSSR
4313	054234	005237	002222'	INC	FATFLG	;SET FATAL ERROR FLAG	
4314	054240			130\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	054240	104406				TRAP	C\$CLP1
4315				:	Do a WRITE NPR to set loopback and tape direction IN		
4316	054242	005000		CLR	R0	;CLR NP.OUT TO SET TAPE DIRECTION IN	
4317	054244	052700	000040'	BIS	#NP.LOOP,R0	;SET LOOPBACK	
4318	054250	004737	061562'	JSR	PC,T19SNPR	;SETUP T19PK2 FOR WRITE NPR	
4319	054254	012704	062220'	MOV	#T19PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
4320	054260	010465	000000	MOV	R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
4321	054264	004737	016146'	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET	
4322	054270			FORCERROR	142\$	;GOODFORCE ERROR IF FORCER=1	
4323	054304	103407		BCS	150\$	;BR IF CARRY SET (GOOD RETURN)	
4324	054306	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR	
4325	054310			NEXT,ERRNO			
4326	054310			142\$:	ERRDF ERRNO,T194SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	
	054310	104455				TRAP	C\$ERDF
	054312	001467				.WORD	823
	054314	060131'				.WORD	T194SSR
	054316	011656'				.WORD	PKTSSR
4327	054320	005237	002222'	INC	FATFLG	;SET FATAL ERROR FLAG	
4328	054324			150\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	054324	104406				TRAP	C\$CLP1
4329				:	Do a WRITE FIFO with byte count equal to 1 and Tape direction IN		
4330	054326	012700	000001	MOV	#1,R0	;WRITE 1 BYTE	
4331	054332	012701	002312'	MOV	#DATA,R1	;FIFO WRITE DATA ADDRESS	
4332	054336	004737	061626'	JSR	PC,T19WFIF	;SETUP T19PK2 FOR WRITE FIFO	
4333	054342	012704	062220'	MOV	#T19PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
4334	054346	010465	000000	MOV	R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
4335	054352	004737	016146'	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET	
4336	054356			FORCERROR	162\$	;GOODFORCE ERROR IF FORCER=1	
4337	054372	103407		BCS	170\$	;BR IF CARRY SET (GOOD RETURN)	
4338	054374	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR	
4339	054376			NEXT,ERRNO			
4340	054376			162\$:	ERRDF ERRNO,T195SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	
	054376	104455				TRAP	C\$ERDF
	054400	001470				.WORD	824
	054402	060174'				.WORD	T195SSR
	054404	011656'				.WORD	PKTSSR
4341	054406	005237	002222'	INC	FATFLG	;SET FATAL ERROR FLAG	
4342	054412			170\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	054412	104406				TRAP	C\$CLP1
4343				:	Do a READ FIFO with tape direction IN to read data		
4344				:	If Data read from FIFO NOT= to Data sent Then Print Error		
4345	054414	012700	000001	MOV	#1,R0	;SET READ BYTE COUNT	

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 8: SUBTEST 2: LOOPBACK READ/WRITE SIGNALS TEST

SEQ 179

4346	054420	004737	061606'	JSR	PC,T19RFIF	;	SETUP T19PK2 FOR READ FIFO
4347	054424	012704	062220'	MOV	#T19PK2,R4	;	GET WRITE SUBSYSTEM COMMAND PACKET
4348	054430	010465	000000	MOV	R4,TSDB(R5)	;	SET THE PACKET ADDRESS TO EXECUTE
4349	054434	004737	016146'	JSR	PC,CHKTSSR	;	WAIT FOR SSR TO SET
4350	054440			FORCERROR	182\$	;	FORCE ERROR IF FORCER=1
4351	054454	103407		BCS	190\$	;	BR IF CARRY SET (GOOD RETURN)
4352	054456	010001		MOV	RO,R1	;	SAVE CONTENTS OF TSSR
4353	054460			NEXT,ERRNO			
4354	054460			182\$:	ERRDF ERRNO,T196SSR,PKTSSR	;	DEVICE FATAL SSR FAILED TO SET
	054460	104455				TRAP	C\$ERDF
	054462	001471				.WORD	825
	054464	060240'				.WORD	T196SSR
	054466	011656'				.WORD	PKTSSR
4355	054470	005237	002222'	INC	FATFLG	;	SET FATAL ERROR FLAG
4356	054474			190\$:	CKLOOP	;	LOOP ON ERROR, IF FLAG SET
	054474	104406				TRAP	C\$CLP1
4357	054476	004737	061760'	JSR	PC,T19SETEXP	;	SET WORDS 0-7 EXPD=RCV (NOT TESTING)
4358	054502	012701	057622'	MOV	#T19EXSTA,R1	;	GET EXPECTED READ STATUS
4359	054506	012702	062112'	MOV	#T198FSTA,R2	;	GET RCV READ STATUS
4360	054512	013711	002312'	MOV	DATA,(R1)	;	SET EXPD WORD #8 = DATA
4361	054516	016261	000002 000002	MOV	2(R2),2(R1)	;	SET EXPD WORD #9 = RCV (NOT TESTING)
4362	054524	005000		CLR	RO	;	HIGH RCV ADDRESS FOR CKMSG2
4363	054526	012701	062072'	MOV	#T198FR,R1	;	LOW RCV ADDRESS FOR CKMSG2
4364	054532	012702	057602'	MOV	#T19EXP,R2	;	EXPD ADDRESS
4365	054536	012703	000022	MOV	#18.,R3	;	NUMBER OF BYTES TO COMPARE
4366	054542	004737	011310'	JSR	PC,CKMSG2	;	EXPD EQUAL RCV?
4367	054546			FORCERROR	202\$,NOTSSR	;	NO
4368	054556	103404		BCS	210\$	;	BR IF YES
4369	054560			NEXT,ERRNO			
4370	054560			202\$:	ERRHRD ERRNO,T199CMP,MSGSUB	;	REPORT ERROR
	054560	104456				TRAP	C\$ERHRD
	054562	001472				.WORD	826
	054564	061200				.WORD	T199CMP
	054566	013852'				.WORD	MSGSUB
4371	054570			210\$:	CKLOOP	;	LOOP ON ERROR, IF FLAG SET
	054570	104406				TRAP	C\$CLP1
4372				;	Do a Write Subsystem READ STATUS		
4373				;	If Input Ready NOT=1 Then Print Error		
4374				;	If Output Ready NOT=0 Then Print Error		
4375				;	If Data In Miss NOT=0 Then Print Error		
4376	054572	004737	061520'	JSR	PC,T19SRD	;	SETUP PACKET FOR READ STATUS
4377	054576	012704	062220'	MOV	#T19PK2,R4	;	GET WRITE SUBSYSTEM COMMAND PACKET
4378	054602	010465	000000	MOV	R4,TSDB(R5)	;	SET THE PACKET ADDRESS TO EXECUTE
4379	054606	004737	016146'	JSR	PC,CHKTSSR	;	WAIT FOR SSR TO SET
4380	054612			FORCERROR	212\$	;	FORCE ERROR IF FORCER=1
4381	054626	103407		BCS	220\$	;	BR IF CARRY SET (GOOD RETURN)
4382	054630	010001		MOV	RO,R1	;	SAVE CONTENTS OF TSSR
4383	054634			NEXT,ERRNO			
4384	054634			212\$:	ERRDF ERRNO,T193SSR,PKTSSR	;	DEVICE FATAL SSR FAILED TO SET
	054634	104455				TRAP	C\$ERDF
	054634	001473				.WORD	827
	054636	060064'				.WORD	T193SSR
	054640	011656'				.WORD	PKTSSR
4385	054642	005237	002222'	INC	FATFLG	;	SET FATAL ERROR FLAG
4386	054646			220\$:	CKLOOP	;	LOOP ON ERROR, IF FLAG SET
	054646	104406				TRAP	C\$CLP1
4387	054650	004737	061760'	JSR	PC,T19SETEXP	;	SET WORDS 0-7 EXPD=RCV (NOT TESTING)

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 8: SUBTEST 2: LOOPBACK READ/WRITE SIGNALS TEST

SEQ 180

```

4388 054654 012701 057622'      MOV      T19EXSTA,R1      ;GET EXPECTED READ STATUS
4389 054660 012702 062112'      MOV      T19BFSTA,R2      ;GET RECV READ STATUS
4390 054664 012221                MOV      (R2)+,(R1)+      ;SET EXPD WORD #8 = RECV TEMP
4391 054666 011211                MOV      (R2),(R1)        ;SET EXPD WORD #9 = RECV TEMP
4392 054670 052711 000020        BIS      S2.INRDY,(R1)    ;SET EXP INPUT READY = 1
4393 054674 042711 000040        BIC      S2.OTRDY,(R1)    ;SET EXP OUTPUT READY = 0
4394 054700 042711 000200        BIC      S2.DIM,(R1)    ;SET EXP DATA IN MISS = 0
4395 054704 005000                CLR      R0            ;HIGH RECV ADDRESS FOR CKMSG2
4396 054706 012701 062072'      MOV      T19BFR,R1        ;LOW RECV ADDRESS FOR CKMSG2
4397 054712 012702 057602'      MOV      T19EXP,R2        ;EXPD ADDRESS
4398 054716 012703 000024        MOV      #20,R3          ;NUMBER OF BYTES TO COMPARE
4399 054722 004737 011310'      JSR      PC,CKMSG2        ;EXPD EQUAL RECV?
4400 054726                FORCERROR 232$,NOTSSR      ;880
4401 054736 103404                BCS      240$          ;BR IF YES
4402 054740                NEXT.ERRNO
4403 054740                232$:  ERRHRD  ERRNO,T196CMP,MSGSTAT ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD     828
                                .WORD     T196CMP
                                .WORD     MSGSTAT
                                054740 104456
                                054742 001474
                                054744 060740'
                                054746 012160'
4404 054750                240$:  CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                054750 104406                TRAP      C$CLP1
4405
4406
4407
4408 054752                FORCEEXIT 255$                ;880
4409 054762 013703 002316'      MOV      TSTPTR,R3        ;RESTORE CURRENT TSTBLK POINTER
4410 054766 020327 003062'      CMP      R3,TBLEND        ;END OF TSTBLK?
4411 054772 103002                BHS      255$          ;BR IF YES
4412 054774 000137 053766'      JMP      100$            ;DO ANOTHER TSTBLK PATTERN
4413 055000                255$:
4414
4415 055000                ENDSUB                          ;////////// END SUBTEST ///////////
                                L10070:
                                TRAP      C$ESUB
                                055000 104403
4416
4417 055002 005737 002222'      TST      FATELG        ;ANY FATAL ERRORS ?
4418 055006 001402                BEQ      260$          ;BRANCH IF NOT
4419 055010 004737 017014'      JSR      PC,CKDROP        ;TRY TO DROP THE UNIT
4420 055014                260$:
4421
4422                .SBTTL  TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST
4423
4424                ;**
4425                ; TEST 8: SUBTEST 3:
4426                ;
4427                ; SUBTEST DESCRIPTION:
4428                ;
4429                ; This subtest verifies the Write Strobe loopback path
4430                ; can strobe data from the FIFO to the Data lines.
4431                ; The signal IRESV3 drives IWSTR (write strobe) to write
4432                ; data from the FIFO to the tape data out latch.
4433                ;
4434                ; TEST STEPS:
4435                ;
4436                ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4437

```

```

4438      ; BEGIN
4439      ; Write to TSSR register to soft initialize the controller
4440      ; Do WRITE CHARACTERISTICS to check for Extended Features Switch
4441      ; If Extended Features Hardware Switch Clear then:
4442      ;   Do Write Subsystem Write Miscellaneous to Set Extended Features.
4443      ; Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
4444      ; Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
4445      ; Do a WRITE NPR to set loopback and tape direction OUT
4446      ; Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
4447      ; Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4448      ; Do a WRITE FORMAT to set IRESV3==>IWSTR = 0 to load write data latch
4449      ; Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
4450      ; Do a WRITE NPR to set loopback and tape direction IN
4451      ; Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
4452      ;   to strobe loopback data into FIFO.
4453      ; Do a READ FIFO with tape direction IN to read data
4454      ; If Data read from FIFO NOT= to Data sent Then Print Error
4455      ; END
4456      ;--
4457      055014      BGNSUB                      ;////////// BEGIN SUBTEST //////////
4458      055014      104402                      T8.3: TRAP C$BSUB
4459      5$:      ; Write to TSSR register to soft initialize the controller
4460      055016      004737 015604'      JSR PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
4461      055022      103405      BCS 10$      ;BR IF SOFT INIT OKAY
4462      055024      010001      MOV R0,R1      ;SAVE CONTENTS OF TSSR
4463      055026      055026 104455      ERDIF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
4464      055030      001474      TRAP C$ERDF
4465      055032      003646'      WORD 828
4466      055034      011644'      WORD SFIERR
4467      055036      005037 002222'      Do WRITE CHARACTERISTICS to check for Extended Features Switch
4468      10$:      CLR FATFLG      ;CLEAR FATAL ERROR FLAG
4469      055042      012704 062050'      MOV #T19PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
4470      055046      004737 010472'      JSR PC,WRTCHR      ;DO WRITE CHARACTERISTICS
4471      055052      103407      FORCERROR 12$      ;GOODFORCE ERROR FLAG=1
4472      055066      010001      BCS 15$      ;BR IF CARRY SET (GOOD RETURN)
4473      055070      010001      MOV R0,R1      ;SAVE CONTENTS OF TSSR
4474      055072      104455      NEXT,ERRNO
4475      12$:      ERDIF ERRNO,T19SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
4476      055074      001475      TRAP C$ERDF
4477      055076      057763'      WORD 82
4478      055100      011656'      WORD T19SSR
4479      055102      005237 002222'      INC FATFLG      ;SET FATAL ERROR FLAG
4480      055106      104406      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
4481      055106      104406      TRAP C$CLP1
4482      ; If Extended Features Hardware Switch Clear then:
4483      ; Do Write Subsystem Write Miscellaneous to Set Extended Features.
4484      055110      012701 062072'      MOV #T19BFR,R1      ;MESSAGE BUFFER ADDRESS
4485      055114      032761 000200 000012      BIT #X2,EXTF,XST2(R1) ;EXTENDED FEATURES SWITCH SET?
4486      055122      001026      BNE 30$      ;BR IF YES
4487      055124      004737 061722'      JSR PC,T19SEXT      ;SETUP PACKET FOR WRITE MISC INVERT
4488      055130      012704 062220'      MOV #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
4489      055134      010465 000000      MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
4490      055140      004737 016146'      JSR PC,CHKTSSR ;WAIT FOR SSR TO SET

```

TSVS - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

SEQ 182

```

4484 055144          FORCERROR          22$          ;00DFORCE ERROR IF FORCER=1
4485 055160 103407   BCS          30$          ;BR IF CARRY SET (GOOD RETURN)
4486 055162 010001   MOV          R0,R1          ;SAVE CONTENTS OF TSSR
4487 055164          NEXT,ERRNO
4488 055164          22$:  ERPDF  ERRNO,T192SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP          C$ERDF
                                .WORD          830
                                .WORD          T192SSR
                                .WORD          PKTSSR
                                055166 001476
                                055170 060020'
                                055172 011656'
4489 055174 005237 002222'   INC          FATFLG          ;SET FATAL ERROR FLAG
4490 055200          30$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP          C$CLP1
                                055200 104406
; Do WRITE CHARACTERISTICS to select reserved unit 7
4491          :
4492 055202 012704 062050'   MOV          #T19PACKET,R4          ;GET THE ADDRESS OF COMMAND PACKET
4493 055206 004737 010472'   JSR          PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
4494 055210          FORCERROR          42$          ;00DFORCE ERROR IF FORCER=1
4495 055226 103407   BCS          50$          ;BR IF CARRY SET (GOOD RETURN)
4496 055230 010001   MOV          R0,R1          ;SAVE CONTENTS OF TSSR
4497 055232          NEXT,ERRNO
4498 055232          42$:  ERPDF  ERRNO,T19SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP          C$ERDF
                                .WORD          831
                                .WORD          T19SSR
                                .WORD          PKTSSR
                                055232 104455
                                055234 001477
                                055236 057763'
                                055240 011656'
4499 055242 005237 002222'   INC          FATFLG          ;SET FATAL ERROR FLAG
4500 055246          50$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP          C$CLP1
                                055246 104406
4501          :
4502          : REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4503 055250 012703 002752'   MOV          #TSTBLK,R3          ;GET FIRST PATTERN ADDRESS
4504 055254 012337 002312'   100$:  MOV          (R3)+,DATA          ;GET A TEST PATTERN
4505 055260 042737 177400 002312'   BIC          #C<377>,DATA          ;DATA IS BYTE
4506 055266 010337 002316'   MOV          R3,TSTPTR          ;SETUP CURRENT TSTBLK POINTER
4507          :
4508 055272 012700 000100   Do a WRITE NPR to set loopback and tape direction OUT
4509 055276 052700 000040   MOV          #NP.OUT,R0          ;SET TAPE DIRECTION OUT
4510 055302 004737 061562'   BIS          #NP.LOOP,R0          ;SET LOOPBACK
4511 055306 012704 062220'   JSR          PC,T19SNPR          ;SETUP T19PK2 FOR WRITE NPR
4512 055312 010465 000000   MOV          #T19PK2,R4          ;GET WRITE SUBSYSTEM COMMAND PACKET
4513 055316 004737 016146'   MOV          R4,TSD8(R5)          ;SET THE PACKET ADDRESS TO EXECUTE
4514 055322          JSR          PC,CHKTSSR          ;WAIT FOR SSR TO SET
4515 055336 103407   FORCERROR          102$          ;00DFORCE ERROR IF FORCER=1
4516 055340 010001   BCS          105$          ;BR IF CARRY SET (GOOD RETURN)
4517 055342          MOV          R0,R1          ;SAVE CONTENTS OF TSSR
4518 055342          102$:  ERPDF  ERRNO,T194SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP          C$ERDF
                                .WORD          832
                                .WORD          T194SSR
                                .WORD          PKTSSR
                                055342 104455
                                055344 001500
                                055346 060131'
                                055350 011656'
4519 055352 005237 002222'   INC          FATFLG          ;SET FATAL ERROR FLAG
4520 055356          105$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP          C$CLP1
                                055356 104406
4521          :
4522 055360 012700 000002   Do a WRITE FORMAT to set IRESV3-->IWSR = 1
4523 055364 004737 061702'   MOV          #WF.I3RES,R0          ;IRESV3-->IWSR=1
4524 055370 012704 062220'   JSR          PC,T19WFMT          ;SETUP T19PK2 FOR WRITE FORMAT
4525 055374 010465 000000   MOV          #T19PK2,R4          ;GET WRITE SUBSYSTEM COMMAND PACKET
                                MOV          R4,TSD8(R5)          ;SET THE PACKET ADDRESS TO EXECUTE

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

SEQ 183

4526	055400	004737	016146'	JSR	PC,CHKTSSR	WAIT FOR SSR TO SET
4527	055404			FORCERROR	1121	IFCEROR FORCE ERROR IF FORCER=1
4528	055420	103407		BCS	1201	IFCEROR FORCE ERROR IF FORCER=1
4529	055422	010001		MOV	R0,R1	IFCEROR FORCE ERROR IF FORCER=1
4530	055424			NEXT,ERRNO		IFCEROR FORCE ERROR IF FORCER=1
4531	055424			ERRDF	ERRNO,T198SSR,PKTSSR	IFCEROR FORCE ERROR IF FORCER=1
	055424	104455				IFCEROR FORCE ERROR IF FORCER=1
	055426	001501				IFCEROR FORCE ERROR IF FORCER=1
	055430	060352'				IFCEROR FORCE ERROR IF FORCER=1
	055432	011656'				IFCEROR FORCE ERROR IF FORCER=1
4532	055434	005237	002222'	INC	FATFLG	IFCEROR FORCE ERROR IF FORCER=1
4533	055440			CKLOOP		IFCEROR FORCE ERROR IF FORCER=1
	055440	104406				IFCEROR FORCE ERROR IF FORCER=1
4534						IFCEROR FORCE ERROR IF FORCER=1
4535	055442	012700	000001	Do a WRITE FIFO with byte count	equal to 1 and Tape direction OUT	IFCEROR FORCE ERROR IF FORCER=1
4536	055446	012701	002312'	MOV	#1,R0	IFCEROR FORCE ERROR IF FORCER=1
4537	055452	004737	061626'	MOV	#DATA,R1	IFCEROR FORCE ERROR IF FORCER=1
4538	055456	012704	062220'	JSR	PC,T19WFIF	IFCEROR FORCE ERROR IF FORCER=1
4539	055462	010465	000000	MOV	#T19PK2,R4	IFCEROR FORCE ERROR IF FORCER=1
4540	055466	004737	016146'	MOV	R4,TSD8(R5)	IFCEROR FORCE ERROR IF FORCER=1
4541	055472			JSR	PC,CHKTSSR	IFCEROR FORCE ERROR IF FORCER=1
4542	055506	103407		FORCERROR	1321	IFCEROR FORCE ERROR IF FORCER=1
4543	055510	010001		BCS	1401	IFCEROR FORCE ERROR IF FORCER=1
4544	055512			MOV	R0,R1	IFCEROR FORCE ERROR IF FORCER=1
4545	055512			NEXT,ERRNO		IFCEROR FORCE ERROR IF FORCER=1
	055512	104455		ERRDF	ERRNO,T195SSR,PKTSSR	IFCEROR FORCE ERROR IF FORCER=1
	055514	001502				IFCEROR FORCE ERROR IF FORCER=1
	055516	060174'				IFCEROR FORCE ERROR IF FORCER=1
	055520	011656'				IFCEROR FORCE ERROR IF FORCER=1
4546	055522	005237	002222'	INC	FATFLG	IFCEROR FORCE ERROR IF FORCER=1
4547	055526			CKLOOP		IFCEROR FORCE ERROR IF FORCER=1
	055526	104406				IFCEROR FORCE ERROR IF FORCER=1
4548						IFCEROR FORCE ERROR IF FORCER=1
4549	055530	005000		Do a WRITE FORMAT to set IRESV3-->IWSTR = 0		IFCEROR FORCE ERROR IF FORCER=1
4550	055532	004737	061702'	CLR	R0	IFCEROR FORCE ERROR IF FORCER=1
4551	055536	012704	062220'	JSR	PC,T19WFMT	IFCEROR FORCE ERROR IF FORCER=1
4552	055542	010465	000000	MOV	#T19PK2,R4	IFCEROR FORCE ERROR IF FORCER=1
4553	055546	004737	016146'	MOV	R4,TSD8(R5)	IFCEROR FORCE ERROR IF FORCER=1
4554	055552			JSR	PC,CHKTSSR	IFCEROR FORCE ERROR IF FORCER=1
4555	055566	103407		FORCERROR	1521	IFCEROR FORCE ERROR IF FORCER=1
4556	055570	010001		BCS	1601	IFCEROR FORCE ERROR IF FORCER=1
4557	055572			MOV	R0,R1	IFCEROR FORCE ERROR IF FORCER=1
4558	055572			NEXT,ERRNO		IFCEROR FORCE ERROR IF FORCER=1
	055572	104455		ERRDF	ERRNO,T198SSR,PKTSSR	IFCEROR FORCE ERROR IF FORCER=1
	055574	001503				IFCEROR FORCE ERROR IF FORCER=1
	055576	060352'				IFCEROR FORCE ERROR IF FORCER=1
	055600	011656'				IFCEROR FORCE ERROR IF FORCER=1
4559	055602	005237	002222'	INC	FATFLG	IFCEROR FORCE ERROR IF FORCER=1
4560	055606			CKLOOP		IFCEROR FORCE ERROR IF FORCER=1
	055606	104406				IFCEROR FORCE ERROR IF FORCER=1
4561						IFCEROR FORCE ERROR IF FORCER=1
4562	055610	012700	000002	Do a WRITE FORMAT to set IRESV3-->IWSTR = 1		IFCEROR FORCE ERROR IF FORCER=1
4563	055614	004737	061702'	MOV	#IRESV3,R0	IFCEROR FORCE ERROR IF FORCER=1
4564	055620	012704	062220'	JSR	PC,T19WFMT	IFCEROR FORCE ERROR IF FORCER=1
4565	055624	010465	000000	MOV	#T19PK2,R4	IFCEROR FORCE ERROR IF FORCER=1
4566	055630	004737	016146'	MOV	R4,TSD8(R5)	IFCEROR FORCE ERROR IF FORCER=1
4567	055634			JSR	PC,CHKTSSR	IFCEROR FORCE ERROR IF FORCER=1
				FORCERROR	1721	IFCEROR FORCE ERROR IF FORCER=1

TSVS - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

SEQ 184

4568	055650	103407		BCS	180:		IBR IF CARRY SET (GOOD RETURN)
4569	055652	010001		MOV	RO,R1		ISAVE CONTENTS OF TSSR
4570	055654			NEXT,ERRNO			
4571	055654		172:	ERRDF	ERRNO,T198SSR,PKTSSR		IDEVICE FATAL SSR FAILED TO SET
	055654	104455				TRAP	C:ERDF
	055656	001504				.WORD	836
	055660	060352				.WORD	T198SSR
	055662	011656				.WORD	PKTSSR
4572	055664	005237	002222	INC	FATFLG		ISSET FATAL ERROR FLAG
4573	055670			CKLOOP			ILLOOP ON ERROR, IF FLAG SET
	055670	104406				TRAP	C:CLP1
4574							
4575							
4576	055672	005000			Do a WRITE NPR to set loopback and tape direction IN		
4577	055674	052700	000040	CLR	RO		ICLR NP.OUT TO SET TAPE DIRECTION IN
4578	055700	004737	061562	BIS	#NP.LOOP,RO		ISSET LOOPBACK
4579	055704	012704	062220	JSR	PC,T19SNPR		ISSETUP T19PK2 FOR WRITE NPR
4580	055710	010465	000000	MOV	#T19PK2,R4		IGET WRITE SUBSYSTEM COMMAND PACKET
4581	055714	004737	016146	MOV	R4,TSDB(R5)		ISSET THE PACKET ADDRESS TO EXECUTE
4582	055720			JSR	PC,CHKTSSR		IWAIT FOR SSR TO SET
4583	055734	103407		FORCERROR	182:		IBBDFORCE ERROR IF FORCER=1
4584	055736	010001		BCS	190:		IBR IF CARRY SET (GOOD RETURN)
4585	055740			MOV	RO,R1		ISAVE CONTENTS OF TSSR
4586	055740			NEXT,ERRNO			
	055740	104455		182:	ERRDF	ERRNO,T194SSR,PKTSSR	IDEVICE FATAL SSR FAILED TO SET
	055742	001505				TRAP	C:ERDF
	055744	060131				.WORD	837
	055746	011656				.WORD	T194SSR
						.WORD	PKTSSR
4587	055750	005237	002222	INC	FATFLG		ISSET FATAL ERROR FLAG
4588	055754			CKLOOP			ILLOOP ON ERROR, IF FLAG SET
	055754	104406				TRAP	C:CLP1
4589							
4590	055756	012700	000001		Do a WRITE FIFO with byte count equal to 1 and Tape direction IN		
4591	055762	012701	002312	MOV	#1,RO		IWRITE 1 BYTE
4592	055766	004737	061626	MOV	#DATA,R1		IFIFO WRITE DATA ADDRESS
4593	055772	012704	062220	JSR	PC,T19WFIF		ISSETUP T19PK2 FOR WRITE FIFO
4594	055776	010465	000000	MOV	#T19PK2,R4		IGET WRITE SUBSYSTEM COMMAND PACKET
4595	056002	004737	016146	MOV	R4,TSDB(R5)		ISSET THE PACKET ADDRESS TO EXECUTE
4596	056006			JSR	PC,CHKTSSR		IWAIT FOR SSR TO SET
4597	056022	103407		FORCERROR	202:		IBBDFORCE ERROR IF FORCER=1
4598	056024	010001		BCS	210:		IBR IF CARRY SET (GOOD RETURN)
4599	056026			MOV	RO,R1		ISAVE CONTENTS OF TSSR
4600	056026			NEXT,ERRNO			
	056026	104455		202:	ERRDF	ERRNO,T195SSR,PKTSSR	IDEVICE FATAL SSR FAILED TO SET
	056030	001506				TRAP	C:ERDF
	056032	060174				.WORD	838
	056034	011656				.WORD	T195SSR
						.WORD	PKTSSR
4601	056036	005237	002222	INC	FATFLG		ISSET FATAL ERROR FLAG
4602	056042			CKLOOP			ILLOOP ON ERROR, IF FLAG SET
	056042	104406				TRAP	C:CLP1
4603							
4604	056044	012700	000001		Do a READ FIFO with tape direction IN to read data		
4605	056050	004737	061606	MOV	#1,RO		ISSET READ BYTE COUNT
4606	056054	012704	062220	JSR	PC,T19RFIF		ISSETUP T19PK2 FOR READ FIFO
4607	056060	010465	000000	MOV	#T19PK2,R4		IGET WRITE SUBSYSTEM COMMAND PACKET
4608	056064	004737	016146	MOV	R4,TSDB(R5)		ISSET THE PACKET ADDRESS TO EXECUTE
4609	056070			JSR	PC,CHKTSSR		IWAIT FOR SSR TO SET
				FORCERROR	222:		IBBDFORCE ERROR IF FORCER=1

TSV5 - HARDWARE TESTS MACRO M1113 01 FEB 84 17:02
 TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

SEQ 185

```

4610 056104 103407      BCS      230$      ;BR IF CARRY SET (GOOD RETURN)
4611 056106 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4612 056110              NEXT.ERRNO
4613 056110              222$: ERRDF  ERRNO,T196SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     839
                                .WORD     T196SSR
                                .WORD     PKTSSR
4614 056120 005237 002222$ INC      FATFLG      ;SET FATAL ERROR FLAG
4615 056124 104406      230$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
4616              ; If Data read from FIFO NOT= to Data sent Then Print Error
4617 056126 004737 061760$ JSR      PC,T19SETEXP ;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
4618 056132 012701 057622$ MOV      @T19EXSTA,R1 ;GET EXPECTED READ STATUS
4619 056136 012702 062112$ MOV      @T198FSTA,R2 ;GET RCV READ STATUS
4620 056142 013711 002312$ MOV      DATA,(R1) ;SET EXPD WORD #8 = DATA
4621 056146 016261 000002 000002$ MOV      2(R2),2(R1) ;SET EXPD WORD #9 = RCV (NOT TESTING)
4622 056154 005000      CLR      R0 ;HIGH RCV ADDRESS FOR CKMSG2
4623 056156 012701 062072$ MOV      @T198FR,R1 ;LOW RCV ADDRESS FOR CKMSG2
4624 056162 012702 057602$ MOV      @T19EXP,R2 ;EXPD ADDRESS
4625 056166 012703 000022$ MOV      @18.,R3 ;NUMBER OF BYTES TO COMPARE
4626 056172 004737 011310$ JSR      PC,CKMSG2 ;EXPD EQUAL RCV?
4627 056176      FORCERROR 242$,NOTSSR ;NO
4628 056206 103404      BCS      250$      ;BR IF YES
4629 056210      NEXT.ERRNO
4630 056210              242$: ERRHRD  ERRNO,T19WSTR,MSGSUB ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD     840
                                .WORD     T19WSTR
                                .WORD     MSGSUB
4631 056220 104406      250$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
4632
4633
4634 056222      FORCEEXIT 255$      ;NO
4635 056232 013703 002316$ MOV      TSTPTR,R3 ;RESTORE CURRENT TSTBLK POINTER
4636 056236 020327 003062$ CMP      R3,@TBLND ;END OF TSTBLK?
4637 056242 103002      BHS      255$      ;BR IF YES
4638 056244 000137 055254$ JMP      100$      ;DO ANOTHER TSTBLK PATTERN
4639 056250              255$:
4640
4641 056250      ENDSUB      ;////////// END SUBTEST //////////
                                L10071:
                                TRAP      C$ESUB
4642
4643 056252 005737 002222$ TST      FATFLG      ;ANY FATAL ERRORS ?
4644 056256 001402      BEQ      260$      ;BRANCH IF NOT
4645 056260 004737 017014$ JSR      PC,CKDROP ;TRY TO DROP THE UNIT
4646
4647
4648      .SBTTL  TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST
4649
4650
4651      ;
4652      ; TEST A: SUBTEST 4:
4653      ; SUBTEST DESCRIPTION:
4654      ;

```

```

4655      | This subtest verifies the Read Strobe loopback path
4656      | can strobe the data from the Data lines to the FIFO.
4657      | The signal IRESV4 drives IRSTR (read strobe) to write
4658      | from the data lines to the FIFO.
4659      |
4660      | TEST STEPS:
4661      |
4662      | REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4663      | BEGIN
4664      |   Write to TSSR register to soft initialize the controller
4665      |   Do WRITE CHARACTERISTICS to check for Extended Features Switch
4666      |   If Extended Features Hardware Switch Clear then:
4667      |     Do Write Subsystem Write Miscellaneous to Set Extended Features.
4668      |   Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
4669      |   Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
4670      |   Do a WRITE NPR to set loopback and tape direction OUT
4671      |   Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
4672      |   Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
4673      |   Do a READ FIFO with tape direction OUT to load tape out write latch
4674      |   Do a WRITE NPR to set loopback and tape direction IN
4675      |   Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 to write loop data to FIFO
4676      |   Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
4677      |     (to strobe loopback data into FIFO.)
4678      |   Do a READ FIFO with tape direction IN to read data
4679      |   If Data read from FIFO NOT= to Data sent Then Print Error
4680      | END
4681      |
4682      |
4683      | BGNSUB                      ;////////// BEGIN SUBTEST //////////
4684      |                               T8.4:
4685      |                               TRAP    C#BSUB
4686      |   Write to TSSR register to soft initialize the controller
4687      | 5$:
4688      |   JSR    PC,SOFINIT          ;WRITE TO TSSR TO SOFT INITIALIZE
4689      |   BCS    10$                ;BR IF SOFT INIT OKAY
4690      |   MOV    R0,R1              ;SAVE CONTENTS OF TSSR
4691      |   ERROF  ERRNO,SFIERR,SFMSG ;DEVICE FATAL DURING INIT
4692      |                               TRAP    C#EROF
4693      |                               .WORD    840
4694      |                               .WORD    SFIERR
4695      |                               .WORD    SFMSG
4696      |   Do WRITE CHARACTERISTICS to check for Extended Features Switch
4697      | 10$:
4698      |   CLR    FATFLG              ;CLEAR FATAL ERROR FLAG
4699      |   MOV    @T19PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
4700      |   JSR    PC,WRICHR          ;DO WRITE CHARACTERISTICS COMMAND
4701      |   FORCERR 12$              ;DO FORCE ERROR IF FORCER=1
4702      |   BCS    15$                ;BR IF CARRY SET (GOOD RETURN)
4703      |   MOV    R0,R1              ;SAVE CONTENTS OF TSSR
4704      |   NEXT ERRNO
4705      | 12$:
4706      |   ERROF  ERRNO,T19SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
4707      |                               TRAP    C#EROF
4708      |                               .WORD    841
4709      |                               .WORD    T19SSR
4710      |                               .WORD    PKTSSR
4711      |   INC    FATFLG              ;SET FATAL ERROR FLAG
4712      | 15$:
4713      |   CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
4714      |                               TRAP    C#CLP1

```

```

4701.                                     ; If Extended Features Hardware Switch Clear then:
4702                                     ; Do Write Subsystem Write Miscellaneous to Set Extended Features.
4703 056360 012701 062072'                MOV    T19BFR,R1                ;MESSAGE BUFFER ADDRESS
4704 056364 032761 000200 000012          BIT    X2.EXTF,XST2(R1)          ;EXTENDED FEATURES SWITCH SET?
4705 056372 001026                        BNE    30$                    ;BR IF YES
4706 056374 004737 061722'                JSR    PC,T19SEXT              ;SETUP PACKET FOR WRITE MISC INVERT
4707 056400 012704 062220'                MOV    T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4708 056404 010465 000000                  MOV    R4,TSD8(R5)            ;SET THE PACKET ADDRESS TO EXECUTE
4709 056410 004737 016146'                JSR    PC,CHKTSSR              ;WAIT FOR SSR TO SET
4710 056414                                FORCERROR 22$                ;GOODFORCE ERROR IF FORCER=1
4711 056430 103407                        BCS    30$                    ;BR IF CARRY SET (GOOD RETURN)
4712 056432 010001                        MOV    R0,R1                    ;SAVE CONTENTS OF TSSR
4713 056434                                NEXT,ERRNO
4714 056434 104455                        22$: ERRDF  ERRNO,T192SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                         TRAP    C$ERDF
                                         .WORD    842
                                         .WORD    T192SSR
                                         .WORD    PKTSSR
                                         TRAP    C$CLP1
056434 104455
056436 001512
056440 060020'
056442 011656'
4715 056444 005237 002222'                INC    FATFLG                ;SET FATAL ERROR FLAG
4716 056450 104406                        30$: CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
                                         TRAP    C$CLP1
056450 104406
4717                                     ; Do WRITE CHARACTERISTICS to select reserved unit 7
4718 056452 012704 062050'                MOV    T19PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
4719 056456 004737 010472'                JSR    PC,WRTHR              ;DO WRITE CHARACTERISTICS COMMAND
4720 056462                                FORCERROR 42$                ;GOODFORCE ERROR IF FORCER=1
4721 056476 103407                        BCS    50$                    ;BR IF CARRY SET (GOOD RETURN)
4722 056500 010001                        MOV    R0,R1                    ;SAVE CONTENTS OF TSSR
4723 056502                                NEXT,ERRNO
4724 056502 104455                        42$: ERRDF  ERRNO,T19SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                         TRAP    C$ERDF
                                         .WORD    843
                                         .WORD    T19SSR
                                         .WORD    PKTSSR
                                         TRAP    C$CLP1
056502 104455
056504 001513
056506 057763'
056510 011656'
4725 056512 005237 002222'                INC    FATFLG                ;SET FATAL ERROR FLAG
4726 056516 104406                        50$: CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
                                         TRAP    C$CLP1
056516 104406
4727
4728                                     ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
4729 056520 012703 002752'                MOV    TSTBLK,R3                ;GET FIRST PATTERN ADDRESS
4730 056524 012337 002312'                100$: MOV    (R3),DATA            ;GET A TEST PATTERN
4731 056530 042737 177400 002312'          BIC    C<377>,DATA            ;DATA IS BYTE
4732 056536 010337 002316'                MOV    R3,TSTPTR              ;SETUP CURRENT TSTBLK POINTER
4733                                     ; Do a WRITE NPR to set loopback and tape direction OUT
4734 056542 012700 000100                  MOV    TNP.OUT,R0                ;SET TAPE DIRECTION OUT
4735 056546 052700 000040                  BIS    TNP.LOOP,R0              ;SET LOOPBACK
4736 056552 004737 061562'                JSR    PC,T19SNPR              ;SETUP T19PK2 FOR WRITE NPR
4737 056556 012704 062220'                MOV    T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
4738 056562 010465 000000                  MOV    R4,TSD8(R5)            ;SET THE PACKET ADDRESS TO EXECUTE
4739 056566 004737 016146'                JSR    PC,CHKTSSR              ;WAIT FOR SSR TO SET
4740 056572                                FORCERROR 102$               ;GOODFORCE ERROR IF FORCER=1
4741 056606 103407                        BCS    105$                   ;BR IF CARRY SET (GOOD RETURN)
4742 056610 010001                        MOV    R0,R1                    ;SAVE CONTENTS OF TSSR
4743 056612                                NEXT,ERRNO
4744 056612 104455                        102$: ERRDF  ERRNO,T194SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                         TRAP    C$ERDF
                                         .WORD    844
                                         .WORD    T194SSR
056612 104455
056614 001514
056616 060131'

```

Address	Offset	Hex	Label	Instruction	Comment	Trap	Word	PKTSSR
4745	056622	005237	002222'	INC FATFLG	SET FATAL ERROR FLAG			PKTSSR
4746	056626			CKLOOP	LOOP ON ERROR, IF FLAG SET			
4747	056626	104406				TRAP		C\$CLP1
4748	056630	012700	000001	Do a WRITE FORMAT to set IRESV4-->IRSTR = 1				
4749	056634	004737	061702'	MOV #WF, I4RES, R0	IRESV4-->IRSTR=1			
4750	056640	012704	062220'	JSR PC, T19WFM	SETUP T19PK2 FOR WRITE FORMAT			
4751	056644	010465	000000	MOV #T19PK2, R4	GET WRITE SUBSYSTEM COMMAND PACKET			
4752	056650	004737	016146'	MOV R4, TSD8(R5)	SET THE PACKET ADDRESS TO EXECUTE			
4753	056654			JSR PC, CHKTSR	WAIT FOR SSR TO SET			
4754	056670	103407		FORCERROR 1124	FORCE ERROR IF FORCER=1			
4755	056672	010001		BCS 1204	BR IF CARRY SET (GOOD RETURN)			
4756	056674			MOV R0, R1	SAVE CONTENTS OF TSSR			
4757	056674			NEXT, ERRNO				
	056674	104455		ERRDF ERRNO, T198SSR, PKTSSR	DEVICE FATAL SSR FAILED TO SET	TRAP		C\$ERDF
	056676	001515				WORD	845	
	056700	060352'				WORD	T198SSR	
	056702	011656'				WORD	PKTSSR	
4758	056704	005237	002222'	INC FATFLG	SET FATAL ERROR FLAG			
4759	056710			CKLOOP	LOOP ON ERROR, IF FLAG SET			
	056710	104406				TRAP		C\$CLP1
4760				Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT				
4761	056712	012700	000001	MOV #1, R0	WRITE 1 BYTE			
4762	056716	012701	002312'	MOV #DATA, R1	FIFO WRITE DATA ADDRESS			
4763	056722	004737	061626'	JSR PC, T19WFI	SETUP T19PK2 FOR WRITE FIFO			
4764	056726	012704	062220'	MOV #T19PK2, R4	GET WRITE SUBSYSTEM COMMAND PACKET			
4765	056732	010465	000000	MOV R4, TSD8(R5)	SET THE PACKET ADDRESS TO EXECUTE			
4766	056736	004737	016146'	JSR PC, CHKTSR	WAIT FOR SSR TO SET			
4767	056742			FORCERROR 1324	FORCE ERROR IF FORCER=1			
4768	056756	103407		BCS 1404	BR IF CARRY SET (GOOD RETURN)			
4769	056760	010001		MOV R0, R1	SAVE CONTENTS OF TSSR			
4770	056762			NEXT, ERRNO				
4771	056762			1324: ERRDF ERRNO, T195SSR, PKTSSR	DEVICE FATAL SSR FAILED TO SET	TRAP		C\$ERDF
	056762	104455				WORD	846	
	056764	001516				WORD	T195SSR	
	056766	060174'				WORD	PKTSSR	
	056770	011656'						
4772	056772	005237	002222'	INC FATFLG	SET FATAL ERROR FLAG			
4773	056776			CKLOOP	LOOP ON ERROR, IF FLAG SET			
	056776	104406				TRAP		C\$CLP1
4774				Do a READ FIFO with tape direction OUT to load tape out write latch				
4775	057000	012700	000001	MOV #1, R0	SET READ BYTE COUNT			
4776	057004	004737	061606'	JSR PC, T19RFI	SETUP T19PK2 FOR READ FIFO			
4777	057010	012704	062220'	MOV #T19PK2, R4	GET WRITE SUBSYSTEM COMMAND PACKET			
4778	057014	010465	000000	MOV R4, TSD8(R5)	SET THE PACKET ADDRESS TO EXECUTE			
4779	057020	004737	016146'	JSR PC, CHKTSR	WAIT FOR SSR TO SET			
4780	057024			FORCERROR 1524	FORCE ERROR IF FORCER=1			
4781	057040	103407		BCS 1604	BR IF CARRY SET (GOOD RETURN)			
4782	057042	010001		MOV R0, R1	SAVE CONTENTS OF TSSR			
4783	057044			NEXT, ERRNO				
4784	057044			1524: ERRDF ERRNO, T196SSR, PKTSSR	DEVICE FATAL SSR FAILED TO SET	TRAP		C\$ERDF
	057044	104455				WORD	847	
	057046	001517				WORD	T196SSR	
	057050	060240'				WORD	PKTSSR	
	057052	011656'						
4785	057054							

H15

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

SEQ 189

```

4786 057060 160$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
057060 104406 TRAP C$CLP1
4787 ; Do a WRITE NPR to set loopback and tape direction IN
4788 057062 005000 CLR R0 ;CLR NP.OUT TO SET TAPE DIRECTION IN
4789 057064 052700 000040 BIS #NP.LOOP,R0 ;SET LOOPBACK
4790 057070 004737 061562' JSR PC,T19SNPR ;SETUP T19PK2 FOR WRITE NPR
4791 057074 012704 062220' MOV #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
4792 057100 010465 000000 MOV R4,TSD8(R5) ;SET THE PACKET ADDRESS TO EXECUTE
4793 057104 004737 016146' JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
4794 057110 FORCERROR 182$ ;GOODFORCE ERROR IF FORCER=1
4795 057124 103407 BCS 190$ ;BR IF CARRY SET (GOOD RETURN)
4796 057126 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
4797 057130 NEXT,ERRNO
4798 057130 182$: ERRDF ERRNO,T194SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
057130 104455 TRAP C$ERDF
057132 001520 .WORD 848
057134 060131' .WORD T194SSR
057136 011656' .WORD PKTSSR
4799 057140 005237 002222' INC FATFLG ;SET FATAL ERROR FLAG
4800 057144 190$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
057144 104406 TRAP C$CLP1
4801 ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 0
4802 057146 005000 CLR R0 ;SET IRESV4==>IRSTR=0
4803 057150 004737 061702' JSR PC,T19WFMT ;SETUP T19PK2 FOR WRITE FORMAT
4804 057154 012704 062220' MOV #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
4805 057160 010465 000000 MOV R4,TSD8(R5) ;SET THE PACKET ADDRESS TO EXECUTE
4806 057164 004737 016146' JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
4807 057170 FORCERROR 202$ ;GOODFORCE ERROR IF FORCER=1
4808 057204 103407 BCS 210$ ;BR IF CARRY SET (GOOD RETURN)
4809 057206 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
4810 057210 NEXT,ERRNO
4811 057210 202$: ERRDF ERRNO,T198SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
057210 104455 TRAP C$ERDF
057212 001521 .WORD 849
057214 060352' .WORD T198SSR
057216 011656' .WORD PKTSSR
4812 057220 005237 002222' INC FATFLG ;SET FATAL ERROR FLAG
4813 057224 210$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
057224 104406 TRAP C$CLP1
4814 ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
4815 057226 012700 000001 MOV #WF.I4RES,R0 ;IRESV4==>IRSTR=1
4816 057232 004737 061702' JSR PC,T19WFMT ;SETUP T19PK2 FOR WRITE FORMAT
4817 057236 012704 062220' MOV #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
4818 057242 010465 000000 MOV R4,TSD8(R5) ;SET THE PACKET ADDRESS TO EXECUTE
4819 057246 004737 016146' JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
4820 057252 FORCERROR 222$ ;GOODFORCE ERROR IF FORCER=1
4821 057266 103407 BCS 230$ ;BR IF CARRY SET (GOOD RETURN)
4822 057270 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
4823 057272 NEXT,ERRNO
4824 057272 222$: ERRDF ERRNO,T198SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
057272 104455 TRAP C$ERDF
057274 001522 .WORD 850
057276 060352' .WORD T198SSR
057300 011656' .WORD PKTSSR
4825 057302 005237 002222' INC FATFLG ;SET FATAL ERROR FLAG
4826 057306 230$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
057306 104406 TRAP C$CLP1

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

SEQ 190

```

4827      : Do a READ FIFO with tape direction IN to read data
4828 057310 012700 000001      MOV    #1,R0      ;SET READ BYTE COUNT
4829 057314 004737 061606'     JSR     PC,T19RFIF    ;SETUP T19PK2 FOR READ FIFO
4830 057320 012704 062220'     MOV     #T19PK2,R4    ;GET WRITE SUBSYSTEM COMMAND PACKET
4831 057324 010465 000000      MOV     R4,TSD8(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
4832 057330 004737 016146'     JSR     PC,CHKTSSR    ;WAIT FOR SSR TO SET
4833 057334      FORCERROR      282$      ;###FORCE ERROR IF FORCER=1
4834 057350 103407      BCS      290$      ;BR IF CARRY SET (GOOD RETURN)
4835 057352 010001      MOV     R0,R1      ;SAVE CONTENTS OF TSSR
4836 057354      NEXT,ERRNO
4837 057354      282$: ERRDF  ERRNO,T196SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     851
                                .WORD     T196SSR
                                .WORD     PKTSSR
                                057354 104455
                                057356 001523
                                057360 060240'
                                057362 011656'
4838 057364 005237 002222'     INC     FATFLG      ;SET FATAL ERROR FLAG
4839 057370      290$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                057370 104406
4840      : If Data read from FIFO NOT= to Data sent Then Print Error
4841 057372 004737 061760'     JSR     PC,T19SETEXP    ;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
4842 057376 012701 057622'     MOV     #T19EXPSTA,R1    ;GET EXPECTED READ STATUS
4843 057402 012702 062112'     MOV     #T198STA,R2     ;GET RCV READ STATUS
4844 057406 013711 002312'     MOV     DATA,(R1)      ;SET EXPD WORD #8 = DATA
4845 057412 016261 000002 000002      MOV     2(R2),2(R1)    ;SET EXPD WORD #9 = RCV (NOT TESTING)
4846 057420 005000      CLR     R0      ;HIGH RCV ADDRESS FOR CKMSG2
4847 057422 012701 062072'     MOV     #T198FR,R1     ;LOW RCV ADDRESS FOR CKMSG2
4848 057426 012702 057602'     MOV     #T19EXP,R2     ;EXPD ADDRESS
4849 057432 012703 000022      MOV     #18,R3      ;NUMBER OF BYTES TO COMPARE
4850 057436 004737 011310'     JSR     PC,CKMSG2      ;EXPD EQUAL RCV?
4851 057442      FORCERROR      302$,NOTSSR    ;###
4852 057452 103404      BCS      310$      ;BR IF YES
4853 057454      NEXT,ERRNO
4854 057454      302$: ERHRD  ERRNO,T19RSTR,MSGSUB    ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD     852
                                .WORD     T19RSTR
                                .WORD     MSGSUB
                                057454 104456
                                057456 001524
                                057460 061370'
                                057462 013552'
4855 057464      310$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                057464 104406
4856
4857
4858 057466      FORCEEXIT      355$      ;###
4859 057476 013703 002316'     MOV     TSTPTR,R3      ;RESTORE CURRENT TSTBLK POINTER
4860 057502 020327 003062'     CMP     R3,#TBLEND    ;END OF TSTBLK?
4861 057506 103002      BHS      355$      ;BR IF YES
4862 057510 000137 056524'     JMP     100$      ;DO ANOTHER TSTBLK PATTERN
4863 057514      355$:
4864
4865 057514      ENDSUB      ;////////// END SUBTEST ///////////
                                L10072:
                                TRAP      C$ESUB
                                057514 104403
4866
4867 057516 005737 002222'     TST     FATFLG      ;ANY FATAL ERRORS ?
4868 057522 001402      BEQ      360$      ;BRANCH IF NOT
4869 057524 004737 017014'     JSR     PC,CKDROP    ;TRY TO DROP THE UNIT
4870 057530      360$:
4871

```

J15

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

SEQ 191

```

4872 057530          EXIT    TST          ;////////// EXIT TEST ////////////
      057530 104432          TRAP    CEXIT
      057532 002602          .WORD   L10066-.
4873
4874
4875          ;+
4876          ;LOCAL STORAGE FOR THIS TEST
4877          ;-
4878 057534 000000      T19PREV:          .WORD   0          ;DRIVE SIGNAL 1-0 TRANSITION FLAG
4879          ;+
4880          ; LOOPBACK DRIVE SIGNAL TABLE
4881          ; THIS TABLE IS USED BY T19CNVT TO SETUP
4882          ; A DRIVE PATTERN FROM THE TEST DATA INPUT PATTERN.
4883          ;
4884          ; WRITE CONTROL SIGNALS ARE OF FORM WC.XXX
4885          ; WRITE FORMAT SIGNALS ARE OF FORM WF.XXXX
4886          ;-
4887 057536          T19FCTL:          ;WRITE CONTROL DRIVE SIGNALS
4888 057536 000001      WC.IG0          ;IG0==>IFPT      DATA<0>
4889 057540 000002      WC.TFEN        ;TFEN==>IFBY      DATA<1>
4890 057542 000004      WC.IRWU        ;IRWU==>IRWD      DATA<2>
4891 057544 000010      WC.IREW        ;IREW==>IDBY      DATA<3>
4892 057546 00200C      WF.IERASE*256. ;IFAD==>ILDPA    DATA<4>
4893 057550 000040      WC.I1TAD       ;ITAD1==>IONL     DATA<5>
4894 057552 000100      WC.IOTAD       ;ITADO==>IRUY     DATA<6>
4895 057554 000200      WC.IFAD        ;IERASE==>ISPEED DATA<7>
4896 057556 004000      WF.IEDIT*256. ;IEDIT==>THER    DATA<8>
4897 057560 010000      WF.IWFM*256.  ;IWFM==>IFMK     DATA<9>
4898 057562 020000      WF.IREV*256.  ;IREV==>ICER     DATA<10>
4899 057564 040000      WF.IWRT*256.  ;IWR1==>IIDENT   DATA<11>
4900 057566 100000      WF.IHISP*256. ;IHISP==>IEOT    DATA<12>
4901 057570 000000      .WORD   0      ;IRESV2 (UNUSED)DATA<13>
4902 057572 000000      .WORD   0      ;IRESV1 (UNUSED)DATA<14>
4903 057574 000000      .WORD   0      ;PARERR (UNTESTED)DATA<15>
4904
4905 057576          T19MSK:          ;MASK OF UNTESTED BITS IN READ STATUS BYTES
4906          ;UNTESTED BITS ARE SET TO 1
4907 057576 377          .BYTE   *C<000> ;BYTE 0 MASK
4908 057577 037          .BYTE   *C<340> ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
4909 057600 360          .BYTE   *C<017> ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
4910 057601 000          .BYTE   0      ;MAKE IT EVEN
4911
4912 057602          T19EXP:          ;BEGIN EXPECTED DATA BUFFER
4913 057602 000000      .WORD   0      ;MESSAGE TYPE
4914 057604 000000      .WORD   0      ;DATA FIELD LENGTH
4915 057606 000000      .WORD   0      ;R6PCR
4916 057610 000000      .WORD   0      ;XST0
4917 057612 000000      .WORD   0      ;XST1
4918 057614 000000      .WORD   0      ;XST2
4919 057616 000000      .WORD   0      ;XST3
4920 057620 000000      .WORD   0      ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
4921 057622          T19EXSTA: .BLKB 64. ;EXPECTED READ STATUS AND WRITE FIFO DATA
4922 057722          T19EXEND:          ;END EXPECTED DATA BUFFER
4923          ;+
4924          ;LOCAL TEXT MESSAGES FOR TEST
4925          ;-
4926

```

```

4927 057722      124      162      141 TST19ID:      .ASCIZ 'Transport Bus Interface Loopback'
4928 057763      127      122      111 T19SSR: .ASCIZ 'WRITE CHARACTERISTICS Failed'
4929 060020      127      122      111 T192SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Misc) Failed'
4930 060064      127      122      111 T193SSR: .ASCIZ 'WRITE SUBSYSTEM (Read Status) Failed'
4931 060131      127      122      111 T194SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Npr) Failed'
4932 060174      127      122      111 T195SSR: .ASCIZ 'WRITE SUBSYSTEM (Write FIFO) Failed'
4933 060240      127      122      111 T196SSR: .ASCIZ 'WRITE SUBSYSTEM (Read FIFO) Failed'
4934 060303      127      122      111 T197SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Control) Failed'
4935 060352      127      122      111 T198SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Format) Failed'
4936 060420      106      111      106 T191CMP: .ASCIZ 'FIFO Status in WORD #9 Incorrect after Initialize'
4937 060502      122      145      141 T192CMP: .ASCIZ 'Read FIFO Data not equal to Write FIFO . Data is in WORD #8'
4938 060576      124      141      160 T193CMP: .ASCIZ 'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
4939 060664      122      145      141 T195CMP: .ASCIZ 'Read FIFO Data not equal to Write FIFO Data'
4940 060740      106      111      106 T196CMP: .ASCIZ 'FIFO Status (in WORD #9) Incorrect after READ FIFO'
4941 061023      124      141      160 T197CMP: .ASCIZ 'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
4942 061111      103      157      156 T198CMP: .ASCIZ 'Control Signal Loopback Data Error. Data is in WORD #8'
4943 061200      122      145      141 T199CMP: .ASCIZ 'Read/Write Loopback Data Error. Data is in WORD #8'
4944 061263      114      157      157 T19WSTR: .ASCIZ 'Loopback Data Error when strobed by Write strobe. Data is in WORD #8'
4945 061370      114      157      157 T19RSTR: .ASCIZ 'Loopback Data Error when strobed by Read Strobe. Data is in WORD #8'
4946
4947                      .EVEN
4948
4949
4950
4951
4952 061474
4953 061474
4954 061500      012701      062072'
4955 061504      012702      000120
4956 061510      105021
4957 061512      005302
4958 061514      003375
4959 061516      000207
4960
4961
4962
4963
4964 061520
4965 061520      004737      061474'
4966 061524      012700      062230'
4967 061530      112720      000005
4968 061534      105010
4969 061536      000207
4970
4971
4972
4973
4974 061540
4975 061540      004737      061474'
4976 061544      012700      062230'
4977 061550      112720      000010
4978 061554      112710      000030
4979 061560      000207
4980
4981
4982
4983

;+
; CLEAR MESSAGE BUFFER
;-
T19CLRBUF:
        SAVREG
        MOV     #T19BFR,R1
        MOV     #T19BEND-T19BFR,R2
10$:    CLR     (R1)+
        DEC     R2
        BGT     10$
        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
        MOV     #PW,RDSTATUS,(R0)+
        CLR     (R0)
        RTS     PC

;CLEAR MESSAGE BUFFER
;WRITE SUBSYSTEM DATA BUFFER
;STORE READ STATUS COMMAND IN BSEL0
;CLEAR BSEL1
;RETURN

;+
; SETUP T19PK2 PACKET FOR WRITE MISC Reset Tape Status F-FLOPS
;-
T19RSFIF:
        JSR     PC,T19CLRBUF
        MOV     #T19DT2,R0
        MOV     #PW,WMISC,(R0)+
        MOV     #MS,RSFIF!MS,RSTAP,(R0)
        RTS     PC

;CLEAR MESSAGE BUFFER
;WRITE SUBSYSTEM DATA BUFFER
;STORE WRITE MISCELLANEOUS IN BSEL0
;STORE BSEL1 CLEAR FIFO CODES
;RETURN

;+
; SETUP T19PK2 PACKET FOR WRITE NPR
;-

```

```

4984      ; INPUT:
4985      ;      R0 CONTAINS BSEL1 NPR DATA
4986      ;
4987      ;      SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
4988      ;
4989      T19SNPR:
4990      JSR      PC,T19CLRBUF      ; CLEAR MESSAGE BUFFER
4991      MOV      @T19DT2,R1        ; WRITE SUBSYSTEM DATA BUFFER
4992      MOVB     @PW.WNPR,(R1)+    ; STORE WRITE NPR IN BSEL0
4993      BIS      @NP.WRP,R0        ; DON'T WRITE WRONG PARITY
4994      MOVB     R0,(R1)           ; STORE NPR DATA IN BSEL1
4995      RTS      PC                ; RETURN
4996
4997      ;
4998      ; SETUP T19PK2 PACKET FOR READ FIFO
4999      ;
5000      ; INPUT:
5001      ;      R0 CONTAINS SEL2 BYTE COUNT
5002      ;
5003      T19RFIF:
5004      JSR      PC,T19CLRBUF      ; CLEAR MESSAGE BUFFER
5005      MOV      @T19DT2,R1        ; WRITE SUBSYSTEM DATA BUFFER
5006      MOVB     @PW.RFIFO,(R1)+  ; STORE READ FIFO IN BSEL0
5007      MOVB     R0,(R1)+         ; STORE BYTE COUNT IN BSEL1
5008      RTS      PC                ; RETURN
5009
5010      ;
5011      ; SETUP T19PK2 PACKET FOR WRITE FIFO
5012      ;
5013      ; INPUT:
5014      ;      R0 CONTAINS BYTE COUNT
5015      ;      R1 CONTAINS DATA PATTERN BLOCK ADDRESS
5016      ;
5017      T19WFIF:
5018      SAVREG
5019      JSR      PC,T19CLRBUF      ; SAVE R1-R5 UNTIL NEXT RETURN
5020      MOV      @T19DT2,R2        ; CLEAR MESSAGE BUFFER
5021      MOVB     @PW.WFIFO,(R2)+  ; WRITE SUBSYSTEM DATA BUFFER
5022      MOVB     R0,(R2)+         ; STORE WRITE FIFO IN BSEL0
5023      CLR      (R2)+            ; STORE BYTE COUNT IN BSEL1
5024      MOVB     (R1)+,(R2)+      ; CLEAR SEL2 (UNUSED)
5025      DEC      R0               ; STORE DATA PATTERN BYTE
5026      BGT     10$             ; DONE ALL BYTES?
5027      BGT     10$             ; BR IF NO
5028      RTS      PC                ; RETURN
5029
5030      ;
5031      ; SETUP T19PK2 FOR WRITE CONTROL
5032      ;
5033      ; INPUT:
5034      ;      R0 CONTAINS DRIVING DATA PATTERN
5035      ;
5036      T19WCTL:
5037      JSR      PC,T19CLRBUF      ; CLEAR MESSAGE BUFFER
5038      MOV      @T19DT2,R1        ; WRITE SUBSYSTEM DATA BUFFER
5039      MOVB     @PW.WCTL,(R1)+    ; STORE WRITE CONTROL IN BSEL0
5040      MOVB     R0,(R1)+         ; STORE DATA WORD IN BSEL1
5041      RTS      PC                ; RETURN
5042
5043      ;
5044      ; SETUP T19PK2 FOR WRITE FORMAT TRANSPORT REGISTER

```

```

5041
5042
5043
5044
5045 061702
5046 061702 004737 061474'
5047 061706 012701 062230'
5048 061712 112721 000007
5049 061716 110021
5050 061720 000207
5051
5052
5053
5054 061722
5055 061722 012700 062230'
5056 061726 112720 000010
5057 061732 112710 000200
5058 061736 000207
5059
5060
5061
5062 061740
5063 061740 012701 057602'
5064 061744 012700 000120
5065 061750 105021
5066 061752 005300
5067 061754 003375
5068 061756 000207
5069
5070
5071
5072
5073 061760
5074 061760 012702 057602'
5075 061764 012703 062072'
5076 061770 012700 000010
5077 061774 012322
5078 061776 005300
5079 062000 003375
5080 062002 000207
5081
5082
5083
5084
5085
5086
5087
5088
5089
5090
5091
5092
5093
5094
5095
5096
5097

; INPUT:
; RO CONTAINS DRIVING DATA PATTERN
;
T19WFMF:
    JSR    PC,T19CLRBUF      ;CLEAR MESSAGE BUFFER
    MOV     T19DT2,R1        ;WRITE SUBSYSTEM DATA BUFFER
    MOV     T19WFMF,(R1)+    ;STORE WRITE FORMAT IN BSEL0
    MOV     R0,(R1)+        ;STORE DATA WORD IN BSEL1
    RTS     PC               ;RETURN

;
; *
; SETUP T19PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
;
T19SEXT:
    MOV     T19DT2,R0        ;WRITE SUBSYSTEM DATA BUFFER
    MOV     T19WMISC,(R0)+   ;STORE WRITE MISCELLANEOUS IN BSEL0
    MOV     T19MEXT,(R0)+   ;STORE INVERT EXTENDED FEATURES IN BSEL1
    RTS     PC               ;RETURN

;
; *
; CLEAR EXPECTED DATA MESSAGE BUFFER
;
T19CLEXP:
    MOV     T19EXP,R1        ;GET EXPD ADDRESS
    MOV     T19EXEND-T19EXP,R0 ;GET EXPD SIZE
10$:   CLR    (R1)+          ;CLEAR A BYTE
    DEC     R0               ;DONE?
    BGT     10$              ;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     T19EXP,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:
;
; RO TEST PATTERN
;
; IMPLICIT INPUTS:
;
; T19BCTL - CONTAINS WRITE CONTROL / WRITE FORMAT CONVERSION BITS
;
; OUTPUTS:
;
; RO - LOW BYTE CONTAINS WRITE CONTROL DATA
; - HIGH BYTE CONTAINS WRITE FORMAT DATA
;

```

N15

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

SEQ 195

5098 062004
5099 062004
5100 062010 012701 057536'
5101 062014 005002
5102 062016 012703 000020
5103 062022 006000
5104 062024 103001
5105 062026 051102
5106 062030 005721
5107 062032 005393
5108 062034 003372
5109 062036 010200
5110 062040 000207
5111
5112
5113
5115 062042
5117
5118
5119
5120 062050
5121 062050 100004
5122 062052 062060'
5123 062054 000000
5124 062056 000012
5125
5126 062060
5127 062060 062072'
5128 062062 000000
5129 062064 000024
5130 062066 000000
5131 062070 000007
5132
5133
5134
5135
5136 0 072
5137 0 072 000000
5138 0 074 000000
5139 062076 000000
5140 062100 000000
5141 062102 000000
5142 062104 000000
5143 062106 000000
5144 062110 000000
5145 062112
5146 062212
5147
5148
5149
5151 062212
5153 062220
5154 062220 100006
5155 062222 062230'
5156 062224 000000
5157 062226 000012
5158

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
20\$: TST (R1)+ ;SET CONVERTED BIT
DEC R3 ;POINT TO NEXT BIT IN CONVERSION TABLE
BGT 10\$;DONE?
MOV R2,R0 ;BR IF NO
RTS PC ;COPY RESULT
;RETURN

.BLKB 10-<,-TSV2&7>
;WRITE CHARACTERISTICS COMMAND PACKET
T19PACKET:
;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND, WITH ACK
;ADDRESS OF CHARACTERISTICS BLOCK
;MINIMUM MESSAGE PACKET SIZE
T19DATA:
;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER
;LENGTH OF MESSAGE BUFFER
;ESS,ENB,EAI,ERI
;EXTENDED FEATURES UNIT NO.
;MESSAGE BUFFER FOR ALL TEST 8 COMMANDS
T19BFR:
;BEGIN MESSAGE BUFFER
;MESSAGE TYPE
;DATA FIELD LENGTH
;RBCR
;XST0
;XST1
;XST2
;XST3
;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
;READ STATUS AND WRITE FIFO BUFFER
;END OF MESSAGE BUFFER
T19BFSTA: .BLKB 64.
T19BEND:
;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
;T19PK2:
;BLKB 10-<,-TSV2&7>
;WRITE SUBSYSTEM WITH ACK
;LOW ADDRESS OF DATA BLOCK
;HIGH ADDRESS OF DATA BLOCK
;MINIMUM MESSAGE PACKET SIZE

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

SEQ 196

```

5159 062230          T19DT2:          ;DATA BLOCK
5160 062230          000              ;BSEL0
5161 062231          000              ;BSEL1
5162 062232          000000          ;SEL2
5163 062234          ;BLKB          64. ;WRITE FIFO DATA OUTPUT BUFFER
5164
5165
5166 062334          ENDTST
      062334
      062334          104401          L10066: TRAP C$ETST
5167
5168          ;SBTTL TEST 9: READ/WRITE DATA PARITY TEST
5169
5170          ;**
5171          ; TEST DESCRIPTION:
5172          ;
5173          ; This test verifies that the Write Data Parity generator
5174          ; and the Read Data Parity checker operate properly. The
5175          ; Transport Bus signal loopback mode is enabled and a
5176          ; Set Wrong parity function is executed. Then various
5177          ; Write Subsystem Memory functions are performed to
5178          ; write data to and from the FIFO in loopback mode.
5179          ; The program then checks to insure a Read Data parity
5180          ; error occurred.
5181          ; A Reset FIFO is done and the Read Data parity
5182          ; error bit is again tested to insure it cleared.
5183          ; Finally a Clear wrong parity function is done
5184          ; and it is verified the data word can pass in loopback
5185          ; mode without setting Read Data parity error.
5186
5187          ; TEST STEPS:
5188          ;
5189          ; REPEAT FOR LOOPCNT
5190          ; BEGIN
5191          ;   Write to TSSR register to soft initialize the controller
5192          ;   Do WRITE CHARACTERISTICS to check for Extended Features Switch
5193          ;   IF Extended Features Hardware Switch Clear then:
5194          ;     Do Write Subsystem Write Miscellaneous to Set Extended Features.
5195          ;   Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
5196          ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
5197          ; BEGIN
5198          ;   (* Verify Write Wrong Parity Sets Parity Error *)
5199          ;   Do a WRITE NPR to set loopback and tape direction OUT
5200          ;     and SET Write Wrong Parity.
5201          ;   Do a WRITE FORMAT to set IRESV4-->IRSTR = 1 (sets read strobe high)
5202          ;   Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
5203          ;   Do a READ FIFO with tape direction OUT to load tape out write latch
5204          ;     (this is when wrong parity (IWP) is set)
5205          ;   Do a WRITE FORMAT to set IRESV4-->IRSTR = 0 (sets read strobe low)
5206          ;     (Read Strobe sets PAR IN H [Parity Error])
5207          ;   Do a WRITE FORMAT to set IRESV4-->IRSTR = 1 (sets read strobe high)
5208          ;   Do a Write Subsystem READ STATUS
5209          ;   If Read Data parity error NOT=1 Then Print Error
5210          ;   Do a Write Misc to RESET FIFO
5211          ;   Do a Write Subsystem READ STATUS
5212          ;   If Read Data parity error NOT=0 Then Print Error
5213          ;
5214          ;   (* Verify Data can be transferred without a Parity Error *)

```

```

5214      | Do a WRITE FORMAT to set IRESV4-->IRSTR = 1 (sets read strobe high)
5215      | Do a WRITE NPR to set loopback and tape direction OUT
5216      | and CLEAR Write Wrong Parity.
5217      | Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
5218      | Do a READ FIFO with tape direction OUT to load tape out write latch
5219      | Do a WRITE FORMAT to set IRESV4-->IRSTR = 0 (sets read strobe low)
5220      | (Read Strobe should NOT set PAR IN H [Parity Error] here)
5221      | Do a WRITE FORMAT to set IRESV4-->IRSTR = 1 (sets read strobe high)
5222      | Do a Write Subsystem READ STATUS
5223      | If Read Data parity error NOT=0 Then Print Error
5224      |
5225      | END
5226      |
5227      |
5228      |
5229      | BGNTST
5230      |
5231      |
5232      |
5233      |
5234      | T9:;
5235      | ASCII MESSAGE TO IDENTIFY TEST
5236      | DO INITIAL TEST SETUP
5237      | PERFORM 10 ITERATIONS
5238      |
5239      | BGNSUB
5240      |
5241      |
5242      |
5243      |
5244      |
5245      |
5246      |
5247      |
5248      |
5249      |
5250      |
5251      |
5252      |
5253      |
5254      |
5255      |
5256      |
5257      |
5258      |
5259      |
5260      |
5261      |
5262      |

```

062336 012700 064722' MOV #TST20ID,R0 ASCII MESSAGE TO IDENTIFY TEST
 062336 004737 016322' JSR PC,TSTSETUP DO INITIAL TEST SETUP
 062346 012737 000012 002216' MOV #10,,LOOPCNT PERFORM 10 ITERATIONS
 062354 T20LOOP: BGNSUB BEGIN SUBTEST
 062354 104402 T9.1: TRAP C#BSUB
 062354 104402 Write to TSSR register to soft initialize the controller
 062356 004737 015604' JSR PC,SOFINIT WRITE TO TSSR TO SOFT INITIALIZE
 062362 103405 BCS 10# BR IF SOFT INIT OKAY
 062364 010001 MOV R0,R1 SAVE CONTENTS OF TSSR
 062366 104455 ERDIF ERRNO,SFERR,SFMSG DEVICE FATAL DURING INIT
 062370 001604 TRAP C#ERDIF
 062372 003646' .WORD 900
 062374 011644' .WORD SFERR
 062374 011644' .WORD SFMSG
 062376 005037 002222' 10#: Do WRITE CHARACTERISTICS to check for Extended Features Switch
 062402 012704 066130' CLR FATFLAG CLEAR FATAL ERROR FLAG
 062406 004737 010472' MOV #T20PACKET,R4 GET THE ADDRESS OF COMMAND PACKET
 062412 JSR PC,WRCHR DO WRITE CHARACTERISTICS COMMAND
 062426 103407 F0HCERROR 12# BR IF CARRY SET (GOOD RETURN)
 062430 010001 BCS 15# BR IF CARRY SET (GOOD RETURN)
 062432 NEXT,ERRNO SAVE CONTENTS OF TSSR
 062432 12#: ERDIF ERRNO,T20SSR,PKTSSR DEVICE FATAL SSR FAILED TO SET
 062432 104455 TRAP C#ERDIF
 062434 001605 .WORD 901
 062436 064751' .WORD T20SSR
 062440 011656' .WORD PKTSSR
 062442 005237 002222' 15#: INC FATFLAG SET FATAL ERROR FLAG
 062446 104406 CKLOOP LOOP ON ERROR, IF FLAG SET
 062446 104406 TRAP C#CLP1
 062450 012701 066152' If Extended Features Hardware Switch Clear then:
 062454 032761 000200 000012 Do Write Subsystem Write Miscellaneous to Set Extended Features.
 062462 001026 MOV #T20BFR,R1 MESSAGE BUFFER ADDRESS
 062464 004737 066036' BIT #X2,EXTF,XST2(R1) EXTENDED FEATURES SWITCH SET?
 BNE 30# BR IF YES
 JSR PC,T20SEXT SETUP PACKET FOR WRITE MISC INVERT

Line	Address	Offset	Hex	Label	Instruction	Comment
5263	062470	012704	061300'		MOV #T20PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
5264	062474	010465	000000		MOV R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE
5265	062500	004737	016146'		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET
5266	062504				FORCERROR 22\$	;SSDFORCE ERROR IF FORCER=1
5267	062520	103407			BCS 30\$	;BR IF CARRY SET (GOOD RETURN)
5268	062522	010001			MOV R0,R1	;SAVE CONTENTS OF TSSR
5269	062524				NEXT,ERRNO	
5270	062524	22\$:			ERRDF ERRNO,T202SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	062524	104455				TRAP C\$ERDF
	062526	001606				.WORD 902
	062530	065006'				.WORD T202SSR
	062532	011656'				.WORD PKTSSR
5271	062534	005237	002222'		INC FATFLG	;SET FATAL ERROR FLAG
5272	062540	30\$:			CKLOOP	;LOOP ON ERROR, IF FLAG SET
	062540	104406				TRAP C\$CLP1
5273					; Do WRITE CHARACTERISTICS to select reserved unit 7	
5274	062542	012704	066130'		MOV #T20PACKET,R4	;GET THE ADDRESS OF COMMAND PACKET
5275	062546	004737	010472'		JSR PC,WRITCHR	;DO WRITE CHARACTERISTICS COMMAND
5276	062552				FORCERROR 42\$	;SSDFORCE ERROR IF FORCER=1
5277	062566	103407			BCS 50\$	;BR IF CARRY SET (GOOD RETURN)
5278	062570	010001			MOV P0,R1	;SAVE CONTENTS OF TSSR
5279	062572				NEXT,ERRNO	
5280	062572	42\$:			ERRDF ERRNO,T20SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	062572	104455				TRAP C\$ERDF
	062574	001607				.WORD 903
	062576	064751'				.WORD T20SSR
	062600	011656'				.WORD PKTSSR
5281	062602	005237	002222'		INC FATFLG	;SET FATAL ERROR FLAG
5282	062606	50\$:			CKLOOP	;LOOP ON ERROR, IF FLAG SET
	062606	104406				TRAP C\$CLP1
5283						
5284						
5285					; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE	
5286	062610	012703	002752'		MOV #TSTBLK,R3	;GET FIRST PATTERN ADDRESS
5287	062614	012337	002312'		MOV (R3),DATA	;GET A TEST PATTERN
5288	062620	042737	177400	002312'	BIC #C<377>,DATA	;DATA IS BYTE
5289	062626	010337	002316'		MOV R3,TSTPTR	;SETUP CURRENT TSTBLK POINTER
5290					; Do a WRITE NPR to set loopback and tape direction OUT and	
5291					and SET Write Wrong Parity.	
5292	062632	012700	000100		MOV #NP.OUT,R0	;SET TAPE DIRECTION OUT
5293	062636	052700	000040		BIS #NP.LOOP,R0	;SET LOOPBACK
5294	062642	042700	000020		BIC #NP.WRP,R0	;SET WRITE WRONG PARITY (INVERTED)
5295	062646	004737	065706'		JSR PC,T20WNPR	;SETUP T20PK2 FOR WRITE NPR
5296	062652	012704	066300'		MOV #T20PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
5297	062656	010465	000000		MOV R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE
5298	062662	004737	016146'		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET
5299	062666				FORCERROR 102\$	;SSDFORCE ERROR IF FORCER=1
5300	062702	103407			BCS 105\$	;BR IF CARRY SET (GOOD RETURN)
5301	062704	010001			MOV R0,R1	;SAVE CONTENTS OF TSSR
5302	062706				NEXT,ERRNO	
5303	062706	102\$:			ERRDF ERRNO,T204SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	062706	104455				TRAP C\$ERDF
	062710	001610				.WORD 904
	062712	065117'				.WORD T204SSR
	062714	011656'				.WORD PKTSSR
5304	062716	005237	002222'		INC FATFLG	;SET FATAL ERROR FLAG
5305	062722	105\$:			CKLOOP	;LOOP ON ERROR, IF FLAG SET

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 9: READ/WRITE DATA PARITY TEST

SEQ 199

5306	062722	104406					TRAP C\$CLP1
5307	062724	012700	000001	:	Do a WRITE FORMAT to set IRESV4=1 (sets read strobe high)		
5308	062730	004737	066002'		MOV @WF,I4RES,R0	IRESV4-->IRSTR = 1	
5309	062734	012704	066300'		JSR PC,T20WFM	SETUP T20PK2 FOR WRITE FORMAT	
5310	062740	010465	000000		MOV @T20PK2,R4	GET WRITE SUBSYSTEM COMMAND PACKET	
5311	062744	004737	016146'		MOV R4,TSDB(R5)	SET THE PACKET ADDRESS TO EXECUTE	
5312	062750				JSR PC,CHKTSSR	WAIT FOR SSR TO SET	
5313	062764	103407			FORCERROR 112\$	FORCE ERROR IF FORCER=1	
5314	062766	010001			BCS 120\$	BR IF CARRY SET (GOOD RETURN)	
5315	062770				MOV RO,R1	SAVE CONTENTS OF TSSR	
5316	062770			112\$:	NEXT,ERRNO		
	062770	104455			ERRDF ERRNO,T208SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET	
	062772	001611				TRAP C\$ERDF	
	062774	065271'				.WORD 905	
	062776	011656'				.WORD T208SSR	
5317	063000	005237	002222'		INC FATFLG	SET FATAL ERROR FLAG	
5318	063004			120\$:	CKLOOP	LOOP ON ERROR, IF FLAG SET	
5319	063006	012700	000001	:	Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT	TRAP C\$CLP1	
5320	063012	012701	002312'		MOV @1,R0	WRITE 1 BYTE	
5321	063016	004737	065746'		MOV @DATA,R1	FIFO WRITE DATA ADDRESS	
5322	063022	012704	066300'		JSR PC,T20WFI	SETUP T20PK2 FOR WRITE FIFO	
5323	063026	010465	000000		MOV @T20PK2,R4	GET WRITE SUBSYSTEM COMMAND PACKET	
5324	063032	004737	016146'		MOV R4,TSDB(R5)	SET THE PACKET ADDRESS TO EXECUTE	
5325	063036				JSR PC,CHKTSSR	WAIT FOR SSR TO SET	
5326	063052	103407			FORCERROR 152\$	FORCE ERROR IF FORCER=1	
5327	063054	010001			BCS 160\$	BR IF CARRY SET (GOOD RETURN)	
5328	063056				MOV RO,R1	SAVE CONTENTS OF TSSR	
5329	063056				NEXT,ERRNO		
5330	063056			152\$:	ERRDF ERRNO,T205SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET	
	063056	104455				TRAP C\$ERDF	
	063060	001612				.WORD 906	
	063062	065162'				.WORD T205SSR	
	063064	011656'				.WORD PKTSSR	
5331	063066	005237	002222'		INC FATFLG	SET FATAL ERROR FLAG	
5332	063072			160\$:	CKLOOP	LOOP ON ERROR, IF FLAG SET	
5333	063072	104406				TRAP C\$CLP1	
5334				:	Do a READ FIFO with tape direction OUT to load tape out write latch		
5335	063074	012700	000001	:	(this is when wrong parity (IWP) is set)		
5336	063100	004737	065726'		MOV @1,R0	SET READ BYTE COUNT	
5337	063104	012704	066300'		JSR PC,T20RFIF	SETUP T20PK2 FOR READ FIFO	
5338	063110	010465	000000		MOV @T20PK2,R4	GET WRITE SUBSYSTEM COMMAND PACKET	
5339	063114	004737	016146'		MOV R4,TSDB(R5)	SET THE PACKET ADDRESS TO EXECUTE	
5340	063120				JSR PC,CHKTSSR	WAIT FOR SSR TO SET	
5341	063134	103407			FORCERROR 172\$	FORCE ERROR IF FORCER=1	
5342	063136	010001			BCS 180\$	BR IF CARRY SET (GOOD RETURN)	
5343	063140				MOV RO,R1	SAVE CONTENTS OF TSSR	
5344	063140			172\$:	NEXT,ERRNO		
	063140	104455			ERRDF ERRNO,T206SSR,PKTSSR	DEVICE FATAL SSR FAILED TO SET	
	063142	001613				TRAP C\$ERDF	
	063144	065226'				.WORD 907	
	063146	011656'				.WORD T206SSR	
5345	063150	005237	002222'		INC FATFLG	SET FATAL ERROR FLAG	
5346	063154			180\$:	CKLOOP	LOOP ON ERROR, IF FLAG SET	
	063154	104406				TRAP C\$CLP1	

```

5347      ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 (sets read strobe low)
5348      ; (Read Strobe sets PAR IN H [Parity Error])
5349 063156 005000      CLR      R0      ; IRESV4==>IRSTR = 0
5350 063160 004737 066002' JSR      PC,T20WFMT      ; SETUP T20PK2 FOR WRITE FORMAT
5351 063164 012704 066300' MOV      @T20PK2,R4      ; GET WRITE SUBSYSTEM COMMAND PACKET
5352 063170 010465 000000' MOV      R4,TSD8(R5)      ; SET THE PACKET ADDRESS TO EXECUTE
5353 063174 004737 016146' JSR      PC,CHKTSSR      ; WAIT FOR SSR TO SET
5354 063200      FORCERROR 192$      ; 000FORCE ERROR IF FORCER=1
5355 063214 103407      BCS      200$      ; BR IF CARRY SET (GOOD RETURN)
5356 063216 010001      MOV      R0,R1      ; SAVE CONTENTS OF TSSR
5357 063220      NEXT,ERRNO
5358 063220 192$: ERRDF  ERRNO,T208SSR,PKTSSR ; DEVICE FATAL SSR FAILED TO SET
      TRAP      C$ERDF
      .WORD      908
      .WORD      T208SSR
      .WORD      PKTSSR
5359 063230 005237 002222' INC      FATFLG      ; SET FATAL ERROR FLAG
5360 063234 104406 200$: CKLOOP      ; LOOP ON ERROR, IF FLAG SET
      TRAP      C$CLP1
5361      ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
5362 063236 012700 000001' MOV      @WF,I4RES,R0      ; IRESV4==>IRSTR = 1
5363 063242 004737 066002' JSR      PC,T20WFMT      ; SETUP T20PK2 FOR WRITE FORMAT
5364 063246 012704 066300' MOV      @T20PK2,R4      ; GET WRITE SUBSYSTEM COMMAND PACKET
5365 063252 010465 000000' MOV      R4,TSD8(R5)      ; SET THE PACKET ADDRESS TO EXECUTE
5366 063256 004737 016146' JSR      PC,CHKTSSR      ; WAIT FOR SSR TO SET
5367 063262      FORCERROR 212$      ; 000FORCE ERROR IF FORCER=1
5368 063276 103407      BCS      220$      ; BR IF CARRY SET (GOOD RETURN)
5369 063300 010001      MOV      R0,R1      ; SAVE CONTENTS OF TSSR
5370 063302      NEXT,ERRNO
5371 063302 212$: ERRDF  ERRNO,T208SSR,PKTSSR ; DEVICE FATAL SSR FAILED TO SET
      TRAP      C$ERDF
      .WORD      909
      .WORD      T208SSR
      .WORD      PKTSSR
5372 063312 005237 002222' INC      FATFLG      ; SET FATAL ERROR FLAG
5373 063316 104406 220$: CKLOOP      ; LOOP ON ERROR, IF FLAG SET
      TRAP      C$CLP1
5374      ; Do a Write Subsystem READ STATUS
5375 063320 004737 065666' JSR      PC,T20SRD      ; SETUP PACKET FOR READ STATUS
5376 063324 012704 066300' MOV      @T20PK2,R4      ; GET WRITE SUBSYSTEM COMMAND PACKET
5377 063330 010465 000000' MOV      R4,TSD8(R5)      ; SET THE PACKET ADDRESS TO EXECUTE
5378 063334 004737 016146' JSR      PC,CHKTSSR      ; WAIT FOR SSR TO SET
5379 063340      FORCERROR 232$      ; 000FORCE ERROR IF FORCER=1
5380 063354 103407      BCS      240$      ; BR IF CARRY SET (GOOD RETURN)
5381 063356 010001      MOV      R0,R1      ; SAVE CONTENTS OF TSSR
5382 063360      NEXT,ERRNO
5383 063360 232$: ERRDF  ERRNO,T203SSR,PKTSSR ; DEVICE FATAL SSR FAILED TO SET
      TRAP      C$ERDF
      .WORD      910
      .WORD      T203SSR
      .WORD      PKTSSR
5384 063370 005237 002222' INC      FATFLG      ; SET FATAL ERROR FLAG
5385 063374 104406 240$: CKLOOP      ; LOOP ON ERROR, IF FLAG SET
      TRAP      C$CLP1
5386      ; If Read Data parity error NOT=1 Then Print Error
5387 063376 004737 066074' JSR      PC,T20SETEXP      ; SET WORDS 0-7 EXPD+RECV (NOT TESTING)
5388 063402 012701 064622' MOV      @T20EXSTA,R1      ; GET EXPECTED READ STATUS

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 9: READ/WRITE DATA PARITY TEST

SEQ 201

5389	063406	012702	066172'	MOV	0T20BFSTA,R2	;GET RECV READ STATUS	
5390	063412	011211		MOV	(R2),(R1)	;SET EXPD WORD 08 = RECV TEMP	
5391	063414	016261	000002 000002	MOV	2(R2),2(R1)	;SET EXPD WORD 09 = RECV (NOT TESTED)	
5392	063422	052711	100000	BIS	0S1.PARERR,(R1)	;SET EXP PAR ERR =1	
5393	063426	005000		CLR	R0	;HIGH RECV ADDRESS FOR CKMSG2	
5394	063430	012701	066152'	MOV	0T20BFR,R1	;LOW RECV ADDRESS FOR CKMSG2	
5395	063434	012702	064602'	MOV	0T20EXP,R2	;EXPD ADDR -SS	
5396	063440	012703	000024	MOV	020.,R3	;NUMBER OF BYTES TO COMPARE	
5397	063444	004737	011310'	JSR	PC,CKMSG2	;EXPD EQUAL RECV?	
5398	063450			FORCERROR	252\$,NOTSSR	;000	
5399	063460	103404		BCS	260\$	;BR IF YES	
5400	063462			NEXT,ERRNO			
5401	063462			252\$:	ERRHRD ERRNO,T20SWP,MSGSTAT	;REPORT ERROR	
	063462	104456				TRAP	C\$ERHRD
	063464	001617				.WORD	911
	063466	065337'				.WORD	T20SWP
	063470	012160'				.WORD	MSGSTAT
5402	063472			260\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	063472	104406				TRAP	C\$CLP1
5403				:	Do a Write Misc to RESET FIFO		
5404	063474	012700	000020	MOV	0MS.RSFIF,R0	;SET RESET FIFO COMMAND	
5405	063500	004737	066022'	JSR	PC,T20WMISC	;SETUP T20PK2 FOR WRITE MISC	
5406	063504	012704	066300'	MOV	0T20PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
5407	063510	010465	000000	MOV	R4,T20B(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
5408	063514	004737	016146'	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET	
5409	063520			FORCERROR	282\$	;000FORCE ERROR IF FORCER=1	
5410	063534	103407		BCS	290\$	;BR IF CARRY SET (GOOD RETURN)	
5411	063536	010001		MOV	R0,R1	;SAVE CONTENTS OF TSSR	
5412	063540			NEXT,ERRNO			
5413	063540			282\$:	ERRDF ERRNO,T202SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	
	063540	104455				TRAP	C\$ERDF
	063542	001620				.WORD	912
	063544	065006'				.WORD	T202SSR
	063546	011656'				.WORD	PKTSSR
5414	063550	005237	002222'	INC	FATFLG	;SET FATAL ERROR FLAG	
5415	063554			290\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	063554	104406				TRAP	C\$CLP1
5416				:	Do a Write Subsystem READ STATUS		
5417				:	If Read Data parity error NOT=0	Then Print Error	
5418	063556	004737	066074'	JSR	PC,T20SETEXP	;SET WORDS 0-7 EXPD=RECV (NOT TESTING)	
5419	063562	012701	064622'	MOV	0T20EXSTA,R1	;GET EXPECTED READ STATUS	
5420	063566	012702	066172'	MOV	0T20BFSTA,R2	;GET RECV READ STATUS	
5421	063572	011211		MOV	(R2),(R1)	;SET EXPD WORD 08 = RECV TEMP	
5422	063574	016261	000002 000002	MOV	2(R2),2(R1)	;SET EXPD WORD 09 = RECV (NOT TESTED)	
5423	063602	042711	100000	BIC	0S1.PARERR,(R1)	;SET EXP PAR ERR =0	
5424	063606	005000		CLR	R0	;HIGH RECV ADDRESS FOR CKMSG2	
5425	063610	012701	066152'	MOV	0T20BFR,R1	;LOW RECV ADDRESS FOR CKMSG2	
5426	063614	012702	064602'	MOV	0T20EXP,R2	;EXPD ADDRESS	
5427	063620	012703	000024	MOV	020.,R3	;NUMBER OF BYTES TO COMPARE	
5428	063624	004737	011310'	JSR	PC,CKMSG2	;EXPD EQUAL RECV?	
5429	063630			FORCERROR	302\$,NOTSSR	;000	
5430	063640	103404		BCS	320\$	;BR IF YES	
5431	063642			NEXT,ERRNO			
5432	063642			302\$:	ERRHRD ERRNO,T20RSF,MSGSTAT	;REPORT ERROR	
	063642	104456				TRAP	C\$ERHRD
	063644	001621				.WORD	913
	063646	065446'				.WORD	T20RSF

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-81 17:02
 TEST 9: READ/WRITE DATA PARITY TEST

SEQ 202

5433	063650	012160'		320\$:	CKLOOP		.WORD MSGSTAT
	063652					;LOOP ON ERROR, IF FLAG SET	
	063652	104406				TRAP C\$CLP1	
5434				:	(* Verify Data can be transferred without a Parity Error *)		
5435				:	Do a WRITE FORMAT to set IRESV4-->IRSTR = 1 (sets read strobe high)		
5436	063654	012700	000001		MOV @WF.I4RES,R0	;IRES/4-->IRSTR = 1	
5437	063660	004737	066002'		JCR PC,T20WFMT	;SETUP T20PK2 FOR WRITE FORMAT	
5438	063664	012704	066300'		MOV @T20PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
5439	063670	010465	000000		MOV R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
5440	063674	004737	016146'		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET	
5441	063700				FORCERROR 332\$	;BDDFORCE ERROR IF FORCER=1	
5442	063714	103407			BCS 340\$	;BR IF CARRY SET (GOOD RETURN)	
5443	063716	010001			MOV R0,R1	;SAVE CONTENTS OF TSSR	
5444	063720				NEXT,ERRNO		
5445	063720			332\$:	ERRDF ERRNO,T20BSSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	
	063720	104455				TRAP C\$ERDF	
	063721	001622				.WORD 914	
	063724	065271'				.WORD T20BSSR	
	063726	011656'				.WORD PKTSSR	
5446	063730	005237	002222'		INC FATFLG	;SET FATAL ERROR FLAG	
5447	063734			340\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	063734	104406				TRAP C\$CLP1	
5448				:	Do a WRITE NPR to set loopback and tape direction OUT and		
5449				:	and CLEAR Write Wrong Parity.		
5450	063736	012700	000100		MOV @NP.OUT,R0	;SET TAPE DIRECTION OUT	
5451	063742	052700	000040		BIS @NP.LOOP,R0	;SET LOOPBACK	
5452	063746	052700	000020		BIS @NP.WRP,R0	;CLEAR WRITE WRONG PARITY (INVERTED)	
5453	063752	004737	065706'		JSR PC,T20WNPR	;SETUP T20PK2 FOR WRITE NPR	
5454	063756	012704	066300'		MOV @T20PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
5455	063762	010465	000000		MOV R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
5456	063766	004737	016146'		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET	
5457	063772				FORCERROR 352\$	;BDDFORCE ERROR IF FORCER=1	
5458	064006	103407			BCS 360\$	;BR IF CARRY SET (GOOD RETURN)	
5459	064010	010001			MOV R0,R1	;SAVE CONTENTS OF TSSR	
5460	064012				NEXT,ERRNO		
5461	064012			352\$:	ERRDF ERRNO,T204SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	
	064012	104455				TRAP C\$ERDF	
	064014	001623				.WORD 915	
	064016	065117'				.WORD T204SSR	
	064020	011656'				.WORD PKTSSR	
5462	064022	005237	002222'		INC FATFLG	;SET FATAL ERROR FLAG	
5463	064026			360\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	064026	104406				TRAP C\$CLP1	
5464				:	Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT		
5465	064030	012700	000001		MOV @1,R0	;WRITE 1 BYTE	
5466	064034	012701	002312'		MOV @DATA,R1	;FIFO WRITE DATA ADDRESS	
5467	064040	004737	065746'		JSR PC,T20WFIF	;SETUP T20PK2 FOR WRITE FIFO	
5468	064044	012704	066300'		MOV @T20PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET	
5469	064050	010465	000000		MOV R4,TSD8(R5)	;SET THE PACKET ADDRESS TO EXECUTE	
5470	064054	004737	016146'		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET	
5471	064060				FORCERROR 372\$	;BDDFORCE ERROR IF FORCER=1	
5472	064074	103407			BCS 380\$	;BR IF CARRY SET (GOOD RETURN)	
5473	064076	010001			MOV R0,R1	;SAVE CONTENTS OF TSSR	
5474	064100				NEXT,ERRNO		
5475	064100			372\$:	ERRDF ERRNO,T205SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	
	064100	104455				TRAP C\$ERDF	
	064102	001624				.WORD 916	

Address	Offset	Label	Operation	Comments
064104	065162'			.WORD T205SSR
064106	011656'			.WORD PKTSSR
5476	064110	005237 002222'	INC FATFLG	;SET FATAL ERROR FLAG
5477	064114	380\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
064114	104406			TRAP C\$CLP1
5478				
5479	064116	012700 000001	Do a READ FIFO with tape direction OUT to load tape out write latch	
5480	064122	004737 065726'	MOV #1,R0	;SET READ BYTE COUNT
5481	064126	012704 066300'	JSR PC,T20RFIF	;SETUP T20PK2 FOR READ FIFO
5482	064132	010465 000000	MOV #T20PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
5483	064136	004737 016146'	MOV R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE
5484	064142		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET
5485	064156	103407	FORCERROR 392\$	;BDFORCE ERROR IF FORCER=1
5486	064160	010001	RCS 400\$	;BR IF CARRY SET (GOOD RETURN)
5487	064162		MOV R0,R1	;SAVE CONTENTS OF TSSR
5488	064162	392\$:	NEXT,ERRNO	
064162	104455		ERRDF ERRNO,T206SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
064164	001625			TRAP C\$ERDF
064166	065226'			.WORD 917
064170	011656'			.WORD T206SSR
5489	064172	005237 002222'	INC FATFLG	;SET FATAL ERROR FLAG
5490	064176	400\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
064176	104406			TRAP C\$CLP1
5491				
5492			Do a WRITE FORMAT to set IRESV4-->IRSTR = 0 (sets read strobe low)	
5493	064200	005000	(Read Strobe sets PAR IN H [Parity Error])	
5494	064202	004737 066002'	CLR R0	;IRESV4-->IRSTR = 0
5495	064206	012704 066300'	JSR PC,T20WFM1	;SETUP T20PK2 FOR WRITE FORMAT
5496	064212	010465 000000	MOV #T20PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
5497	064216	004737 016146'	MOV R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE
5498	064222		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET
5499	064236	103407	FORCERROR 412\$	;BDFORCE ERROR IF FORCER=1
5500	064240	010001	BCS 420\$	;BR IF CARRY SET (GOOD RETURN)
5501	064242		MOV R0,R1	;SAVE CONTENTS OF TSSR
5502	064242	412\$:	NEXT,ERRNO	
064242	104455		ERRDF ERRNO,T208SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
064244	001626			TRAP C\$ERDF
064246	065271'			.WORD 918
064250	011656'			.WORD T208SSR
5503	064252	005237 002222'	INC FATFLG	;SET FATAL ERROR FLAG
5504	064256	420\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
064256	104406			TRAP C\$CLP1
5505				
5506	064260	012700 000001	Do a WRITE FORMAT to set IRESV4-->IRSTR = 1 (sets read strobe high)	
5507	064264	004737 066002'	MOV #WF.I4RES,R0	;IRESV4-->IRSTR = 1
5508	064270	012704 066300'	JSR PC,T20WFM1	;SETUP T20PK2 FOR WRITE FORMAT
5509	064274	010465 000000	MOV #T20PK2,R4	;GET WRITE SUBSYSTEM COMMAND PACKET
5510	064300	004737 016146'	MOV R4,TSDB(R5)	;SET THE PACKET ADDRESS TO EXECUTE
5511	064304		JSR PC,CHKTSSR	;WAIT FOR SSR TO SET
5512	064320	103407	FORCERROR 432\$	;BDFORCE ERROR IF FORCER=1
5513	064322	010001	BCS 440\$	;BR IF CARRY SET (GOOD RETURN)
5514	064324		MOV R0,R1	;SAVE CONTENTS OF TSSR
5515	064324	432\$:	NEXT,ERRNO	
064324	104455		ERRDF ERRNO,T208SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
064326	001627			TRAP C\$ERDF
064330	065271'			.WORD 919
064332	011656'			.WORD T208SSR
				.WORD PKTSSR

TSVS - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 9: READ/WRITE DATA PARITY TEST

SEQ 204

```

5516 064334 005237 002222'      INC      FATFLG      ;SET FATAL ERROR FLAG
5517 064340      440$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      064340 104406      TRAP      C$CLP1
5518
5519      ; Do a Write Subsystem READ STATUS
5520 064342 004737 065666'      JSR      PC,T20SRD      ;SETUP PACKET FOR READ STATUS
5521 064346 012704 066300'      MOV      @T20FK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
5522 064352 010465 000000'      MOV      R4,T20B(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
5523 064356 004737 016146'      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
5524 064362      FORCERROR 452$      ;###FORCE ERROR IF FORCER=1
5525 064376 103407      BCS      460$      ;BR IF CARRY SET (GOOD RETURN)
5526 064400 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
5527 064402      NEXT,ERRNO
5528 064402      452$: ERDRF  ERRNO,T203SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      064402 104455      TRAP      C$ERDF
      064404 001630      .WORD      920
      064406 065052'      .WORD      T203SSR
      064410 011656'      .WORD      PKTSSR
5529 064412 005237 002222'      INC      FATFLG      ;SET FATAL ERROR FLAG
5530 064416      460$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      064416 104406      TRAP      C$CLP1
5531      ; If Read Data parity error NOT=0 Then Print Error
5532 064420 004737 066074'      JSR      PC,T20SETEXP      ;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
5533 064424 012701 064622'      MOV      @T20EXSTA,R1      ;GET EXPECTED READ STATUS
5534 064430 012702 066172'      MOV      @T20BFSTA,R2      ;GET RCV READ STATUS
5535 064434 011211      MOV      (R2),(R1)      ;SET EXPD WORD #8 = RCV TEMP
5536 064436 016261 000002 000002      MOV      2(R2),2(R1)      ;SET EXPD WORD #9 = RCV (NOT TESTED)
5537 064444 042711 100000      BIC      @S1.PARERR,(R1)      ;SET EXP PAR ERR =0
5538 064450 005000      CLR      R0      ;HIGH RCV ADDRESS FOR CKMSG2
5539 064452 012701 066152'      MOV      @T20BFR,R1      ;LOW RCV ADDRESS FOR CKMSG2
5540 064456 012702 064602'      MOV      @T20EXP,R2      ;EXPD ADDRESS
5541 064462 012703 000024      MOV      @20,R3      ;NUMBER OF BYTES TO COMPARE
5542 064466 004737 011310'      JSR      PC,CKMSG2      ;EXPD EQUAL RCV?
5543 064472      FORCERROR 472$,NOTSSR      ;###
5544 064502 103404      BCS      480$      ;BR IF YES
5545 064504      NEXT,ERRNO
5546 064504      472$: ERHRD  ERRNO,T20CWP,MSGSTAT      ;REPORT ERROR
      064504 104456      TRAP      C$ERHRD
      064506 001631      .WORD      921
      064510 065547'      .WORD      T20CWP
      064512 012160'      .WORD      MSGSTAT
5547 064514      480$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      064514 104406      TRAP      C$CLP1
5548
5549 064516      FORCEEXIT 555$      ;###
5550 064526 013703 002316'      MOV      TSTPTR,R3      ;RESTORE CURRENT TSTBLK POINTER
5551 064532 020327 003062'      CMP      R3,@TBLEND      ;END OF TSTBLK?
5552 064536 103002      BFI S 555$      ;BR IF YES
5553 064540 000137 062614'      JMP      100$      ;DO ANOTHER TSTBLK PATTERN
5554 064544      555$:
5555
5556 064544      ENDSUB      ;////////// END SUBTEST //////////
      064544      L10074:
      064544 104403      TRAP      C$ESUB
5557
5558 064546 005737 002222'      TST      FATFLG      ;ANY FATAL ERRORS ?
5559 064552 001402      BEQ      560$      ;BRANCH IF NOT

```

```

5560 064554 004737 017014'      JSR      PC,CKDROP      ;TRY TO DROP THE UNIT
5561 064560      560$:      JSR      PC,TSTLOOP      ;DO ITERATIONS?
5562 064560 004737 016270'      BCC      565$      ;BR IF NO
5563 064564 103002      JMP      T18LOOP      ;LOOP UNTIL ITERATIONS DONE
5564 064566 000137 050246'      565$:      EXIT      TST      ;////////// EXIT TEST //////////
5565 064572      TRAP      C$EXIT
5566 064572 104432      .WORD      L10073-.
5567 064574 001620
5568
5569      ;+
5570      ;LOCAL STORAGE FOR THIS TEST
5571      ;-
5572
5573      T20MSK:      ;MASK OF UNTESTED BITS IN READ STATUS
5574      ;UNTESTED BITS ARE SET TO 1
5575 064576      377      .BYTE      +C<000>      ;BYTE 0 MASK
5576 064577      037      .BYTE      +C<340>      ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
5577 064600      360      .BYTE      +C<017>      ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
5578 064601      000      .BYTE      0      ;MAKE IT EVEN
5579
5580      T20EXP:      ;BEGIN EXPECTED DATA BUFFER
5581 064602 000000      .WORD      0      ;MESSAGE TYPE
5582 064604 000000      .WORD      0      ;DATA FIELD LENGTH
5583 064606 000000      .WORD      0      ;RBPGR
5584 064610 000000      .WORD      0      ;XST0
5585 064612 000000      .WORD      0      ;XST1
5586 064614 000000      .WORD      0      ;XST2
5587 064616 000000      .WORD      0      ;XST3
5588 064620 000000      .WORD      0      ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
5589 064622      T20EXSTA: .BLKB 64.      ;EXPECTED READ STATUS AND WRITE FIFO DATA
5590 064722      T20EXEND:      ;END EXPECTED DATA BUFFER
5591
5592      ;+
5593      ;LOCAL TEXT MESSAGES FOR TEST
5594      ;-
5595 064722      122      145      141      TST20ID:      .ASCIIZ      'Read/Write Data Parity'
5596 064751      127      122      111      T20SSR: .ASCIIZ      'WRITE CHARACTERISTICS Failed'
5597 065006      127      122      111      T202SSR: .ASCIIZ      'WRITE SUBSYSTEM (Write Misc) Failed'
5598 065052      127      122      111      T203SSR: .ASCIIZ      'WRITE SUBSYSTEM (Read Status) Failed'
5599 065117      127      122      111      T204SSR: .ASCIIZ      'WRITE SUBSYSTEM (Write Nor) Failed'
5600 065162      127      122      111      T205SSR: .ASCIIZ      'WRITE SUBSYSTEM (Write FIFO) Failed'
5601 065226      127      122      111      T206SSR: .ASCIIZ      'WRITE SUBSYSTEM (Read FIFO) Failed'
5602 065271      127      122      111      T208SSR: .ASCIIZ      'WRITE SUBSYSTEM (Write Format) Failed'
5603 065337      122      145      141      T20SWP: .ASCIIZ      'Read Data Parity Error (PARERR) Failed to Set after Write Wrong Parity'
5604 065446      122      145      141      T20RSF: .ASCIIZ      'Read Data Parity Error (PARERR) Failed to Clear after RESET FIFO'
5605 065547      122      145      141      T20CWP: .ASCIIZ      'Read Data Parity Error (PARERR) occurred in Data Loopback'
5606      .EVEN
5607
5608
5609      ;+
5610      ; CLEAR MESSAGE BUFFER
5611
5612      T20CLRBUF:      ;
5613      SAVREG      ;SAVE R1 R5 UNTIL NEXT RETURN
5614 065642 012701 0066152'      MOV      #T20BFR,R1      ;GET MESSAGE BUFFER ADDRESS
5615 065652 012702 000120      MOV      #T20BEND-T20BFR,R2      ;SIZE OF MESSAGE BUFFER IN BYTES

```

L16

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 9: READ/WRITE DATA PARITY TEST

SEQ 206

5615	065656	105021		10\$: CLR B (R1)+	;CLEAR A BYTE
5616	065660	005302		DEC R2	;DONE?
5617	065662	003375		BGT 10\$	;BR IF NO
5618	065664	000207		RTS PC	;RETURN
5619					
5620					
5621				;+ SETUP T20PK2 PACKET FOR READ STATUS	
5622					
5623	065666			T20SRD:	
5624	065666	004737	065642'	JSR PC,T20CLRBUF	;CLEAR MESSAGE BUFFER
5625	065672	012700	066310'	MOV #T20DT2,R0	;WRITE SUBSYSTEM DATA BUFFER
5626	065676	112720	000005	MOVB #PW,RDSTATUS,(R0)+	;STORE READ STATUS COMMAND IN BSELO
5627	065702	105010		CLR B (R0)	;CLEAR BSEL1
5628	065704	000207		RTS PC	;RETURN
5629					
5630					
5631					
5632				;+ SETUP T20PK2 PACKET FOR WRITE NPR	
5633					
5634				; INPUT:	
5635				; R0 CONTAINS BSEL1 NPR DATA	
5636					
5637					
5638	065706			T20WNPR:	
5639	065706	004737	065642'	JSR PC,T20CLRBUF	;CLEAR MESSAGE BUFFER
5640	065712	012701	066310'	MOV #T20DT2,R1	;WRITE SUBSYSTEM DATA BUFFER
5641	065716	112721	000011	MOVB #PW,WNPR,(R1)+	;STORE WRITE NPR IN BSELO
5642	065722	110011		MOVB R0,(R1)	;STORE NPR DATA IN BSEL1
5643	065724	000207		RTS PC	;RETURN
5644					
5645					
5646				;+ SETUP T20PK2 PACKET FOR READ FIFO	
5647					
5648				; INPUT:	
5649				; R0 CONTAINS SEL2 BYTE COUNT	
5650					
5651	065726			T20RFIF:	
5652	065726	004737	065642'	JSR PC,T20CLRBUF	;CLEAR MESSAGE BUFFER
5653	065732	012701	066310'	MOV #T20DT2,R1	;WRITE SUBSYSTEM DATA BUFFER
5654	065736	112721	000003	MOVB #PW,RFIFO,(R1)+	;STORE READ FIFO IN BSELO
5655	065742	110021		MOVB R0,(R1)+	;STORE BYTE COUNT IN BSEL1
5656	065744	000207		RTS PC	;RETURN
5657					
5658				;+ SETUP T20PK2 PACKET FOR WRITE FIFO	
5659					
5660				; INPUT:	
5661				; R0 CONTAINS BYTE COUNT	
5662				; R1 CONTAINS DATA PATTERN BLOCK ADDRESS	
5663					
5664	065746			T20WFIF:	
5665	065746			SAVREG	;SAVE R1-R5 UNTIL NEXT RETURN
5666	065752	004737	065642'	JSR PC,T20CLRBUF	;CLEAR MESSAGE BUFFER
5667	065756	012702	066310'	MOV #T20DT2,R2	;WRITE SUBSYSTEM DATA BUFFER
5668	065762	112722	000004	MOVB #PW,WFIFO,(R2)+	;STORE WRITE FIFO IN BSELO
5669	065766	110022		MOVB R0,(R2)+	;STORE BYTE COUNT IN BSEL1
5670	065770	005022		CLR (R2)+	;CLEAR SEL2 (UNUSED)
5671	065772	112122		10\$: MOVB (R1)+,(R2)+	;STORE DATA PATTERN BYTE

M16

TSVS - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
 TEST 9: READ/WRITE DATA PARITY TEST

SEQ 207

5672	065774	005300		DEC	R0		;DONE ALL BYTES?
5673	065776	003375		BGT	10\$		;BR IF NO
5674	066000	000207		RTS	PC		;RETURN
5675							
5676							
5677							
5678							
5679							
5680							
5681							
5682	066002						
5683	066002	004737	065642'				
5684	066006	012701	066310'				
5685	066012	112721	000007				
5686	066016	110021					
5687	066020	000207					
5688							
5689							
5690							
5691							
5692							
5693	066022						
5694	066022	012701	066310'				
5695	066026	112721	000010				
5696	066032	110011					
5697	066034	000207					
5698							
5699							
5700							
5701	066036						
5702	066036	012700	066310'				
5703	066042	112720	000010				
5704	066046	112710	000200				
5705	066052	000207					
5706							
5707							
5708							
5709	066054						
5710	066054	012701	064602'				
5711	066060	012700	000120				
5712	066064	105021					
5713	066066	005300					
5714	066070	003375					
5715	066072	000207					
5716							
5717							
5718							
5719							
5720	066074						
5721	066074	012702	064602'				
5722	066100	012703	066152'				
5723	066104	012700	000010				
5724	066110	012322					
5725	066112	005300					
5726	066114	003375					
5727	066116	000207					
5728							


```

;+
; 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 BSELO
  MOVB     R0,(R1)+          ;STORE DATA WORD IN BSEL1
  RTS      PC                ;RETURN

;+
; SETUP T20PK2 PACKET FOR WRITE MISC.
;   R0 CONTAINS WRITE MISC DATA
;-
T20WMISC:
  MOV      #T20DT2,R1       ;WRITE SUBSYSTEM DATA BUFFER
  MOVB     #PW.WMISC,(R1)+   ;STORE WRITE MISCELLANEOUS IN BSELO
  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 BSELO
  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
  CLR      (R1)+             ;CLEAR A BYTE
  DEC      R0                ;DONE
  BGT      10$               ;BR IF NO
  RTS      PC                ;RETURN

10$:
;+
; Set WORDS 0-7 of expd message BUFFER = to recv since not testing
;-
T20SETEXP:
  MOV      #T20EXP,R2       ;GET EXPD
  MOV      #T20BFR,R3       ;GET READ STATUS RECV BUFFER
  MOV      #8,R0            ;SET WORDS 0-7 EXP=RECV
  MOV      (R3)+,(R2)+      ;SET EXPD=RECV
  DEC      R0               ;DONE WORDS 0-7 WORDS?
  BGT      5$               ;BR IF NO
  RTS      PC                ;RETURN

5$:

```

....B1	PKTMES - PRINT TSSR....B5	TEST 3: SUBTEST 4: N....B9	TEST 6: SUBTEST 6: B13
....C1	FIFEXP - PRINT FIFO....C5	TEST 3: SUBTEST 4: N....C9	TEST 6: SUBTEST 6: C13
....D1	MSGSTAT - PRINT STAT....D5	TEST 3: SUBTEST 4: N....D9	TEST 6: SUBTEST 6: D13
....E1	MSGSUB - PRINT WRITE....E5	TEST 3: SUBTEST 4: N....E9	TEST 6: SUBTEST 6: E13
....F1	PRAMPKT - PRINT RAM F5	TEST 3: SUBTEST 4: N....F9	TEST 6: SUBTEST 6: F13
....G1	PRMESS - PRINT CONT....G5	TEST 3: SUBTEST 4: N....G9	TEST 6: SUBTEST 6: G13
....H1	PRMSGEXP - PRINT EXP....H5	TEST 3: SUBTEST 4: N....H9	TEST 7: STATIC TRAN....H13
....I1	PRMSGEXP - PRINT EXP....I5	TEST 3: SUBTEST 4: N....I9	TEST 7: STATIC TRAN....I13
....J1	PRBYTEXP - PRINT ERR....J5	TEST 4: RAM EXERCIS....J9	TEST 7: STATIC TRAN....J13
....K1	EXPBREC - PRINT EXPD....K5	TEST 4: RAM EXERCIS....K9	TEST 7: STATIC TRAN....K13
....L1	RAMTADD - PRINT TEST....L5	TEST 4: RAM EXERCIS....L9	TEST 7: STATIC TRAN....L13
....M1	BADSSR - PRINT TSSR M5	TEST 4: RAM EXERCIS....M9	TEST 8: SUBTEST 1: M13
....N1	SOFINIT - SOFT INITI....N5	TEST 4: RAM EXERCIS....N9	TEST 8: SUBTEST 1: N13
....B2	CHKAMB - CHECK TSSR....B6	TEST 4: RAM EXERCIS....B10	TEST 8: SUBTEST 1: B14
....C2	INTR - INTERRUPT C6	TEST 4: RAM EXERCIS....C10	TEST 8: SUBTEST 1: C14
....D2	CHKTSSR - CHECK TSSR....D6	TEST 4: RAM EXERCIS....D10	TEST 8: SUBTEST 1: D14
....E2	XNXM - CHECK FOR E6	TEST 5: SUBTEST 1: E10	TEST 8: SUBTEST 1: E14
....F2	TSTSETUP - PRINT TES....F6	TEST 5: SUBTEST 1: F10	TEST 8: SUBTEST 1: F14
PROGRAM HEADER	TSTEND - PRINT ERRO....G6	TEST 5: SUBTEST 1: G10	TEST 8: SUBTEST 1: G14
DISPATCH TABLE	CKDROP - CHECK IF U....H6	TEST 5: SUBTEST 1: H10	TEST 8: SUBTEST 2: H14
SOFTWARE P-TABLE	SETMAP - SETUP PAR6....I6	TEST 5: SUBTEST 1: I10	TEST 8: SUBTEST 2: I14
GLOBAL EQUATES SECTI....J2	FILLMEM - FILL MEMOR....J6	TEST 5: SUBTEST 2: J10	TEST 8: SUBTEST 2: J14
MEMORY MANAGEMENT DE....K2	CMPMEM - COMPARE ME....K6	TEST 5: SUBTEST 2: K10	TEST 8: SUBTEST 2: K14
MEMORY MANAGEMENT DE....L2	REGSAV - SAVE R1-R5....L6	TEST 5: SUBTEST 2: L10	TEST 8: SUBTEST 2: L14
TSU05 REGISTER AND P....M2	GETPAT - GET 8 BIT M6	TEST 5: SUBTEST 2: M10	TEST 8: SUBTEST 3: M14
TSU05 REGISTER AND P....N2	GETSEL - ISSUE MENU....N6	TEST 5: SUBTEST 2: N10	TEST 8: SUBTEST 3: N14
TSU05 REGISTER AND P....B3	CHKMAN - CHECK MANU....B7	TEST 5: SUBTEST 2: B11	TEST 8: SUBTEST 3: B15
TSU05 REGISTER AND P....C3	KTINIT - SETUP K111....C7	TEST 5: SUBTEST 2: C11	TEST 8: SUBTEST 3: C15
TSU05 REGISTER AND P....D3	KTINIT - SETUP K111....D7	TEST 6: FIFO EXERCIS....D11	TEST 8: SUBTEST 3: D15
TSU05 REGISTER AND P....E3	KTINIT - SETUP K111....E7	TEST 6: SUBTEST 1: E11	TEST 8: SUBTEST 4 L....E15
SPECTAL MACROS AND O....F3	KTINIT - SETUP K111....F7	TEST 6: SUBTEST 1: F11	TEST 8: SUBTEST 4 L....F15
GLOBAL DATA SECTION G3	INITIALIZE SECTION G7	TEST 6: SUBTEST 2: G11	TEST 8: SUBTEST 4 L....G15
TSTBLK - TEST DATA H3	INITIALIZE SECTION H7	TEST 6: SUBTEST 2: H11	TEST 8: SUBTEST 4 L....H15
GLOBAL ENVIRONMENT S....I3	INITIALIZE SECTION I7	TEST 6: SUBTEST 2: I11	TEST 8: SUBTEST 4 L....I15
GLOBAL TEXT MESSAGES....J3	ADD AND DROP UNITS S....J7	TEST 6: SUBTEST 2: J11	TEST 8: SUBTEST 4 L....J15
GLOBAL ERROR REPORT K3	CLEAN-UP AND REPORT K7	TEST 6: SUBTEST 2: K11	TEST 8: SUBTEST 4 L....K15
PRITSSR - PRINT TSSR....L3	CLEAN-UP AND REPORT L7	TEST 6: SUBTEST 3: L11	TEST 8: SUBTEST 4 L....L15
PRITSSR - PRINT TSSR... M3	CLEAN-UP AND REPORT M7	TEST 6: SUBTEST 3: M11	TEST 8: SUBTEST 4 L....M15
PRIPKT - PRINT THE N3	TEST 1: INITIALIZE N7	TEST 6: SUBTEST 3: N11	TEST 8: SUBTEST 4 L....N15
PRIBXOR - PRINT EXPD....B4	TEST 1: INITIALIZE B8	TEST 6: SUBTEST 3: B12	TEST 8: SUBTEST 4 L....B16
PRIXOR - PRINT EXPD....C4	TEST 1: INITIALIZE C8	TEST 6: SUBTEST 3: C12	TEST 9: READ/WRITE C16
PRIADD - PRINT MEMO....D4	TEST 2: BASIC WRITE D8	TEST 6: SUBTEST 3: D12	TEST 9: READ/WRITE D16
PRITADD - PRINT MEMO....L4	TEST 2: BASIC WRITE E8	TEST 6: SUBTEST 4: E12	TEST 9: READ/WRITE E16
SPACE - SPACE RECO....F4	TEST 2: BASIC WRITE F8	TEST 6: SUBTEST 4: F12	TEST 9: READ/WRITE F16
WRTHR - WRITE CHAR....G4	TEST 2: BASIC WRITE G8	TEST 6: SUBTEST 5: G12	TEST 9: READ/WRITE G16
WRTHR - WRITE CHAR....H4	TEST 2: BASIC WRITE H8	TEST 6: SUBTEST 5: H12	TEST 9: READ/WRITE H16
REWIND - POSITION T....I4	TEST 3: SUBTEST 1: G....I8	TEST 6: SUBTEST 5: I12	TEST 9: READ/WRITE I16
CKRAM2 - COMPARE RA....J4	TEST 3: SUBTEST 1: G....J8	TEST 6: SUBTEST 5: J12	TEST 9: READ/WRITE J16
CKRAM2 - COMPARE RA....K4	TEST 3: SUBTEST 1: G....K8	TEST 6: SUBTEST 5: K12	TEST 9: READ/WRITE K16
CKMSG - COMPARE WR....L4	TEST 3: SUBTEST 2: M....L8	TEST 6: SUBTEST 5: L12	TEST 9: READ/WRITE L16
CKMSG2 - COMPARE EX....M4	TEST 3: SUBTEST 3: C....M8	TEST 6: SUBTEST 6: M12	TEST 9: READ/WRITE M16
CKMSG2 - COMPARE EX....N4	TEST 3: SUBTEST 3: C....N8	TEST 6: SUBTEST 6: N12	

```

5729
5730
5732 066120          .BLKB  10-<.-TSV2&7>
5734
5735      ;WRITE CHARACTERISTICS COMMAND PACKET
5736      ;
5737 066130      T20PACKET:      ;COMMAND PACKET FOR TEST
5738 066130 100004      .WORD  100004      ;WRITE CHARACTERISTICS COMMAND, WITH ACK
5739 066132 066140'    .WORD  T20DATA      ;ADDRESS OF CHARACTERISTICS BLOCK
5740 066134 000000      .WORD  0
5741 066136 000012      .WORD  10.         ;MINIMUM MESSAGE PACKET SIZE
5742
5743 066140      T20DATA:      ;CHARACTERISTICS DATA BLOCK
5744 066140 066152'    .WORD  T20BFR        ;ADDRESS OF MESSAGE BUFFER
5745 066142 000000      .WORD  0
5746 066144 000024      .WORD  20.         ;LENGTH OF MESSAGE BUFFER
5747 066146 000000      .WORD  0
5748 066150 000007      .WORD  7           ;ESS,ENB,EAI,ERI
5749                                     ;EXTENDED FEATURES UNIT NO.
5750
5751      ;MESSAGE BUFFER FOR ALL TEST 17 COMMANDS
5752
5753 066152      T20BFR:      ;BEGIN MESSAGE BUFFER
5754 066152 000000      .WORD  0             ;MESSAGE TYPE
5755 066154 000000      .WORD  0             ;DATA FIELD LENGTH
5756 066156 000000      .WORD  0             ;RBPGR
5757 066160 000000      .WORD  0             ;XST0
5758 066162 000000      .WORD  0             ;XST1
5759 066164 000000      .WORD  0             ;XST2
5760 066166 000000      .WORD  0             ;XST3
5761 066170 000000      .WORD  0             ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
5762 066172      T20BFSTA: .BLKB 64.         ;READ STATUS AND WRITE FIFO BUFFER
5763 066272      T20BEND:      ;END OF MESSAGE BUFFER
5764
5765      ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
5766      ;
5768 066272          .BLKB  10-<.-TSV2&7>
5770 066300      T20PK2:      ;WRITE SUBSYSTEM WITH ACK
5771 066300 100006      .WORD  P.WRTSUB!P.ACK ;LOW ADDRESS OF DATA BLOCK
5772 066302 066310'    .WORD  T20DT2        ;HIGH ADDRESS OF DATA BLOCK
5773 066304 000000      .WORD  0
5774 066306 000012      .WORD  10.         ;MINIMUM MESSAGE PACKET SIZE
5775
5776 066310      T20DT2:      ;DATA BLOCK
5777 066310 000        .BYTE  0             ;BSELO
5778 066311 000        .BYTE  0             ;BSEL1
5779 066312 000000      .WORD  0             ;SEL2
5780 066314          .BLKB  64.         ;WRITE FIFO DATA OUTPUT BUFFER
5781
5782
5783 066414          .BLKB  104401
5784
5785      .SBTTL  TEST 10: MANUAL INTERVENTION
5786
5787      ;THE MANUAL INTERVENTION TEST IS A STANDALONE ROUTINE (NOT REALLY A "TEST")

```

L10073: TRAP C\$ETST

```

5788      ; THAT ALLOWS THE OPERATOR TO CHECK OUT VARIOUS ELEMENTS AND FUNCTIONS OF
5789      ; THE SUBSYSTEM THAT CANNOT BE MANIPULATED BY THE PROGRAM ALONE. WHEN
5790      ; THIS ROUTINE IS STARTED, IT FIRST PRINTS OUT A MENU OF SELECTABLE
5791      ; SUBTESTS AND THEN WAITS FOR THE OPERATOR TO TYPE IN A SELECTION CODE.
5792      ; THE ONLY WAYS TO EXIT THIS ROUTINE AND RETURN TO THE DIAGNOSTIC SUPERVISOR
5793      ; ARE BY TYPING <CTRL-C> OR SELECTING CODE 7.
5794      ; SELECTION CODES AND SUBROUTINES ARE:
5795      ;
5796      ;
5797      ;      CODE      ROUTINE
5798      ;
5799      ;      0      HELP. PRINTS THIS MENU.
5800      ;      1      TURN ON ALL M7455 LED INDICATORS
5801      ;      2      TURN OFF ALL M7455 LED INDICATORS
5802      ;      3      OFFLINE/ONLINE ATTENTION TEST
5803      ;      4      WRITE-PROTECT TEST
5804      ;      5      INITIATE TRANSPORT SERVO EXERCISER
5805      ;      6      PRINT EXTENDED TRANSPORT STATUS
5806      ;      7      EXIT (RETURN TO SUPERVISOR)
5807      ;
5808      ;
5809      ; EACH MENU ITEM CORRESPONDS TO A SUBTEST, AS FOLLOWS:
5810      ;
5811      ;
5812      ;
5813      ; PRINTS OUT THE MENU ON THE CONSOLE TERMINAL.
5814      ;
5815      ;
5816      ; CAUSES ALL THREE LED INDICATORS ON THE M7455 MODULE
5817      ; TO BE ILLUMINATED. AFTER INITIATING THIS ROUTINE, THE OPERATOR
5818      ; SHOULD OBSERVE THE LED'S AND VERIFY THAT THEY ARE INDEED ALL LIT.
5819      ; THIS ROUTINE FIRST USES THE WRITE SUBSYSTEM MEMORY COMMAND TO
5820      ; SET THE FORCE WRONG PARITY FLIP-FLOP, WHICH SERVES TO DRIVE THE
5821      ; "PROCESSOR NOT OK" LED. THEN IT ENTERS A LOOP THAT CONTINUALLY
5822      ; WRITES THE LOW BYTE OF ISDB AND READS THE TSSR. THESE LATTER TWO
5823      ; OPERATIONS WILL CAUSE THE "NOT SSR" AND "DRIVING BUS" LED'S TO
5824      ; GLOW -- THEY ARE NOT REALLY LIT AT ALL TIMES BUT SHOULD APPEAR
5825      ; REASONABLY VISIBLE.
5826      ;
5827      ;
5828      ; INITIALIZES THE CONTROLLER TO CAUSE ALL LED'S TO
5829      ; EXTINGUISH.
5830      ;
5831      ;
5832      ;
5833      ;
5834      ; THIS ROUTINE INITIALIZES THE CONTROLLER, ISSUES A
5835      ; WRITE CHARACTERISTICS COMMAND TO ENABLE ATTENTION INTERRUPTS,
5836      ; ISSUES A MESSAGE BUFFER RELEASE COMMAND, PRINTS A MESSAGE ON THE
5837      ; CONSOLE TERMINAL INSTRUCTING THE OPERATOR TO TOGGLE THE ON-LINE
5838      ; SWITCH ON THE TRANSPORT, THEN WAITS FOR AN ATTENTION INTERRUPT.
5839      ; EACH TIME THE TRANSPORT TRANSITIONS FROM ON-LINE TO OFF-LINE OR
5840      ; VICE-VERSA, AN ATTENTION INTERRUPT SHOULD BE GENERATED. THE PROGRAM
5841      ; WILL REPORT THE INTERRUPT AND THE CURRENT STATE OF THE TRANSPORT.
5842      ; THE OPERATOR SHOULD VERIFY THAT THE REPORTED STATE MATCHES THE
5843      ; STATE INDICATED BY THE LED ON THE FRONT PANEL OF THE TRANSPORT.
5844      ; IN ADDITION, WHEN THE TRANSPORT IS PLACED OFF-LINE, THE PROGRAM

```

```

5845      ISSUES A SEQUENCE OF TAPE-MOTION COMMANDS (READ, WRITE, POSITION, ETC.
5846      AND VERIFIES THAT, FOR EACH COMMAND, FUNCTION REJECT TERMINATION
5847      RESULTS, ALONG WITH THE NON-EXECUTABLE FUNCTION (NEF) ERROR BIT BEING
5848      SET.
5849
5850      THIS ROUTINE INSTRUCTS THE OPERATOR TO MOUNT A SCRATCH
5851      TAPE REEL THAT DOES NOT HAVE A WRITE-ENABLE RING INSTALLED, THEN
5852      WAITS FOR THE OPERATOR TO RESPOND THAT THIS HAS BEEN ACCOMPLISHED.
5853      UPON THE RESPONSE, THE PROGRAM VERIFIES THAT THE TRANSPORT SHOWS
5854      A WRITE-PROTECTED STATUS, THEN ATTEMPTS TO WRITE DATA ON THE
5855      TAPE AND EXPECTS THE APPROPRIATE ERROR TERMINATION INDICATING THAT
5856      THE WRITE FUNCTION COULD NOT BE PERFORMED BECAUSE THE REEL IS
5857      WRITE-PROTECTED. IF THE APPROPRIATE TERMINATION IS NOT RECEIVED,
5858      AN ERROR IS REPORTED.
5859
5860
5861
5862      INSTRUCTS THE OPERATOR TO PLACE THE TAPE TRANSPORT(S)
5863      ON-LINE (IF ANY ARE OFF-LINE) THEN ATTEMPTS TO PERFORM AN EXTENDED
5864      STATUS READOUT. FOR EACH TRANSPORT EQUIPPED WITH THIS FEATURE,
5865      THE PROGRAM FORMATS AND PRINTS OUT THE RESULTING STATUS. IF THE
5866      TRANSPORT IS NOT EQUIPPED WITH THIS FEATURE, A MESSAGE INDICATING
5867      SUCH IS ISSUED.
5868
5869
5870
5871
5872      BGNTST
5873      RFLAGS  R0
5874      GET OPERATOR FLAGS
5875      TRAP     C$RFLA
5876      BR, IF OK TO RUN
5877      "TEST NOT EXECUTED"
5878      JUMP IF NOT FIRST TEST
5879
5880      21$:
5881      MOV     #T38ID,R0
5882      JSR     PC,TSTSETUP
5883      JSR     PC,CHKMAN
5884      BCS     22$
5885      JMP     64$
5886
5887      22$:
5888      CLR     FATFLG
5889      MOV     #65000.,T38DLY
5890      JSR     PC,SOFINIT
5891      BCS     23$
5892      MOV     R0,R1
5893      BIT     #SSR,R1
5894      BNE     23$
5895      DELAY   250
5896
5897      23$:
5898      CLR     FATFLG
5899      MOV     #65000.,T38DLY
5900      JSR     PC,SOFINIT
5901      BCS     23$
5902      MOV     R0,R1
5903      BIT     #SSR,R1
5904      BNE     23$
5905      DELAY   250
5906
5907      24$:
5908      CLR     FATFLG
5909      MOV     #65000.,T38DLY
5910      JSR     PC,SOFINIT
5911      BCS     23$
5912      MOV     R0,R1
5913      BIT     #SSR,R1
5914      BNE     23$
5915      DELAY   250
5916
5917      25$:
5918      CLR     FATFLG
5919      MOV     #65000.,T38DLY
5920      JSR     PC,SOFINIT
5921      BCS     23$
5922      MOV     R0,R1
5923      BIT     #SSR,R1
5924      BNE     23$
5925      DELAY   250
5926
5927      26$:
5928      CLR     FATFLG
5929      MOV     #65000.,T38DLY
5930      JSR     PC,SOFINIT
5931      BCS     23$
5932      MOV     R0,R1
5933      BIT     #SSR,R1
5934      BNE     23$
5935      DELAY   250
5936
5937      27$:
5938      CLR     FATFLG
5939      MOV     #65000.,T38DLY
5940      JSR     PC,SOFINIT
5941      BCS     23$
5942      MOV     R0,R1
5943      BIT     #SSR,R1
5944      BNE     23$
5945      DELAY   250
5946
5947      28$:
5948      CLR     FATFLG
5949      MOV     #65000.,T38DLY
5950      JSR     PC,SOFINIT
5951      BCS     23$
5952      MOV     R0,R1
5953      BIT     #SSR,R1
5954      BNE     23$
5955      DELAY   250
5956
5957      29$:
5958      CLR     FATFLG
5959      MOV     #65000.,T38DLY
5960      JSR     PC,SOFINIT
5961      BCS     23$
5962      MOV     R0,R1
5963      BIT     #SSR,R1
5964      BNE     23$
5965      DELAY   250
5966
5967      30$:
5968      CLR     FATFLG
5969      MOV     #65000.,T38DLY
5970      JSR     PC,SOFINIT
5971      BCS     23$
5972      MOV     R0,R1
5973      BIT     #SSR,R1
5974      BNE     23$
5975      DELAY   250
5976
5977      31$:
5978      CLR     FATFLG
5979      MOV     #65000.,T38DLY
5980      JSR     PC,SOFINIT
5981      BCS     23$
5982      MOV     R0,R1
5983      BIT     #SSR,R1
5984      BNE     23$
5985      DELAY   250
5986
5987      32$:
5988      CLR     FATFLG
5989      MOV     #65000.,T38DLY
5990      JSR     PC,SOFINIT
5991      BCS     23$
5992      MOV     R0,R1
5993      BIT     #SSR,R1
5994      BNE     23$
5995      DELAY   250
5996
5997      33$:
5998      CLR     FATFLG
5999      MOV     #65000.,T38DLY
6000      JSR     PC,SOFINIT
6001      BCS     23$
6002      MOV     R0,R1
6003      BIT     #SSR,R1
6004      BNE     23$
6005      DELAY   250
6006
6007      34$:
6008      CLR     FATFLG
6009      MOV     #65000.,T38DLY
6010      JSR     PC,SOFINIT
6011      BCS     23$
6012      MOV     R0,R1
6013      BIT     #SSR,R1
6014      BNE     23$
6015      DELAY   250
6016
6017      35$:
6018      CLR     FATFLG
6019      MOV     #65000.,T38DLY
6020      JSR     PC,SOFINIT
6021      BCS     23$
6022      MOV     R0,R1
6023      BIT     #SSR,R1
6024      BNE     23$
6025      DELAY   250
6026
6027      36$:
6028      CLR     FATFLG
6029      MOV     #65000.,T38DLY
6030      JSR     PC,SOFINIT
6031      BCS     23$
6032      MOV     R0,R1
6033      BIT     #SSR,R1
6034      BNE     23$
6035      DELAY   250
6036
6037      37$:
6038      CLR     FATFLG
6039      MOV     #65000.,T38DLY
6040      JSR     PC,SOFINIT
6041      BCS     23$
6042      MOV     R0,R1
6043      BIT     #SSR,R1
6044      BNE     23$
6045      DELAY   250
6046
6047      38$:
6048      CLR     FATFLG
6049      MOV     #65000.,T38DLY
6050      JSR     PC,SOFINIT
6051      BCS     23$
6052      MOV     R0,R1
6053      BIT     #SSR,R1
6054      BNE     23$
6055      DELAY   250
6056
6057      39$:
6058      CLR     FATFLG
6059      MOV     #65000.,T38DLY
6060      JSR     PC,SOFINIT
6061      BCS     23$
6062      MOV     R0,R1
6063      BIT     #SSR,R1
6064      BNE     23$
6065      DELAY   250
6066
6067      40$:
6068      CLR     FATFLG
6069      MOV     #65000.,T38DLY
6070      JSR     PC,SOFINIT
6071      BCS     23$
6072      MOV     R0,R1
6073      BIT     #SSR,R1
6074      BNE     23$
6075      DELAY   250
6076
6077      41$:
6078      CLR     FATFLG
6079      MOV     #65000.,T38DLY
6080      JSR     PC,SOFINIT
6081      BCS     23$
6082      MOV     R0,R1
6083      BIT     #SSR,R1
6084      BNE     23$
6085      DELAY   250
6086
6087      42$:
6088      CLR     FATFLG
6089      MOV     #65000.,T38DLY
6090      JSR     PC,SOFINIT
6091      BCS     23$
6092      MOV     R0,R1
6093      BIT     #SSR,R1
6094      BNE     23$
6095      DELAY   250
6096
6097      43$:
6098      CLR     FATFLG
6099      MOV     #65000.,T38DLY
6100      JSR     PC,SOFINIT
6101      BCS     23$
6102      MOV     R0,R1
6103      BIT     #SSR,R1
6104      BNE     23$
6105      DELAY   250
6106
6107      44$:
6108      CLR     FATFLG
6109      MOV     #65000.,T38DLY
6110      JSR     PC,SOFINIT
6111      BCS     23$
6112      MOV     R0,R1
6113      BIT     #SSR,R1
6114      BNE     23$
6115      DELAY   250
6116
6117      45$:
6118      CLR     FATFLG
6119      MOV     #65000.,T38DLY
6120      JSR     PC,SOFINIT
6121      BCS     23$
6122      MOV     R0,R1
6123      BIT     #SSR,R1
6124      BNE     23$
6125      DELAY   250
6126
6127      46$:
6128      CLR     FATFLG
6129      MOV     #65000.,T38DLY
6130      JSR     PC,SOFINIT
6131      BCS     23$
6132      MOV     R0,R1
6133      BIT     #SSR,R1
6134      BNE     23$
6135      DELAY   250
6136
6137      47$:
6138      CLR     FATFLG
6139      MOV     #65000.,T38DLY
6140      JSR     PC,SOFINIT
6141      BCS     23$
6142      MOV     R0,R1
6143      BIT     #SSR,R1
6144      BNE     23$
6145      DELAY   250
6146
6147      48$:
6148      CLR     FATFLG
6149      MOV     #65000.,T38DLY
6150      JSR     PC,SOFINIT
6151      BCS     23$
6152      MOV     R0,R1
6153      BIT     #SSR,R1
6154      BNE     23$
6155      DELAY   250
6156
6157      49$:
6158      CLR     FATFLG
6159      MOV     #65000.,T38DLY
6160      JSR     PC,SOFINIT
6161      BCS     23$
6162      MOV     R0,R1
6163      BIT     #SSR,R1
6164      BNE     23$
6165      DELAY   250
6166
6167      50$:
6168      CLR     FATFLG
6169      MOV     #65000.,T38DLY
6170      JSR     PC,SOFINIT
6171      BCS     23$
6172      MOV     R0,R1
6173      BIT     #SSR,R1
6174      BNE     23$
6175      DELAY   250
6176
6177      51$:
6178      CLR     FATFLG
6179      MOV     #65000.,T38DLY
6180      JSR     PC,SOFINIT
6181      BCS     23$
6182      MOV     R0,R1
6183      BIT     #SSR,R1
6184      BNE     23$
6185      DELAY   250
6186
6187      52$:
6188      CLR     FATFLG
6189      MOV     #65000.,T38DLY
6190      JSR     PC,SOFINIT
6191      BCS     23$
6192      MOV     R0,R1
6193      BIT     #SSR,R1
6194      BNE     23$
6195      DELAY   250
6196
6197      53$:
6198      CLR     FATFLG
6199      MOV     #65000.,T38DLY
6200      JSR     PC,SOFINIT
6201      BCS     23$
6202      MOV     R0,R1
6203      BIT     #SSR,R1
6204      BNE     23$
6205      DELAY   250
6206
6207      54$:
6208      CLR     FATFLG
6209      MOV     #65000.,T38DLY
6210      JSR     PC,SOFINIT
6211      BCS     23$
6212      MOV     R0,R1
6213      BIT     #SSR,R1
6214      BNE     23$
6215      DELAY   250
6216
6217      55$:
6218      CLR     FATFLG
6219      MOV     #65000.,T38DLY
6220      JSR     PC,SOFINIT
6221      BCS     23$
6222      MOV     R0,R1
6223      BIT     #SSR,R1
6224      BNE     23$
6225      DELAY   250
6226
6227      56$:
6228      CLR     FATFLG
6229      MOV     #65000.,T38DLY
6230      JSR     PC,SOFINIT
6231      BCS     23$
6232      MOV     R0,R1
6233      BIT     #SSR,R1
6234      BNE     23$
6235      DELAY   250
6236
6237      57$:
6238      CLR     FATFLG
6239      MOV     #65000.,T38DLY
6240      JSR     PC,SOFINIT
6241      BCS     23$
6242      MOV     R0,R1
6243      BIT     #SSR,R1
6244      BNE     23$
6245      DELAY   250
6246
6247      58$:
6248      CLR     FATFLG
6249      MOV     #65000.,T38DLY
6250      JSR     PC,SOFINIT
6251      BCS     23$
6252      MOV     R0,R1
6253      BIT     #SSR,R1
6254      BNE     23$
6255      DELAY   250
6256
6257      59$:
6258      CLR     FATFLG
6259      MOV     #65000.,T38DLY
6260      JSR     PC,SOFINIT
6261      BCS     23$
6262      MOV     R0,R1
6263      BIT     #SSR,R1
6264      BNE     23$
6265      DELAY   250
6266
6267      60$:
6268      CLR     FATFLG
6269      MOV     #65000.,T38DLY
6270      JSR     PC,SOFINIT
6271      BCS     23$
6272      MOV     R0,R1
6273      BIT     #SSR,R1
6274      BNE     23$
6275      DELAY   250
6276
6277      61$:
6278      CLR     FATFLG
6279      MOV     #65000.,T38DLY
6280      JSR     PC,SOFINIT
6281      BCS     23$
6282      MOV     R0,R1
6283      BIT     #SSR,R1
6284      BNE     23$
6285      DELAY   250
6286
6287      62$:
6288      CLR     FATFLG
6289      MOV     #65000.,T38DLY
6290      JSR     PC,SOFINIT
6291      BCS     23$
6292      MOV     R0,R1
6293      BIT     #SSR,R1
6294      BNE     23$
6295      DELAY   250
6296
6297      63$:
6298      CLR     FATFLG
6299      MOV     #65000.,T38DLY
6300      JSR     PC,SOFINIT
6301      BCS     23$
6302      MOV     R0,R1
6303      BIT     #SSR,R1
6304      BNE     23$
6305      DELAY   250
6306
6307      64$:
6308      CLR     FATFLG
6309      MOV     #65000.,T38DLY
6310      JSR     PC,SOFINIT
6311      BCS     23$
6312      MOV     R0,R1
6313      BIT     #SSR,R1
6314      BNE     23$
6315      DELAY   250
6316
6317      65$:
6318      CLR     FATFLG
6319      MOV     #65000.,T38DLY
6320      JSR     PC,SOFINIT
6321      BCS     23$
6322      MOV     R0,R1
6323      BIT     #SSR,R1
6324      BNE     23$
6325      DELAY   250
6326
6327      66$:
6328      CLR     FATFLG
6329      MOV     #65000.,T38DLY
6330      JSR     PC,SOFINIT
6331      BCS     23$
6332      MOV     R0,R1
6333      BIT     #SSR,R1
6334      BNE     23$
6335      DELAY   250
6336
6337      67$:
6338      CLR     FATFLG
6339      MOV     #65000.,T38DLY
6340      JSR     PC,SOFINIT
6341      BCS     23$
6342      MOV     R0,R1
6343      BIT     #SSR,R1
6344      BNE     23$
6345      DELAY   250
6346
6347      68$:
6348      CLR     FATFLG
6349      MOV     #65000.,T38DLY
6350      JSR     PC,SOFINIT
6351      BCS     23$
6352      MOV     R0,R1
6353      BIT     #SSR,R1
6354      BNE     23$
6355      DELAY   250
6356
6357      69$:
6358      CLR     FATFLG
6359      MOV     #65000.,T38DLY
6360      JSR     PC,SOFINIT
6361      BCS     23$
6362      MOV     R0,R1
6363      BIT     #SSR,R1
6364      BNE     23$
6365      DELAY   250
6366
6367      70$:
6368      CLR     FATFLG
6369      MOV     #65000.,T38DLY
6370      JSR     PC,SOFINIT
6371      BCS     23$
6372      MOV     R0,R1
6373      BIT     #SSR,R1
6374      BNE     23$
6375      DELAY   250
6376
6377      71$:
6378      CLR     FATFLG
6379      MOV     #65000.,T38DLY
6380      JSR     PC,SOFINIT
6381      BCS     23$
6382      MOV     R0,R1
6383      BIT     #SSR,R1
6384      BNE     23$
6385      DELAY   250
6386
6387      72$:
6388      CLR     FATFLG
6389      MOV     #65000.,T38DLY
6390      JSR     PC,SOFINIT
6391      BCS     23$
6392      MOV     R0,R1
6393      BIT     #SSR,R1
6394      BNE     23$
6395      DELAY   250
6396
6397      73$:
6398      CLR     FATFLG
6399      MOV     #65000.,T38DLY
6400      JSR     PC,SOFINIT
6401      BCS     23$
6402      MOV     R0,R1
6403      BIT     #SSR,R1
6404      BNE     23$
6405      DELAY   250
6406
6407      74$:
6408      CLR     FATFLG
6409      MOV     #65000.,T38DLY
6410      JSR     PC,SOFINIT
6411      BCS     23$
6412      MOV     R0,R1
6413      BIT     #SSR,R1
6414      BNE     23$
6415      DELAY   250
6416
6417      75$:
6418      CLR     FATFLG
6419      MOV     #65000.,T38DLY
6420      JSR     PC,SOFINIT
6421      BCS     23$
6422      MOV     R0,R1
6423      BIT     #SSR,R1
6424      BNE     23$
6425      DELAY   250
6426
6427      76$:
6428      CLR     FATFLG
6429      MOV     #65000.,T38DLY
6430      JSR     PC,SOFINIT
6431      BCS     23$
6432      MOV     R0,R1
6433      BIT     #SSR,R1
6434      BNE     23$
6435      DELAY   250
6436
6437      77$:
6438      CLR     FATFLG
6439      MOV     #65000.,T38DLY
6440      JSR     PC,SOFINIT
6441      BCS     23$
6442      MOV     R0,R1
6443      BIT     #SSR,R1
6444      BNE     23$
6445      DELAY   250
6446
6447      78$:
6448      CLR     FATFLG
6449      MOV     #65000.,T38DLY
6450      JSR     PC,SOFINIT
6451      BCS     23$
6452      MOV     R0,R1
6453      BIT     #SSR,R1
6454      BNE     23$
6455      DELAY   250
6456
6457      79$:
6458      CLR     FATFLG
6459      MOV     #65000.,T38DLY
6460      JSR     PC,SOFINIT
6461      BCS     23$
6462      MOV     R0,R1
6463      BIT     #SSR,R1
6464      BNE     23$
6465      DELAY   250
6466
6467      80$:
6468      CLR     FATFLG
6469      MOV     #65000.,T38DLY
6470      JSR     PC,SOFINIT
6471      BCS     23$
6472      MOV     R0,R1
6473      BIT     #SSR,R1
6474      BNE     23$
6475      DELAY   250
6476
6477      81$:
6478      CLR     FATFLG
6479      MOV     #65000.,T38DLY
6480      JSR     PC,SOFINIT
6481      BCS     23$
6482      MOV     R0,R1
6483      BIT     #SSR,R1
6484      BNE     23$
6485      DELAY   250
6486
6487      82$:
6488      CLR     FATFLG
6489      MOV     #65000.,T38DLY
6490      JSR     PC,SOFINIT
6491      BCS     23$
6492      MOV     R0,R1
6493      BIT     #SSR,R1
6494      BNE     23$
6495      DELAY   250
6496
6497      83$:
6498      CLR     FATFLG
6499      MOV     #65000.,T38DLY
6500      JSR     PC,SOFINIT
6501      BCS     23$
6502      MOV     R0,R1
6503      BIT     #SSR,R1
6504      BNE     23$
6505      DELAY   250
6506
6507      84$:
6508      CLR     FATFLG
6509      MOV     #65000.,T38DLY
6510      JSR     PC,SOFINIT
6511      BCS     23$
6512      MOV     R0,R1
6513      BIT     #SSR,R1
6514      BNE     23$
6515      DELAY   250
6516
6517      85$:
6518      CLR     FATFLG
6519      MOV     #65000.,T38DLY
6520      JSR     PC,SOFINIT
6521      BCS     23$
6522      MOV     R0,R1
6523      BIT     #SSR,R1
6524      BNE     23$
6525      DELAY   250
6526
6527      86$:
6528      CLR     FATFLG
6529      MOV     #65000.,T38DLY
6530      JSR     PC,SOFINIT
6531      BCS     23$
6532      MOV     R0,R1
6533      BIT     #SSR,R1
6534      BNE     23$
6535      DELAY   250
6536
6537      87$:
6538      CLR     FATFLG
6539      MOV     #65000.,T38DLY
6540      JSR     PC,SOFINIT
6541      BCS     23$
6542      MOV     R0,R1
6543      BIT     #SSR,R1
6544      BNE     23$
6545      DELAY   250
6546
6547      88$:
6548      CLR     FATFLG
6549      MOV     #65000.,T38DLY
6550      JSR     PC,SOFINIT
6551      BCS     23$
6552      MOV     R0,R1
6553      BIT     #SSR,R1
6554      BNE     23$
6555      DELAY   250
6556
6557      89$:
6558      CLR     FATFLG
6559      MOV     #65000.,T38DLY
6560      JSR     PC,SOFINIT
6561      BCS     23$
6562      MOV     R0,R1
6563      BIT     #SSR,R1
6564      BNE     23$
6565      DELAY   250
6566
6567      90$:
6568      CLR     FATFLG
6569      MOV     #65000.,T38DLY
6570      JSR     PC,SOFINIT
6571      BCS     23$
6572      MOV     R0,R1
6573      BIT     #SSR,R1
6574      BNE     23$
6575      DELAY   250
6576
6577      91$:
6578      CLR     FATFLG
6579      MOV     #65000.,T38DLY
6580      JSR     PC,SOFINIT
6581      BCS     23$
6582      MOV     R0,R1
6583      BIT     #SSR,R1
6584      BNE     23$
6585      DELAY   250
6586
6587      92$:
6588      CLR     FATFLG
6589      MOV     #65000.,T38DLY
6590      JSR     PC,SOFINIT
6591      BCS     23$
6592      MOV     R0,R1
6593      BIT     #SSR,R1
6594      BNE     23$
6595      DELAY   250
6596
6597      93$:
6598      CLR     FATFLG
6599      MOV     #65000.,T38DLY
6600      JSR     PC,SOFINIT
6601      BCS     23$
6602      MOV     R0,R1
6603      BIT     #SSR,R1
6604      BNE     23$
6605      DELAY   250
6606
6607      94$:
6608      CLR     FATFLG
6609      MOV     #65000.,T38DLY
6610      JSR     PC,SOFINIT
6611      BCS     23$
6612      MOV     R0,R1
6613      BIT     #SSR,R1
6614      BNE     23$
6615      DELAY   250
6616
6617      95$:
6618      CLR     FATFLG
6619      MOV     #65000.,T38DLY
6620      JSR     PC,SOFINIT
6621      BCS     23$
6622      MOV     R0,R1
6623      BIT     #SSR,R1
6624      BNE     23$
6625      DELAY   250
6626
6627      96$:
6628      CLR     FATFLG
6629      MOV     #65000.,T38DLY
6630      JSR     PC,SOFINIT
6631      BCS     23$
6632      MOV     R0,R1
6633      BIT     #SSR,R1
6634      BNE     23$
6635      DELAY   250
6636
6637      97$:
6638      CLR     FATFLG
6639      MOV     #65000.,T38DLY
6640      JSR     PC,SOFINIT
6641      BCS     23$
6642      MOV     R0,R1
6643      BIT     #SSR,R1
6644      BNE     23$
6645      DELAY   250
6646
6647      98$:
6648      CLR     FATFLG
6649      MOV     #65000.,T38DLY
6650      JSR     PC,SOFINIT
6651      BCS     23$
6652      MOV     R0,R1
6653      BIT     #SSR,R1
6654      BNE     23$
6655      DELAY   250
6656
6657      99$:
6658      CLR     FATFLG
6659      MOV     #65000.,T38DLY
6660      JSR     PC,SOFINIT
6661      BCS     23$
6662      MOV     R0,R1
6663      BIT     #SSR,R1
6664      BNE     23$
6665      DELAY   250
6666
6667      100$:
6668      CLR     FATFLG
6669      MOV     #65000.,T38DLY
6670      JSR     PC,SOFINIT
6671      BCS     23$
6672      MOV     R0,R1
6673      BIT     #SSR,R1
6674      BNE     23$
6675      DELAY   250
6676
6677      101$:
6678      CLR     FATFLG
6679      MOV     #65000.,T38DLY
6680      JSR     PC,SOFINIT
6681      BCS     23$
6682      MOV     R0,R1
6683      BIT     #SSR,R1
6684      BNE     23$
6685      DELAY   250
6686
6687      102$:
6688      CLR     FATFLG
6689      MOV     #65000.,T38DLY
6690      JSR     PC,SOFINIT
6691      BCS     23$
6692      MOV     R0,R1
6693      BIT     #SSR,R1
6694      BNE     23$
6695      DELAY   250
6696
6697      103$:
6698      CLR     FATFLG
6699      MOV     #65000.,T38DLY
6700      JSR     PC,SOFINIT
6701      BCS     23$
6702      MOV     R0,R1
6703      BIT     #SSR,R1
6704      BNE     23$
6705      DELAY   250
6706
6707      104$:
6708      CLR     FATFLG
6709      MOV     #65000.,T38DLY
6710      JSR     PC,SOFINIT
6711      BCS     23$
6712      MOV     R0,R1
6713      BIT     #SSR,R1
6714      BNE     23$
6715      DELAY   250
6716
6717      105$:
6718      CLR     FATFLG
6719      MOV     #65000.,T38DLY
6720      JSR     PC,SOFINIT
6721      BCS     23$
6722      MOV     R0,R1
6723      BIT     #SSR,R1
6724      BNE     23$
6725      DELAY   250
6726
6727      106$:
6728      CLR     FATFLG
6729      MOV     #65000.,T38DLY
6730      JSR     PC,SOFINIT
6731      BCS     23$
6732      MOV     R0,R1
6733      BIT     #SSR,R1
6734      BNE     23$
6735      DELAY   250
6736
6737      107$:
6738      CLR     FATFLG
6739      MOV     #65000.,T38DLY
6740      JSR     PC,SOFINIT
6741      BCS     23$
6742      MOV     R0,R1
6743      BIT     #SSR,R1
6744      BNE     23$
6745      DELAY   250
6746
6747      108$:
6748      CLR     FATFLG
6749      MOV     #65000.,T38DLY
6750      JSR     PC,SOFINIT
6751      BCS     23$
6752      MOV     R0,R1
6753      BIT     #SSR,R1
6754      BNE     23$
6755      DELAY   250
6756
6757      109$:
6758      CLR     FATFLG
6759      MOV     #65000.,T38DLY
6760      JSR     PC,SOFINIT
6761      BCS     23$
6762      MOV     R0,R1
6763      BIT     #SSR,R1
6764      BNE     23$
6765      DELAY   250
6766
6767      110$:
6768      CLR     FATFLG
6769      MOV     #65000.,T38DLY
6770      JSR     PC,SOFINIT
6771      BCS     23$
6772      MOV     R0,R1
6773      BIT     #SSR,R1
6774      BNE     23$
6775      DELAY   250
6776
6777      111$:
6778      CLR     FATFLG
6779      MOV     #65000.,T38DLY
6780      JSR     PC,SOFINIT
6781      BCS     23$
6782      MOV     R0,R1
6783      BIT     #SSR,R1
6784      BNE     23$
6785      DELAY   250
6786
6787      112$:
6788      CLR     FATFLG
6789      MOV     #65000.,T38DLY
6790      JSR     PC,SOFINIT
6791      BCS     23$
6792      MOV     R0,R1
6793      BIT     #SSR,R1
6794      BNE     23$
6795      DELAY   250
6796
6797      113$:
6798      CLR     FATFLG
6799      MOV     #65000.,T38DLY
6800      JSR     PC,SOFINIT
6801      BCS     23$
6802      MOV     R0,R1
6803      BIT     #SSR,R1
6804      BNE     23$
6805      DELAY   250
6806
6807      114$:
6808      CLR     FATFLG
6809      MOV     #65000.,T38DLY
6810      JSR     PC,SOFINIT
6811      BCS     23$
6812      MOV     R0,R1
6813      BIT     #SSR,R1
6814      BNE     23$
6815      DELAY   250
6816
6817      115$:
6818      CLR     FATFLG
6819      MOV     #65000.,T38DLY
6820      JSR     PC,SOFINIT
6821      BCS     23$
6822      MOV     R0,R1
6823      BIT     #SSR,R1
6824      BNE     23$
6825      DELAY   250
6826
6827      116$:
6828      CLR     FATFLG
6829      MOV     #65000.,T3
```

E1

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 10: MANUAL INTERVENTION

SEQ 211

```

5899 066532 005337 071212'      DEC      T38DLY      ;BUMP COUNTER DOWN
5900 066536 001352              BNE      5$          ;BR, IF MORE TIME LEFT
5901 066540              ERRDF   ERRNO,SFIERR,SFIMSG ;REPORT FATAL ERROR
      066540 104455              TRAP      C$ERDF
      066542 001751              .WORD    1001
      066544 003646'            .WORD    SFIERR
      066546 011644'            .WORD    SFIMSG
5902 066550 012700 073142'      23$:  MOV     @MIMENU,R0      ;MENU OF MANUAL INTERVENTIONS
5903 066554 012701 000006      MOV     @6,R1          ;MAXIMUM ALLOWED SELECTION
5904 066560 004737 020076'      JSR     PC,GETSEL      ;GO GET THE OPERATORS SELECTION
5905 066564 010004              MOV     R0,R4          ;GET NUMBER FROM ROUTINE
5906 066566 006304              ASL     R4             ;CONVERT TO WORD OFFSET
5907 066570 000174 066574'      JMP     @6$(R4)        ;JUMP TO PROPER LOOP
5908 066574 066452'            6$:  .WORD    2$          ;RETYPE THE MENU
5909 066576 066612'            .WORD    10$         ; 1 TURN ON LED'S
5910 066600 067074'            .WORD    15$         ; 2 TURN OFF LED'S
5911 066602 067326'            .WORD    20$         ; 3 ONLINE ATTENTION
5912 066604 067762'            .WORD    25$         ; 4 WRITE PROTECT
5913 066606 070716'            .WORD    35$         ; 5 EXTENDED TRANSPORT STATUS
5914 066610 071174'            .WORD    63$         ; 6 LEAVE THE TEST
5915 066612              10$:  PRINTF   @T38MS2      ;TELL OPERATOR TO CNTRL-C FOR EXIT
      066612 012746 073011'      MOV     @T38MS2,-(SP)
      066616 012746 000001      MOV     @1,-(SP)
      066622 010600              MOV     SP,R0
      066624 104417              TRAP      C$PNTF
      066626 062706 000004      ADD     @4,SP
5916 066632 004737 073546'      JSR     PC,T38PEST      ;SET PACKET TO INITIAL VALUES
5917 066636 004737 015604'      JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
5918 066642 103405              BCS     100$         ;BR IF SOFT INIT = OK
5922 066644 010001              MOV     R0,R1          ;SAVE CONTENTS OF TSSR
5923 066646              ERRDF   ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
      066646 104455              TRAP      C$ERDF
      066650 001752              .WORD    1002
      066652 003646'            .WORD    SFIERR
      066654 011644'            .WORD    SFIMSG
5924 066656 013737 002202' 071740' 100$: MOV     UNITN,T38D$W      ;SET UNIT NUMBER
5925
5926 066664 012704 071720'      MOV     @T38PK2,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
5927 066670 004737 010472'      JSR     PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
5928 066674 103405              BCS     110$         ;BR, IF COMMAND ISSUED OK
5932 066676 010001              MOV     R0,R1          ;SAVE CONTENTS OF TSSR
5933 066700              ERRHARD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      066700 104456              TRAP      C$ERHARD
      066702 001753              .WORD    1003
      066704 005052'            .WORD    WRTMSG
      066706 011644'            .WORD    SFIMSG
5934 066710              110$: MOV     @0,T38BS1      ;CLEAR BIT #4
5935 066710 112737 000000 071231' MOV     @11,T38BS0      ;WRITE MISC COMMAND
5936 066716 112737 000011 071230' MOV     @T38PACKET,R4      ;SET UP NEW WRT. SUBSYS MEM. COMMAND
5937 066724 012704 071220'
5938
5939      ;NOTE: THIS COMMAND TURNS ON THE PROCESSOR FAIL LED
5940
5941 066730 010465 000000      MOV     R4,TSD8(R5)      ;SET THE PACKET ADDRESS
5942 066734 004737 016146'      JSR     PC,CHKTSSR      ;WAIT FOR SSH TO SET
5943 066740 103405              BCS     150$         ;BR IF CARRY SET (GOOD RETURN)
5944 066742 010001              MOV     R0,R1          ;SAVE CONTENTS OF TSSR

```

```

5989
5990
5991
5992
5993
5994 067166
5995 067166 112737 000000 071231'
5996 067174 112737 000025 071230'
5997 067202 012704 071220'
5998 067206 010465 000000
5999 067212 004737 016146'
6000 067216 103405
6001 067220 010001
6005 067222
      067222 104455
      067224 001757
      067226 072416'
      067230 011656'
6006 067232
      067232 104406
6007 067234
      067234 012700 000340
      067240 104441
6008 067242 005037 071204'
6009 067246 032737 000100 177560
6010 067254 001005
6011 067256 005237 071204'
6012 067262 052737 000100 177560
6013 067270 012701 000060
6014 067274 011137 071206'
6015 067300 012721 071200'
6016 067304 011137 071210'
6017 067310 012711 000340
6018 067314
      067314 012700 000000
      067320 104441
6019 067322 000240
6020 067324 000776
6021
6022
6023 067326
      067326 012746 073011'
      067332 012746 000001
      067336 010600
      067340 104417
      067342 062706 000004
6024 067346
      067346 012700 000000
      067352 104441
6025 067354 005037 002224'
6026 067360 004737 015604'
6027 067364 103405
6031 067366 010001
6032 067370
      067370 104455
      067372 001760
      067374 003646'

;*****
;
;   THIS WRITE SUB-SYSTEM MEMORY COMMAND JUST HOLDS THE LEDS OFF
;
;*****
210$:
      MOVB    #0,T38BS1          ;CLEAR BIT #4
      MOVB    #25,T38BS0        ;STOP DRIVE TEST 22
      MOV     #T38PACKET,R4      ;SET UP NEW WRT. SUBSYS MEM. COMMAND
      MOV     R4,TSD8(R5)        ;SET THE PACKET ADDRESS
      JSR     PC,CHKTSSR         ;WAIT FOR SSR TO SET
      BCS     250$              ;BR IF CARRY SET (GOOD RETURN)
      MOV     R0,R1             ;SAVE CONTENTS OF TSSR
      ERDIF   ERRNO,T38SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP C$ERDF
                                .WORD 1007
                                .WORD T38SSR
                                .WORD PKTSSR
250$:  CKLOOP                   ;LOOP ON ERROR, IF FLAG SET
                                TRAP C$CLP1
      SETPRI  #PRI07            ;RAISE THE PRIORITY
                                MOV     #PRI07,R0
                                TRAP C$SPRI
      CLR     TTIION2           ;ASSUME INTERRUPTS ARE ENABLED
      BIT     #100,#TTICSR      ;ARE TTI INTERRUPTS ON ?
      BNE     710$              ;BRANCH IF YES
      INC     TTIION2           ;FLAG SET IF INTERRUPTS OFF
      BIS     #100,#TTICSR      ;ENABLE INTERRUPTS
      MOV     #TTIVEC,R1        ;START OF TTI VECTORS
      MOV     (R1),TVSAV2       ;SAVE THE CURRENT TTI VECTOR
      MOV     #590$,(R1)        ;SET NEW INTERRUPT ROUTINE
      MOV     (R1),TPSAV2       ;SAVE THE VECTOR PRIORITY
      MOV     #PRI07,(R1)       ;USE PRIORITY SEVEN
      SETPRI  #PRI00           ;LOWER INTERRUPT BR LEVEL
                                MOV     #PRI00,R0
                                TRAP C$SPRI
260$:  NOP                      ;ALLOW CNTL C
      BR      260$              ;LOOP UNTIL STOPPED
20$:  PRINTF    #T38MS2         ;TELL'EM WHAT TO TYPE
                                MOV     #T38MS2,-(SP)
                                MOV     #1,-(SP)
                                MOV     SP,R0
                                TRAP C$PNTF
                                ADD     #4,SP
                                MOV     #PRI00,R0
                                TRAP C$SPRI
      SETPRI  #PRI00           ;LOWER PRIORITY TO ALLOW INTERRUPTS
0$:  CLR     INTRECV            ;CLEAR INTERRUPT RECEIVED FLAG
      JSR     PC,SOFINIT        ;DO SOFT INIT OF CONTROLLER
      BCS     300$              ;BR IF SOFT INIT = OK
      MOV     R0,R1             ;SAVE CONTENTS OF TSSR
      ERDIF   ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
                                TRAP C$ERDF
                                .WORD 1008
                                .WORD SFIERR

```

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 10: MANUAL INTERVENTION

SEQ 214

Address	Offset	Hex	ASCII	Label	Operation	Comments
6033	067376	011644				
6034	067400			300\$:		
6035	067400	013737	002202	071740	MOV	UNITN,T38DSW ;SET UNIT NUMBER IN PACKET
6036	067406	012737	000040	071736	MOV	*BIT5,T38EAI ;ENABLE ATTENTION INTERRUPTS
6037	067414	012704	071720		MOV	*T38PK2,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
6038	067420	004737	010472		JSR	PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
6038	067424	103405			BCS	310\$;BR, IF COMMAND ISSUED OK
6042	067426	010001			MOV	R0,R1 ;SAVE CONTENTS OF TSSR
6043	067430	104456			ERRHRD	ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTICS FAILED
	067432	00161				TRAP C\$ERRHRD
	067434	005052				.WORD 1009
	067436	011644				.WORD WRTMSG
						.WORD SFMSG
6044	067440			310\$:		
6045	067440	012704	071750		MOV	*T38PK3,R4 ;SET UP NEW PACKET FOR MESS BUF REL
6046	067444	010465	000000		MOV	R4,T38DB(R5) ;MESSAGE BUFFER RELEASE,ACK,CVC*1 CMD
6047	067450	004737	016060		JSR	PC,WAITF ;WAIT FOR SSR TO SET
6048	067454	005002			CLR	R2 ;MAKE SURE ALL IS CLEAR
6049	067456	016501	000002		MOV	TSSR(R5),R1 ;GET TSSR STATUS
6050	067462	032701	000100		BIT	*OFL,R1 ;IS OFL SET
6051	067466	001402			BEQ	320\$;BR, IF OFL IS NOT SET
6052	067470	052702	000100		BIS	*OFL,R2 ;SET OFL IN EXPECTED
6053	067474	052702	000200	320\$:	BIS	*SSR,R2 ;SET UP EXPECTED
6054	067500	020201			CMP	R2,R1 ;IS EVERYTHING OK
6055	067502	001404			BEQ	350\$;BR, IF ALL IS WELL
6059	067504				ERRHRD	ERRNO,T38SST,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
	067504	104456				TRAP C\$ERRHRD
	067506	001762				.WORD 1010
	067510	072626				.WORD T38SST
	067512	011656				.WORD PKTSSR
6060	067514			350\$:	CKLOOP	;LOOP ON ERROR, IF FLAG SET
6061	067514	104406				TRAP C\$CLP1
	067516				PRINTF	*T38MS1 ;TELL OPERATOR TO TOGGLE SWITCH
	067516	012746	072716			MOV *T38MS1,-(SP)
	067522	012746	000001			MOV *1,-(SP)
	067526	010600				MOV SP,R0
	067530	104417				TRAP C\$PNTF
	067532	062706	000004			ADD *4,SP
6062	067536				PRINTF	*T38MS2 ;TELL OPERATOR TO DO *C TO EXIT
	067536	012746	073011			MOV *T38MS2,-(SP)
	067542	012746	000001			MOV *1,-(SP)
	067546	010600				MOV SP,R0
	067550	104417				TRAP C\$PNTF
	067552	062706	000004			ADD *4,SP
6063	067556				SETPRI	*PRI07 ;RAISE THE PRIORITY
	067556	012700	000340			MOV *PRI07,R0
	067562	104441				TRAP C\$SPRI
6064	067564	005037	071204		CLR	TTION2 ;ASSUME INTERRUPTS ARE ENABLED
6065	067570	032737	000100	177560	BIT	*100,*TTICSR ;ARE TTI INTERRUPTS ON ?
6066	067576	001005			BNE	720\$;BRANCH IF YES
6067	067600	005237	071204		INC	TTION2 ;FLAG SET IF INTERRUPTS OFF
6068	067604	052737	000100	177560	BIS	*100,*TTICSR ;ENABLE INTERRUPTS
6069	067612	012701	000060		MOV	*TTIVEC,R1 ;START OF TTI VECTORS
6070	067616	011137	071206		MOV	(R1),TVSAV2 ;SAVE THE CURRENT TTI VECTOR
6071	067622	012721	070500		MOV	*590,(R

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 10: MANUAL INTERVENTION

SEQ 215

6074	067636			SETPRI	0PRI00		LOWER INTERRUPT BR LEVEL	
	067636	012700	000000					MOV 0PRI00,R0
	067642	104441						TRAP C\$SPRI
6075	067644	000240		360\$:	NOP		ALLOW CONTROL C	
6076	067646	005737	002224'		TST INTRECV		DID AN INTERRUPT OCCUR ?	
6077	067652	001001			BNE 370\$		BRANCH IF YES	
6078	067654	000773			BR 360\$		WAIT SOME MORE FOR INTERRUPT	
6079	067656			370\$:	PRINTF 0T38INT		"INTERRUPT RECEIVED"	
	067656	012746	072506'					MOV 0T38INT,-(SP)
	067662	012746	000001					MOV 01,-(SP)
	067666	010600						MOV SP,R0
	067670	104417						TRAP C\$PNTF
	067672	062706	030004					ADD 04,SP
6080	067676	016501	000002		MOV TSSR(R5),R1		READ TSSR STATUS	
6081	067702	032701	000100		BIT 00FL,R1		CHECK THE OFF-LINE BIT	
6082	067706	001011			BNE 380\$		BR, IF DRIVE IS OFF-LINE	
6083	067710				PRINTF 0T38ONL		"DRIVE IS NOW ON-LINE"	
	067710	012746	072536'					MOV 0T38ONL,-(SP)
	067714	012746	000001					MOV 01,-(SP)
	067720	010600						MOV SP,R0
	067722	104417						TRAP C\$PNTF
	067724	062706	000004					ADD 04,SP
6084	067730	000410			BR 390\$		ALMOST DONE	
6085	067732			380\$:	PRINTF 0T38OFL		"DRIVE IS NOW OFF-LINE"	
	067732	012746	072572'					MOV 0T38OFL,-(SP)
	067736	012746	000001					MOV 01,-(SP)
	067742	010600						MOV SP,R0
	067744	104417						TRAP C\$PNTF
	067746	062706	000004					ADD 04,SP
6086	067752	005037	002224'	390\$:	CLR INTRECV		CLEAR INTERRUPT FLAG	
6087	067756	000137	067400'		JMP 300\$		TRY AGAIN	
6088	067762			25\$:	GMANIL T38MSG,T38DAT,-1,NO		WAIT FOR OPERATOR TO MOUNT TAPE	
	067762	104443						TRAP C\$GMAN
	067764	000404						BR 10000\$
	067766	073544'						.WORD T38DAT
	067770	000120						.WORD T\$CODE
	067772	073055'						.WORD T38MSG
	067774	177777						.WORD -1
	067776							
6089	067776				BNCOMPLETE 25\$		RETRY IF ERROR	10000\$:
	067776	103371						
6090	070000	005737	073544'		TST T38DAT		DID OPERATOR SAY 'YES' ?	BCC 25\$
6091	070004	001002			BNE 27\$		BRANCH IF YES	
6092	070006	000137	066452'		JMP 2\$		RETURN TO MAIN MENU	
6093	070012			27\$:				
6094	070012	004737	015604'		JSR PC,SOFINIT		DO SOFT INIT OF CONTROLLER	
6095	070016	103405			BCS 400\$		BR IF SOFT INIT - OK	
6099	070020	010001			MOV R0,R1		SAVE CONTENTS OF TSSR	
6100	070022				ERRDF ERRNO,SFIERR,SFIMSG		DEVICE FATAL ERROR DURING INIT	
	070022	104455						TRAP C\$ERDF
	070024	001763						.WORD 1011
	070026	003646'						.WORD SFIERR
	070030	011644'						.WORD SFIMSG
6101	070032			400\$:	CKLOOP		LOOP IF SELECTED	
	070032	104406						TRAP C\$CLP1
6102	070034	013737	002202' 071740'		MOV UNITN,T38DSW		SET UNIT NUMBER	
6103	070042	012704	071720'		MOV 0T38PK2,R4		SUBROUTINE NEEDS PACKET ADDRESS	

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 10: MANUAL INTERVENTION

SEQ 216

```

6104 070046 004737 010472'      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
6105 070052 103405              BCS      410$      ;BR, IF COMMAND ISSUED OK
6109 070054 010001              MOV      R0,R1      ;SAVE CONTENTS OF TSSR
6110 070056              ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
                                TRAP      C$ERHRD
                                .WORD     1012
                                .WORD     WRTMSG
                                .WORD     SFIMSG
6111 070066              410$: CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
6112 070070 013701 071244'      MOV      T38BFR+6,R1    ;PICK UP XSTO CONTENTS
6113 070074 010102              MOV      R1,R2      ;SET UP EXPECTED
6114 070076 052702 000004      BIS      @BIT2,R2    ;SET UP THE WRITE LOCKED BIT
6115 070102 020102              CMP      R1,R2      ;ARE THEY CORRECT
6116 070104 01406              BEQ      430$      ;BR, IF ALL IS WELL (OK)
6120 070106              ERRHRD  ERRNO,T38WRL,EXPREC ;"WRITE LOCKED BIT IS NOT SET ETC."
                                TRAP      C$ERHRD
                                .WORD     1013
                                .WORD     T38WRL
                                .WORD     EXPREC
6121 070116 005237 002222'      INC      FATFLG      ;SET FATAL FLAG
6122 070122 104406              430$: CKLOOP      ;LOOP IF SELECTED
                                TRAP      C$CLP1
6123 070124 005737 002222'      TST      FATFLG      ;WAS THE DRIVE NOT WRITE LOCKED
6124 070130 001402              BEQ      435$      ;BR, IF FLAG NOT SET
6125 070132 000137 066452'      JMP      2$      ;RE-WRITE MENU
6126 070136 017737 112762 071772' 435$: MOV      @FREE,T38WR    ;SET UP WRITE BUFFER ADDRESS
6127 070144 012704 071770'      MOV      @T38PK4,R4    ;GET PACKET ADDRESS
6128 070150 010465 000000      MOV      R4,T50B(R5)  ;SET THE PACKET ADDRESS
6129 070154 004737 016060'      JSR      PC,WAITF    ;WAIT FOR SSR TO SET
6130 070160 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR
6131 070164 012702 100206      MOV      @SC!SSR!BIT1!BIT2,R2 ;SET UP EXPECTED
6132 070170 020102              CMP      R1,R2      ;ARE THEY EQUAL (CORRECT)
6133 070172 001404              BEQ      440$      ;BR, IF CORRECT STATUS
6137 070174              ERRHRD  ERRNO,T38WRT,PKTSSR ;"TSSR INCORRECT AFTER WRITE COMMAND
                                TRAP      C$ERHRD
                                .WORD     1014
                                .WORD     T38WRT
                                .WORD     PKTSSR
6138 070204              440$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
6139 070206 013701 071244'      MOV      T38BFR+6,R1    ;READ XSTO CONTENTS
6140 070212 010102              MOV      R1,R2      ;SET UP EXPECTED
6141 070214 052702 004000      BIS      @BIT11,R2   ;SET THE WRITE LOCK ERROR BIT (XSTO)
6142 070220 020102              CMP      R1,R2      ;WAS THE BIT SET
6143 070222 001404              BEQ      450$      ;BR, IF IT WAS (GOOD)
6147 070224              ERRHRD  ERRNO,T38WLE,EXPREC ;"WRITE LOCK ERROR BIT NOT SET"
                                TRAP      C$ERHRD
                                .WORD     1015
                                .WORD     T38WLE
                                .WORD     EXPREC
6148 070234              450$: CKLOOP      ;LOOP IF SELECTED
6149 070236 000137 066452'      JMP      2$      ;GO BACK TO MENU
6150
6151
6152
;*****
;      SERVO EXERCISER NO LONGER USED

```

```

6153 ;*****
6154 070242 30$: PRINTB 0T38MS3 ;EXE ANY OTHER MENU SELECTION TO STOP
6155 070242 012746 072055' MOV 0T38MS3,-(SP)
      070246 012746 000001 MOV 01,-(SP)
      070252 010600 MOV SP,R0
      070254 104414 TRAP C$PNTB
      070256 062706 000004 ADD 04,SP
6156 070262 004737 073546' JSR PC,T38REST ;SET PACKET TO INITIAL VALUES
6157 070266 004737 015604' JSR PC,SOFINIT ;DO SOFT INIT OF CONTROLLER
6158 070272 103405 BCS 500$ ;BR IF SOFT INIT = OK
6162 070274 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
6163 070276 ERRDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
      070276 104455 TRAP C$ERDF
      070300 001770 .WORD 1016
      070302 003646' .WORD SFIERR
      070304 011644' .WORD SFIMSG
6164 070306 013737 002202' 071740' 500$: MOV UNITN,T38DSW ;SET UNIT NUMBER
6165 070314 012704 071720' MOV 0T38PK2,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
6166 070320 004737 010472' JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
6167 070324 103405 BCS 510$ ;BR, IF COMMAND ISSUED OK
6171 070326 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
6172 070330 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
      070330 104456 TRAP C$ERHRD
      070332 001771 .WORD 1017
      070334 005052' .WORD WRTMSG
      070336 011644' .WORD SFIMSG
6173 070340 510$:
6174 070340 112737 000000 071231' MOVB 00,T38BS1 ;CLEAR BIT 04
6175 070346 112737 000020 071230' MOVB 020,T38BS0 ;EXECUTE DRIVE TEST 22
6176 070354 012704 071220' MOV 0T38PACKET,R4 ;SET UP NEW WRT. SUBSYS MEM. COMMAND
6177 070360 010465 000000 MOV R4,TSR(R5) ;SET THE PACKET ADDRESS
6178 070364 004737 016146' JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
6179 070370 103405 BCS 550$ ;BR IF CARRY SET (GOOD RETURN)
6180 070372 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
6184 070374 ERRDF ERRNO,T38SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      070374 104455 TRAP C$ERDF
      070376 001772 .WORD 1018
      070400 072416' .WORD T38SSR
      070402 011656' .WORD PKTSSR
6185 070404 550$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
      070404 104406 TRAP C$CLP1
6186 070406 SETPRI 0PRI07 ;RAISE THE PRIORITY
      070406 012700 000340 MOV 0PRI07,R0
      070412 104441 TRAP C$SPRI
6187 070414 005037 071204' CLR TTION2 ;ASSUME INTERRUPTS ARE ENABLED
6188 070420 032737 000100 177560 BIT 0100,0TTICSR ;ARE TTI INTERRUPTS ON ?
6189 070426 001000 BNE 555$ ;BRANCH IF YES
6190 070430 005237 071204' INC TTION2 ;FLAG SET IF INTERRUPTS OFF
6191 070434 052737 000100 177560 BIS 0100,0TTICSR ;ENABLE INTERRUPTS
6192 070442 012701 000060 555$: MOV 0TTIVEC,R1 ;START OF TTI VECTORS
6193 070446 011137 071206' MOV (R1),TVSAV2 ;SAVE THE CURRENT TTI VECTOR
6194 070452 012721 070500' MOV 0590,(R1) ;SET NEW INTERRUPT ROUTINE
6195 070456 011137 071210' MOV (R1),TPSAV2 ;SAVE THE VECTOR PRIORITY
6196 070462 012711 000340 MOV 0PRI07,(R1) ;USE PRIORITY SEVEN
6197 070466 012700 000000 SETPRI 0PRI00 ;LOWER INTERRUPT BR LEVEL
      MOV 0PRI00,R0

```

[illegible]

N1

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 10: MANUAL INTERVENTION

SEQ 220

6287	071160	005037	003134'	CLR	KTENABLE				
6288	071164	004737	073604'	JSR	PC,T38MBP				;NO KT11 STUFF EITHER
6289	071170	000137	066452'	JMP	2\$				;GO PRINT MESSAGE BUFFER CONTENTS
6290									;GO BACK TO MENU
6291									
6292	071174	000137	000200	63\$:	JMP	200			;REALLY RETURN TO THE SUPERVISOR
6293	071200			64\$:	EXIT	TST			;LEAVE TEST
	071200	104432							
	071202	003054							TRAP J\$EXIT
									.WORD L10075-
6294									
6295									
6296									
6297									
6298									
6299									
6300									
6301									
6302									
6303									
6304									
6305	071204	000000		TTION2:	.WORD	0			;WORD SET IF SUPERVISOR TTI INTER OFF
6306	071206	000000		TVSAV2:	.WORD	0			;SAVE TTI VECTOR
6307	071210	000000		TPSAV2:	.WORD	0			;SAVE TTI PRIORITY
6308									
6309	071212	000000		T38DLY:	.WORD	0			;DELAY COUNTER FOR TEST
6311	071214				.BLKB	10-<.-TSV2&7>			
6313	071220			T38PACKET:					;COMMAND PACKET FOR TEST
6314	071220	140006			.WORD	140006			;WRITE SUBSYSTEM MEM. CMD. ACK,CVC=1
6315	071222	071230'			.WORD	T38TAD			;ADDRESS OF CHARACTERISTICS BLOCK
6316	071224	000000			.WORD	0			
6317	071226	000012			.WORD	10.			;STARTING VALUE OF BLOCK SIZE
6318	071230			T38TAD:					;CHARACTERISTICS DATA BLOCK
6319	071230	00)		T38BS0:	.BYTE	0			;BSEL0 BYTE
6320	071231	00)		T38BS1:	.BYTE	0			;BSEL1 BYTE
6321	071232	000000		T38BS2:	.WORD	0			;BSEL1 WORD
6322	071234	000000			.WORD	0			;DATA
6323	071236			T38BFR:	.BLKB	150.			;MESSAGE BUFFER
6324	071712	000000		T38EB:	.WORD				;END OF BUFFER ADDRESS
6325									
6326									
6328	071714				.BLKB	10-<.-TSV2&7>			
6330	071720			T38PK2:					;COMMAND PACKET FOR TEST
6331	071720	140004			.WORD	140004			;WRITE CHARA. MEM. CMD. ACK,CVC=1
6332	071722	071730'			.WORD	T38DTA			;ADDRESS OF SELECT DATA BLOCK
6333	071724	000000			.WORD	0			
6334	071726	000012			.WORD	10.			;STARTING VALUE OF BLOCK SIZE
6335									
6336									
6337	071730			T38DTA:					;SELECT DATA BLOCK
6338	071730	071236'			.WORD	T38BFR			;ADDRESS OF MESSAGE BUFFER
6339	071732	000000			.WORD	0			
6340	071734	000400			.WORD	256.			;LENGTH OF MESSAGE BUFFER
6341	071736	000000		T38EAI:	.WORD	0			;EAI BIT WORD
6342	071740	000000		T38DSW:	.WORD	0			;DRIVE SELECT WORD ETC
6344	071742				.BLKB	10-<.-TSV2&7>			
6346	071750	140212		T38PK3:	.WORD	140212			;MESSAGE BUFFER RELEASE COMMAND
6347	071752	000000			.WORD	0			;NOT USED

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 10: MANUAL INTERVENTION

SEQ 221

```

6348 071754 000000      .WORD 0      ;NOT USED
6349 071756 000000      .WORD 0      ;NOT USED
6350 071760 000000      .WORD 0      ;NOT USED
6351
6352      ;WRITE TAPE PACKET
6353
6355 071762
6357 071770 140005      T38PK4: .BLK8 10-<.-TSV2&7>
6358 071772 000000      T38WR:  .WORD 140005      ;WRITE, ACK, CVC=1 COMMAND
6359 071774 000000      .WORD 0      ;ADDRESS OF WRITE BUFFER
6360 071776 000400      T38SIZ: .WORD 0      ;MORE ADDRESS OF WRITE BUFFER
6361
6362
6363
6364
6365
6366      ;LOCAL TEXT MESSAGES FOR TEST
6367
6368
6369
6370
6371
6372
6373 072000      123      164      141      T38NE:  .ASCIZ 'Stand-alone Manual Intervention Not Executed'
6374 072055      045      116      045      T38MS3: .ASCIZ 'MNA Type <RETURN> To Stop Servo Exerciser, Return To Menu'
6375 072150      124      123      123      T38WRT: .ASCIZ 'TSSR Not Correct After WRITE, With WRITE PROTECT On'
6376 072234      127      122      111      T38WRL: .ASCIZ 'WRITE LOCKED Bit Not Set In XST0'
6377 072275      127      122      111      T38WLE: .ASCIZ 'WRITE LOCK ERROR Bit Not Set In XST0'
6378 072342      127      122      111      T38NBA: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted'
6379 072416      103      157      156      T38SSR: .ASCIZ 'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
6380 072506      045      116      045      T38INT: .ASCIZ 'MNA Interrupt Received'
6381 072536      045      116      045      T38OVL: .ASCIZ 'MNA Drive Is Now ON-LINE'
6382 072572      045      116      045      T38OFL: .ASCIZ 'MNA Drive Is Now OFF-LINE'
6383 072626      103      157      156      T38SST: .ASCIZ 'Contents of TSSR Incorrect After MESSAGE BUFFER RELEASE'
6384 072716      045      116      045      T38MS1: .ASCIZ 'MNA Type ON-LINE Switch to Generate ATTENTION Interrupts'
6385 073011      045      116      045      T38MS2: .ASCIZ 'MNA Type RETURN To Return To Menu'
6386 073055      111      163      040      T38MSG: .ASCIZ 'Is Write-Protected Tape Mounted'
6387 073115      115      141      156      T38ID:  .ASCIZ 'Manual Intervention'
6388
6389 073142      073166' 073240' 073266' MIMENU: .EVEN
6390 073156      073435' 073500' 073543' .WORD 11,21,31,41,51,61
6391
6392 073166      012      123      105      11:  .ASCIZ <12>' SELECT OPERATION FROM FOLLOWING OPTIONS:'
6393 073240      012      011      060      21:  .ASCIZ <12>' 0 Display This Menu'
6394 073266      011      061      011      31:  .ASCIZ ' 1 Turn On All M7455 LED's'
6395 073320      011      062      011      41:  .ASCIZ ' 2 Turn Off All M7455 LED's'
6396 073353      011      063      011      51:  .ASCIZ ' 3 Offline/Online Attention'
6397 073407      011      064      011      61:  .ASCIZ ' 4 Write Protect Test'
6398 073435      011      065      011      81:  .ASCIZ ' 5 Print Extended Transport Status'
6399 073500      011      066      011      91:  .ASCIZ ' 6 Return to Diagnostic Supervisor'
6400 073543      000
6401
6402
6403
6404      ;LOCAL STORAGE FOR THIS TEST
6405
6406

```

6407	073544	000000		T38DAT: .WORD 0					
6408	073546			T38REST:					
6409	073546			SAVREG					
6410	073552	012701	071220'	MOV	0T38PACKET,R1				
6411	073556	012721	140206	MOV	0140206,(R1)+				
6412	073562	012721	071230'	MOV	0T38TAD,(R1)+				
6413	073566	005021		CLR	(R1)+				
6414	073570	012721	000006	MOV	06.,(R1)+				
6415	073574	005021		CLR	(R1)+				
6416	073576	005021		CLR	(R1)+				
6417	073600	005011		CLR	(R1)				
6418	073602	000207		RTS	PC				
6419									
6420									
6421									
6422									
6423									
6424									
6425									
6426									
6427									
6428									
6429									
6430									
6431									
6432									
6433									
6434									
6435									
6436	073604			T38MBP:					
6437	073604			SAVREG					
6438	073610	010005		MOV	R0,R5				
6439	073612	005737	003134'	TST	KTENABLE				
6440	073616	001001		BNE	9104				
6441	073620	005001		CLR	R1				
6442	073622	010103		MOV	R1,R3				
6443	073624	006100		ROL	R0				
6444	073626	006101		ROL	R1				
6445	073630			PRINTX	0T38AS0,R1,R5				
	073630	010546							
	073632	010146							
	073634	012746	074106'						
	073640	012746	000003						
	073644	010600							
	073646	104415							
	073650	062706	000010						
6446	073654			PRINTX	0T38AS1				
	073654	012746	074153'						
	073660	012746	000001						
	073664	010600							
	073666	104415							
	073670	062706	000004						
6447	073674	010501		MOV	R5,R1				
6448	073676	010300		MOV	R3,R0				
6449	073700	001403		BEQ	9134				
6450	073702	004737	017130'	JSR	PC,SETMAP				
6451	073706	010005		MOV	R0,R5				

LOGICAL RESPONSE TO QUESTION

SAVE THE REGISTERS
 START OF THE PACKET
 WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
 ADDRESS OF DATA BLOCK
 EXTENDED ADDRESS
 SIZE OF DATA BLOCK IN BYTES
 CLEAR BSEL0 AND BSEL1
 CLEAR SEL2
 CLEAR DATA AREA
 RETURN

THIS ROUTINE PRINTS THE CONTENTS OF
 THE 256 BYTE MESSAGE BUFFER RETURNED BY THE
 TSV-05.

INPUT:

R0 LOW ORDER ADDRESS OF MESSAGE BUFFER
 R1 HIGH ORDER ADDRESS OF MESSAGE BUFFER
 NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR

SAVE THE REGISTERS
 SAVE LOW ORDER ADDRESS
 ADDRESS ABOVE 28K?
 BR IF YES
 SET HIGH ORDER ADDRESS TO 0
 SAVE HIGH ORDER ADDRESS
 SHIFT BIT15 TO C BIT
 SHIFT TO HIGH ORDER FOR PRINTOUT
 PRINT MESSAGE BUFFER ADDRESS

MOV R5, -(SP)
 MOV R1, -(SP)
 MOV 0T38AS0, -(SP)
 MOV 03, -(SP)
 MOV SP,R0
 TRAP C:PNTX
 ADD 010,SP

PRINT HEADER FOR CONTENTS

MOV 0T38AS1, -(SP)
 MOV 01, -(SP)
 MOV SP,R0
 TRAP C:PNTX
 ADD 04,SP

COPY LOW ORDER ADDRESS
 COPY HIGH ORDER ADDRESS
 BR IF NOT ABOVE 28K
 SETUP PAR ADDRESS IN R0
 GET PAR FORMAT ADDRESS ABOVE 28K

D2

TSV5 - HARDWARE TEST 10: MANUAL INTERVENTION MACRO M1113 01-FEB-84 17:02

SEQ 223

```

6452 073710 010537 074254'      913$: MOV R5,T38CNT      ;HOLD ADDRESS
6453 073714 011504              911$: MOV (R5),R4      ;GET BUFFER ENTRY
6454 073716 022704 125252        CMP #125252,R4      ;CHECK FOR NO LOAD CONDITION
6455 073722 001417              BEQ 912$              ;BR, IF BUFFER WASN'T LOADED
6456 073724 010403              MOV R4,R3              ;MAKE COPY
6457 073726 042704 170377        BIC #170377,R4      ;ONLY BITS 11,10,9 AND 8 ARE SAVED
6458 073732 000241              CLC                  ;CLEAR CARRY
6459 073734 006004              ROR R4                  ;11 TO 10 BIT POSITION
6460 073736 006004              ROR R4                  ;10 TO 9 BIT POSITION
6461 073740 006004              ROR R4                  ;9 TO 8 BIT POSITION
6462 073742 006004              ROR R4                  ;8 TO 7 BIT POSITION
6463 073744 042703 177760        BIC #177760,R3      ;ONLY BITS 3,2,1 AND 0 ARE SAVED
6464 073750 060403              ADD R4,R3              ;"OR'EM TOGETHER
6465 073752 010325              MOV R3,(R5)+          ;PUT BACK IN BUFFER
6466 073754 020527 071712'      CMP R5,#T38EB      ;END OF BUFFER YET
6467 073760 001355              BNE 911$              ;BR, IF NOT AT END YET
6468 073762 013705 074254'      912$: MOV T38CNT,R5    ;PUT ADDRESS BACK
6469 073766 012704 000001      MOV #1,R4            ;START BYTE NUMBER AT ONE
6470 073772              915$: PRINTX #T38ASN,R4,(R5)+ ;PRT MEM BUFFER W/NEWLINE
        073772 012546              MOV (R5)+,-(SP)
        073774 010446              MOV R4,-(SP)
        073776 012746 074230'      MOV #T38ASN,-(SP)
        074002 012746 000003      MOV #3,-(SP)
        074006 010600              MOV SP,R0
        074010 104415              TRAP C$PNTX
        074012 062706 000010      ADD #10,SP
6471 074016 005037 074254'      CLR T38CNT          ;CLEAR COUNTER
6472 074022 000412              BR 921$              ;SKIP OTHER PRINT
6473 074024              920$: PRINTX #T38ASC,R4,(R5)+ ;PRINT THE CONTENTS OF MEMORY BUFFER
        074024 012546              MOV (R5)+,-(SP)
        074026 010446              MOV R4,-(SP)
        074030 012746 074211'      MOV #T38ASC,-(SP)
        074034 012746 000003      MOV #3,-(SP)
        074040 010600              MOV SP,R0
        074042 104415              TRAP C$PNTX
        074044 062706 000010      ADD #10,SP
6474 074050 005237 074254'      921$: INC T38CNT      ;BUMP COUNTER
6475 074054 005204              INC R4              ;NUMBER OF THE NEXT
6476 074056 020427 000200        CMP R4,#128        ;DONE ALL YET ?
6477 074062 003010              BGT 50$              ;BRANCH IF ALL DONE
6478 074064 023727 074254' 000004 CMP T38CNT,#4    ;DONE FOUR YET
6479 074072 001401              BEQ 925$              ;BR, IF THREE DONE
6480 074074 000753              BR 920$              ;KEEP GOING
6481 074076 005037 074254'      925$: CLR T38CNT      ;CLEAR COUNTER
6482 074102 000733              BR 915$              ;PRINT WITH NEW LINE
6483 074104 000207              50$: RTS PC          ;RETURN
6484
6485 074106 045 116 045 T38AS0: .ASCIZ 'NNA Message Buffer Address = #01#05'
6486 074153 045 116 045 T38AS1: .ASCIZ 'NNA Message Buffer Contents:'
6487 074211 045 101 040 T38ASC: .ASCIZ 'A #D4#A: #03'
6488 074230 045 116 045 T38ASN: .ASCIZ 'NNA Byte#D4#A: #03'
6489
6490 074254 000000      T38CNT: .WORD          ;COUNTER FOR PRINT
6491 074256
        074256
        074256 104401
6492

```

L10075: TRAP C\$ETST

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 11: CONFIGURATION TYPEOUT

SEQ 224

6493
6494
6495
6496
6497
6498
6499
6500
6501
6502
6503
6504
6505
6506
6507
6508
6509
6510
6511
6512
6513
6514
6515
6516
6517
6518
6519
6520
6521
6522
6523
6524
6525
6526
6527
6528
6529
6530
6531 074260
074260
6536 074260
074260 104421
6537 074262 001403
6538 074264 012700 076253'
6539 074270 000402
6540 074272 012700 077402'
6541 074276 004737 016322'
6542 074302 004737 020320'
6543 074306 103402
6544 074310 000137 075470'
6545 074314
6546 074314 004737 015604'
6547 074320 103405
6551 074322 010001
6552 074324
074324 104455
074326 002115

.SBTTL TEST 11: CONFIGURATION TYPEOUT

;THIS IS A STANDALONE ROUTINE THAT PRINTS OUT ON THE CONSOLE TERMINAL
;THE CONFIGURATION OF THE M7455 MODULE AND TSV05 SUBSYSTEM. SPECIFICALLY,
;THE FOLLOWING INFORMATION IS PRESENTED:

- ; 1.0 STATE OF THE EXTENDED FEATURES SWITCH ON THE M7455: ON (EXTENDED
; FEATURES ENABLED) OR OFF (EXTENDED FEATURES DISABLED),
- ; 2.0 STATE OF THE BUFFERING ENABLE SWITCH ON THE M7455: ON
; (BUFFERING ENABLED) OR OFF (BUFFERING DISABLED),
- ; 3.0 MICROCODE REVISION LEVEL OF THE M7455,
- ; 4.0 NUMBER OF TAPE TRANSPORTS CONNECTED TO THE CONTROLLER,
- ; 5.0 UNIT SELECT CODE AND STATE (ONLINE/OFFLINE, WRITE ENABLED/PROTECTED)
; OF EACH CONNECTED TRANSPORT. IN ADDITION, THE PROGRAM WILL INDICATE,
; FOR EACH ON-LINE TRANSPORT, WHETHER OR NOT IT IS EQUIPPED WITH THE
; EXTENDED TAPE STATUS READOUT FEATURE.

;THE OPERATOR IS EXPECTED TO READ THE PRINTOUT AND VERIFY THAT IT MATCHES
;THE ACTUAL CONFIGURATION AT HAND. IF, FOR EXAMPLE, THE PROGRAM INDICATES
;THAT IT "SEES" TWO TRANSPORTS CONNECTED WHEN IN FACT ONLY ONE IS PRESENT,
;THE OPERATOR MUST INTERPRET THIS AS AN ERROR AND ATTEMPT TO FIND THE
;CAUSE (BAD CABLE, FAULTY UNIT-SELECT DECODING IN THE TRANSPORT, ETC.).
;[SINCE THE CONTROLLER CAN ONLY ACCESS UNIT 0 IF IT IS IN "STANDARD"
;MODE, THE PROGRAM WILL FORCE THE MODULE INTO EXTENDED MODE VIA THE
;WRITE SUBSYSTEM MEMORY COMMAND IN ORDER TO SCAN FOR CONNECTED TRANSPORTS.]

;THIS ROUTINE, WHEN ITS ACTIONS ARE COMPLETED, WILL EXIT BACK TO THE
;DIAGNOSTIC SUPERVISOR SO THAT IF ADDITIONAL UNITS (CONTROLLERS) ARE
;SELECTED (E.G., FROM THE INITIAL STARTUP DIALOG), THE ROUTINE WILL BE
;REENTERED SO THAT THEIR CONFIGURATIONS CAN BE PRINTED.

BGNTST

RFLAGS RO

BEQ 10;
MOV 0T39NE,RC
BR 11;
10: MOV 0TST39ID,RO
11: JSR PC,TSTSETUP
JSR PC,CHKMAN
BCS 20;
JMP 64;
20: JSR PC,SOFINIT
BCS 25;
MOV RO,R1
ERRDF ERRNO,SFIERR,SFIMSG

T11::
;GET OPERATOR FLAGS
TRAP C3RFLA
;BR, IF OK TO RUN
;"TEST NOT EXECUTED"
;JUMP OUT OF TEST IF NOT
;TEST ID MESSAGE
;DO THE COMMON SETUP
;IS MANUAL INTERVENTION ALLOWED?
;BR, IF MANUAL INTERVENTION ALLOWED
;JUMP TO OUT IF NOT
;DO SOFT INIT OF CONTROLLER
;BR IF SOFT INIT = OK
;SAVE CONTENTS OF TSSR
;DEVICE FATAL ERROR DURING INIT
TRAP C3ERDF
;WORD 1101

TSVS - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 11: CONFIGURATION TYPEOUT

SEQ 226

Address	Instruction	Comments	Trap	Trap Vector
074550	104415		TRAP	C\$PNTX
074552	062706	000004	ADD	04,SP
6577 074556	042701	177700		
6578 074562	010137	077340'		
6579 074566				
074566	013746	077340'		
074572	012746	077263'		
074576	012746	000002		
074602	010600			
074604	104415			
074606	062706	000006		
6590 074612	004737	015604'		
6581 074616	103405			
6585 074620	010001			
6586 074622				
074622	104455			
074624	002117			
074626	003646'			
074630	011644'			
6587 074632				
074632	104406			
6588 074634	013737	002202' 076220'		
6589 074642	012704	076200'		
6590 074646	004737	010472'		
6591 074652	103405			
6595 074654	010001			
6596 074656				
074656	104456			
074660	002120			
074662	005052'			
074664	011644'			
6597 074666				
074666	104406			
6598 074670	005737	002226'		
6599 074674	001036			
6600 074676	112737	000200 075511'		
6601 074704	112737	000010 075510'		
6602 074712	012704	075500'		
6603 074716	010465	000000		
6604 074722	004737	016146'		
6605 074726	103405			
6606 074730	010001			
6610 074732				
074732	104456			
074734	002121			
074736	076755'			
074740	011656'			
6611 074742				
074742	104406			
6612 074744	012704	076200'		
6613				
6614				
6615				
6616				
6617				
6618				
6619 074750	004737	010472'		

TSVS - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 11: CONFIGURATION TYPEOUT

SEQ 227

PC	PC+4	PC+8	PC+12	PC+16	PC+20	PC+24	PC+28	PC+32	PC+36	PC+40	PC+44	PC+48	PC+52	PC+56	PC+60	PC+64	PC+68	PC+72	PC+76	PC+80	PC+84	PC+88	PC+92	PC+96	PC+100	PC+104	PC+108	PC+112	PC+116	PC+120	PC+124	PC+128	PC+132	PC+136	PC+140	PC+144	PC+148	PC+152	PC+156	PC+160	PC+164	PC+168	PC+172	PC+176	PC+180	PC+184	PC+188	PC+192	PC+196	PC+200	PC+204	PC+208	PC+212	PC+216	PC+220	PC+224	PC+228	PC+232	PC+236	PC+240	PC+244	PC+248	PC+252	PC+256	PC+260	PC+264	PC+268	PC+272	PC+276	PC+280	PC+284	PC+288	PC+292	PC+296	PC+300	PC+304	PC+308	PC+312	PC+316	PC+320	PC+324	PC+328	PC+332	PC+336	PC+340	PC+344	PC+348	PC+352	PC+356	PC+360	PC+364	PC+368	PC+372	PC+376	PC+380	PC+384	PC+388	PC+392	PC+396	PC+400	PC+404	PC+408	PC+412	PC+416	PC+420	PC+424	PC+428	PC+432	PC+436	PC+440	PC+444	PC+448	PC+452	PC+456	PC+460	PC+464	PC+468	PC+472	PC+476	PC+480	PC+484	PC+488	PC+492	PC+496	PC+500	PC+504	PC+508	PC+512	PC+516	PC+520	PC+524	PC+528	PC+532	PC+536	PC+540	PC+544	PC+548	PC+552	PC+556	PC+560	PC+564	PC+568	PC+572	PC+576	PC+580	PC+584	PC+588	PC+592	PC+596	PC+600	PC+604	PC+608	PC+612	PC+616	PC+620	PC+624	PC+628	PC+632	PC+636	PC+640	PC+644	PC+648	PC+652	PC+656	PC+660	PC+664	PC+668	PC+672	PC+676	PC+680	PC+684	PC+688	PC+692	PC+696	PC+700	PC+704	PC+708	PC+712	PC+716	PC+720	PC+724	PC+728	PC+732	PC+736	PC+740	PC+744	PC+748	PC+752	PC+756	PC+760	PC+764	PC+768	PC+772	PC+776	PC+780	PC+784	PC+788	PC+792	PC+796	PC+800	PC+804	PC+808	PC+812	PC+816	PC+820	PC+824	PC+828	PC+832	PC+836	PC+840	PC+844	PC+848	PC+852	PC+856	PC+860	PC+864	PC+868	PC+872	PC+876	PC+880	PC+884	PC+888	PC+892	PC+896	PC+900	PC+904	PC+908	PC+912	PC+916	PC+920	PC+924	PC+928	PC+932	PC+936	PC+940	PC+944	PC+948	PC+952	PC+956	PC+960	PC+964	PC+968	PC+972	PC+976	PC+980	PC+984	PC+988	PC+992	PC+996	PC+1000	PC+1004	PC+1008	PC+1012	PC+1016	PC+1020	PC+1024	PC+1028	PC+1032	PC+1036	PC+1040	PC+1044	PC+1048	PC+1052	PC+1056	PC+1060	PC+1064	PC+1068	PC+1072	PC+1076	PC+1080	PC+1084	PC+1088	PC+1092	PC+1096	PC+1100	PC+1104	PC+1108	PC+1112	PC+1116	PC+1120	PC+1124	PC+1128	PC+1132	PC+1136	PC+1140	PC+1144	PC+1148	PC+1152	PC+1156	PC+1160	PC+1164	PC+1168	PC+1172	PC+1176	PC+1180	PC+1184	PC+1188	PC+1192	PC+1196	PC+1200	PC+1204	PC+1208	PC+1212	PC+1216	PC+1220	PC+1224	PC+1228	PC+1232	PC+1236	PC+1240	PC+1244	PC+1248	PC+1252	PC+1256	PC+1260	PC+1264	PC+1268	PC+1272	PC+1276	PC+1280	PC+1284	PC+1288	PC+1292	PC+1296	PC+1300	PC+1304	PC+1308	PC+1312	PC+1316	PC+1320	PC+1324	PC+1328	PC+1332	PC+1336	PC+1340	PC+1344	PC+1348	PC+1352	PC+1356	PC+1360	PC+1364	PC+1368	PC+1372	PC+1376	PC+1380	PC+1384	PC+1388	PC+1392	PC+1396	PC+1400	PC+1404	PC+1408	PC+1412	PC+1416	PC+1420	PC+1424	PC+1428	PC+1432	PC+1436	PC+1440	PC+1444	PC+1448	PC+1452
----	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------

Address	Instruction	Comments	Register/Label	Value	Operation	Register/Label	Value
075176	104415						
075200	062706	000006					
6651 075204	012737	125252	075616'	220\$:	MOV	0125252,T39BFR+100	;SET 1 LOC TO KNOWN VALUE
6652 075212	112737	000000	075511'		MOVB	00,T39BS1	;EXTENDED TAPE STATUS
6653 075220	112737	000024	075510'		MOVB	024,T39BS0	;EXTENDED TAPE STATUS
6654 075226	012704	075500'			MOV	0T39PACKET,R4	;WRITE SUBSYS MEM PACKET
6655 075232	010465	000000			MOV	R4,TSD8(R5)	;ISSUE COMMAND
6656 075236	012737	000144	075474'		MOV	0100.,T39DLY	;SET UP DELAY ROUTINE
6657 075244	004737	016060'		222\$:	JSR	PC,WAITF	;WAIT AWHILE FOR SSR TO SET
6658 075250	016501	000002			MOV	TSSR(R5),R1	;SEE IF IT REALLY DID
6659 075254	032701	000200			BIT	0SSR,R1	;JUST CHECK THAT BIT
6660 075260	001017				BNE	225\$	;BR, IF SSR IS SET
6661 075262					DELAY	250	;DELAY ABOUT .25 SEC
075262	012727	000250					
075266	000000						
075270	013727	002116'					
075274	000000						
075276	005367	177772					
075302	001375						
075304	005367	177756					
075310	001367						
6662 075312	005337	075474'			DEC	T39DLY	;START DELAY COUNT DOWN
6663 075316	001352				BNE	222\$	;BR, IF COUNTER IS NOT AT DONE
6664 075320	004737	016146'		225\$:	JSR	PC,CHKTSSR	;WAIT FOR SSR
6665 075324	103405				BCS	230\$	;BR, IF NO ERROR
6666 075326	010001				MOV	R0,R1	;ERROR, SAVE TSSR
6670 075330					ERRHRD	ERRNO,T39NBA,PKTSSR	;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
075330	104456						
075332	002124						
075334	076755'						
075336	011656'						
6671 075340				230\$:	CKLOOP		;LOOP IF SELECTED
075340	104406						
6672 075342	023727	075616'	125252		CMP	T39BFR+100,0125252	;DID LOC GET OVER WRITTEN
667 075350	001013				BNE	240\$	;BR, IF IT DIDN'T GET ETC.
6674 075352					PRINTX	0T39ETN,UNITN	; "DRIVE DOESN'T HAVE EXT TAPE STATUS
075352	013746	002202'					
075356	012746	076421'					
075362	012746	000002					
075366	010600						
075370	104415						
075372	062706	000006					
6675 075376	000412						
6676 075400				240\$:	BR	250\$	;SKIP OVER
075400	013746	002202'			PRINTX	0T39ETS,UNITN	; "DRIVE HAS EXT TAPE STATUS"
075404	012746	076332'					
075410	012746	000002					
075414	010600						
075416	104415						
0 020	062706	000006					
6677 075424	005237	002202'		250\$:	INC	UNITN	;BUMP DRIVE NUMBER
6678 075430	023727	002202'	000003		CMP	UNITN,03	;AT END OF DRIVES YET
6679 075436	001402				REQ	63\$	;BR

Address	Instruction	Comment	Register	Value	Operation	Register	Value
075454	010600				MOV	SP, R0	
075456	104415				TRAP	C\$PN1X	
075460	062706	000004			ADD	04, SP	
6682 075464	000137	000200					
6683 075470			64\$:	JMP	200		
075470	104432			EXIT	TST		
075472	001736						
6684							
6685							
6686							
6687							
6688							
6689							
6690							
6691 075474	000000		T39DLY:	.WORD	0		
6693 075476				.BLKB	10- < . - TSV2&7>		
6695 075500			T39PACKET:				
6696 075500	140006			.WORD	140006		
6697 075502	075510			.WORD	T39TAD		
6698 075504	000000			.WORD	0		
6699 075506	000012			.WORD	10.		
6700 075510			T39TAD:				
6701 075510	000		T39B50:	.BYTE	0		
6702 075511	000		T39B51:	.BYTE	0		
6703 075512	000000		T39B52:	.WORD	0		
6704 075514	000000			.WORD	0		
6705 075516			T39BFR:	.BLKB	150.		
6706							
6707							
6709 076172				.BLKB	10- < . - TSV2&7>		
6711 076200			T39PK2:				
6712 076200	140004			.WORD	140004		
6713 076202	076210			.WORD	T39DTA		
6714 076204	000000			.WORD	0		
6715 076206	000012			.WORD	10.		
6716							
6717							
6718 076210			T39DTA:				
6719 076210	075516			.WORD	T39BFR		
6720 076212	000000			.WORD	0		
6721 076214	000400			.WORD	256.		
6722 076216	000000		T39EA1:	.WORD	0		
6723 076220	000000		T39DSW:	.WORD	0		
6725 076222				.BLKB	10- < . - TSV2&7>		
6727 076230	140012		T39PK3:	.WORD	140012		
6728 076232	000000			.WORD	0		
6729							
6730							
6731							
6733 076234				.BLKB	10- < . - TSV2&7>		
6735 076240	140005		T39PK4:	.WORD	140005		
6736 076242	000000		T39WR:	.WORD	0		
6737 076244	000000			.WORD	0		
6738 076246	000400		T39SI2:	.WORD	256.		
6739							
6740							
6741							

K2

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 11: CONFIGURATION TYPEOUT

SEQ 230

```

6742
6743
6744          ;*
6745          ;LOCAL TEXT MESSAGES FOR TEST
6746          ;*
6747
6748
6749
6750 076250      045      116      000  T39NFL: .ASCIZ  'N'
6751 076253      123      164      141  T39NE: .ASCIZ  'Stand-alone Configuration Typeout Not Executed'
6752 076332      045      116      045  T39ETS: .ASCIZ  'N#A Extended Tape Status Available, Drive Number #D2'
6753 076421      045      116      045  T39ETN: .ASCIZ  'N#A Extended Tape Status NOT Available, Drive Number #D2'
6754 076514      045      116      045  T39OF2: .ASCIZ  'N#A Drive Number #D2#A Is Off-Line'
6755 076560      045      116      045  T39ON2: .ASCIZ  'N#A Drive Number #D2#A Is On-Line'
6756 076623      045      116      045  T39WRT: .ASCIZ  'N#A Drive Number #D2#A Is Write Protected'
6757 076676      045      116      045  T39WPN: .ASCIZ  'N#A Drive Number #D2#A Is NOT Write Protected'
6758 076755      127      122      111  T39NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
6759 077031      103      157      156  T39SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
6760
6761 077121      045      116      045  T39SFS: .ASCIZ  'N#A State Of Extended Features Switch  ='
6762 077173      045      116      045  T39SBS: .ASCIZ  'N#A State Of Buffering Switch      ='
6763 077245      045      101      040  T39OFF: .ASCIZ  'A OFF'
6764 077254      045      101      040  T39ON: .ASCIZ  'A ON'
6765 077263      045      116      045  T39MCL: .ASCIZ  'N#A M7455 Microcode Revision Level  =#G2'
6766
6767 077340      000000      T39RL: .WORD  0
6768
6769
6770
6771
6772          ;*
6773          ;LOCAL STORAGE FOR THIS TEST
6774          ;*
6775 077342      000000      T39DAT: .WORD  0          ;LOGICAL RESPONSE TO QUESTION
6776 077344
6777 077344      T39REST:
6778 077350      012701      075506'      SAVREG
6779 077354      012721      140006      MOV      #T39PACKET,R1      ;SAVE THE REGISTERS
6780 077360      012721      075510'      MOV      #140006,(R1)+      ;START OF THE PACKET
6781 077364      005021      MOV      #T39TAD,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK,CVC*1
6782 077366      012721      000006      CLR      (R1)+      ;ADDRESS OF DATA BLOCK
6783 077372      005021      MOV      #6,(R1)+      ;EXTENDED ADDRESS
6784 077374      005021      CLR      (R1)+      ;SIZE OF DATA BLOCK IN BYTES
6785 077376      005011      CLR      (R1)+      ;CLEAR BSEL0 AND BSEL1
6786 077400      000207      CLR      (R1)+      ;CLEAR SEL2
6787
6788
6789
6790          ;*
6791          ;LOCAL TEXT MESSAGES FOR TEST
6792          ;*
6792 077402      103      157      156  TST39ID: .ASCIZ  'Configuration Typeout'
6793
6794 077430      .EVEN
6794 077430      ENDTST
6795
6796
6795
6796          .SBTTL  TEST 12: SCOPE LOOPS

```

L10076: TRAP C\$ETST

TSV5 - HARDWARE TESTS MACRO M1113 01-FEB-84 17:02
TEST 12: SCOPE LOOPS

SEQ 232

Address	Offset	Hex	Label	Instruction	Comment
077504	011644				.WORD SFIMSG
6856	077506	012700	100164	5\$: MOV	#SCMENU,R0 ;MENU OF SCOPE LOOP SELECTIONS
6857	077512	012701	000010	MOV	#8.,R1 ;MAXIMUM ALLOWED SELECTION
6858	077516	004737	020076	JSR	PC,GETSEL ;GO GET THE OPERATORS SELECTION
6859	077522	005700		TST	R0 ;WAS ZERO SPECIFIED ?
6860	077524	001760		BEQ	2\$;REPEAT MENU IF YES.
6861	077526	020027	000007	CMP	R0,#7 ;EXTENDED TSSR ?
6862	077532	001015		BNE	3\$;BRANCH IF NOT
6863	077534	005737	002226	TST	EXTFEA ;CHECK FOR EXTENDED FEATURE SET
6864	077540	001012		BNE	3\$;BR, IF IT IS ON
6865	077542			PRINTF	#EXFMSG ;WARN OPERATOR EXTENDED FEATURES CLEAR
	077542	012746	100747		MOV #EXFMSG,-(SP)
	077546	012746	000001		MOV #1,-(SP)
	077552	010600			MOV SP,R0
	077554	104417			TRAP C\$PNTF
	077556	062706	000004		ADD #4,SP
6866	077562	000137	077466		
6867	077566	010004		3\$: JMP	2\$;GO BACK TO BASIC MENU
6868	077570			MOV	R0,R4 ;SAVE THE MENU SELECTION
	077570	012700	000340	SETPRI	#PRI07 ;RAISE THE PRIORITY
	077574	104441			MOV #PRI07,R0
6869	077576	005037	100156		TRAP C\$SPRI
6870	077602	032737	000100	177560	CLR TTION ;ASSUME INTERRUPTS ARE ENABLED
6871	077610	001005		BIT	#100,#TTICSR ;ARE TTI INTERRUPTS ON ?
6872	077612	005237	100156	BNE	4\$;BRANCH IF YES
6873	077616	052737	000100	177560	INC TTION ;FLAG SET IF INTERRUPTS OFF
6874	077624	012701	000060	4\$: BIS	#100,#TTICSR ;ENABLE INTERRUPTS
6875	077630	011137	100160	MOV	#TTIVEC,R1 ;START OF TTI VECTORS
6876	077634	012721	100060	MUV	(R1),TVECSAV ;SAVE THE CURRENT TTI VECTOR
6877	077640	011137	100162	MOV	#60,(R1) ;SET NEW INTERRUPT ROUTINE
6878	077644	012711	000340	MOV	(R1),TPRISAV ;SAVE THE VECTOR PRIORITY
6879	077650			MOV	#PRI07,(R1) ;USE PRIORITY SEVEN
	077650	012700	000000	SETPRI	#PRI00 ;LOWER INTERRUPT BR LEVEL
	077654	104441			MOV #PRI00,R0
6880	077656	006304			TRAP C\$SPRI
6881	077660	000174	077664		
6882	077664	077466		6\$: ASL	R4 ;CONVERT TO WORD OFFSET
6883	077666	077706		JMP	#6\$(R4) ;JUMP TO PROPER LOOP
6884	077670	077716		.WORD	2\$;RETYPE THE MENU
6885	077672	077730		.WORD	10\$;TSBA READ ACCESS
6886	077674	077750		.WORD	15\$;TSSR READ ACCESS
6887	077676	077774		.WORD	20\$;TSSR WRITE ACCESS
6888	077700	100020		.WORD	25\$;TSDB HIGH BYTE WRITE ACCESS
6889	077702	100040		.WORD	30\$;TSDB LOW BYTE WRITE ACCESS
6890	077704	100152		.WORD	40\$;TSDB MAINTENANCE MODE
6891				.WORD	65\$;TSDBX WRITE ACCESS
6892					;LEAVE THE TEST
6893	077706	105065	000000	10\$: CLRB	TSDB(R5) ;ENTER MAINTENANCE MODE
6894	077712	011500		12\$: MOV	(R5),R0 ;READ TSBA REGISTER
6895	077714	000776		BR	12\$;LOOP UNTIL HALTED
6896					
6897					
6898	077716	012703	000002	15\$: MOV	#TSSR,R3 ;ADDRESS OF TSSR REGISTER
6899	077722	060503		ADD	R5,R3 ;POINT TO TSV05'S REGISTERS
6900	077724	011300		18\$: MOV	(R3),R0 ;READ TSSR REGISTER
6901	077726	000776		BR	18\$;LOOP UNTIL STOPPED
6902					

N2

TSV5 - HARDWARE TESTS
TEST 12: SCOPE LOOPS

MACRO M1113 01-FEB-84 17:02

SEQ 233

6903	077730	004737	020014'	20\$:	JSR	PC,GETPAT	;READ THE DATA PATTERN
6904	077734	010001			MOV	R0,R1	;DATA PATTERN FOR LOOP
6905	077736	012703	000002		MOV	#TSSR,R3	;ADDRESS OF TSSR
6906	077742	060503			ADD	R5,R3	;POINT TO TSV05'S REGISTERS
6907	077744	010113		22\$:	MOV	R1,(R3)	;WRITE DATA TO TSSR
6908	077746	000776			BR	22\$	;LOOP
6909							
6910							
6911	077750	105065	000000	25\$:	CLRB	TSDB(R5)	;ENTER MAINTENANCE MODE
6912	077754	004737	020014'		JSR	PC,GETPAT	;READ THE DATA PATTERN
6913	077760	010001			MOV	R0,R1	;DATA PATTERN FOR LOOP
6914	077762	012703	000001		MOV	#TSDBH,R3	;ADDRESS OF HIGH BYTE OF TSDB
6915	077766	060503			ADD	R5,R3	;POINT TO TSV05'S REGISTERS
6916	077770	110113		27\$:	MOVB	R1,(R3)	;WRITE THE DATA TO TSDB, HIGH BYTE
6917	077772	000776			BR	27\$	;LOOP UNTIL STOPPED
6918							
6919							
6920	077774	105065	000000	30\$:	CLRB	TSDB(R5)	;ENTER MAINTENANCE MODE
6921	100000	004737	020014'		JSR	PC,GETPAT	;READ THE DATA PATTERN
6922	100004	010001			MOV	R0,R1	;DATA PATTERN FOR LOOP
6923	100006	012703	000000		MOV	#TSDB,R3	;ADDRESS OF TSSR
6924	100012	060503			ADD	R5,R3	;POINT TO TSV05'S REGISTERS
6925	100014	110113		32\$:	MOVB	R1,(R3)	;WRITE DATA TO TSSR, LOW BYTE
6926	100016	000776			BR	32\$	;LOOP UNTIL HALTED BY OPERATOR
6927							
6928	100020	004737	020014'	35\$:	JSR	PC,GETPAT	;READ THE DATA PATTERN
6929	100024	010001			MOV	R0,R1	;DATA PATTERN FOR LOOP
6930	100026	012703	000000		MOV	#TSDB,R3	;SELECT TSDB
6931	100032	060503			ADD	R5,R3	;POINT TO TSV05'S REGISTERS
6932	100034	010113		37\$:	MOV	R1,(R3)	;WRITE THE DATA PATTERN
6933							
6934	100036	000776			BR	37\$	;LOOP UNTIL HALTED
6935							
6936	100040	004737	020014'	40\$:	JSR	PC,GETPAT	;READ THE DATA PATTERN
6937	100044	010001			MOV	R0,R1	;SAVE THE DATA PATTERN
6938	100046	012703	000003		MOV	#TSSRH,R3	;BYTE ADDRESS OF TSSR, HIGH BYTE
6939	100052	060503			ADD	R5,R3	;POINT TO TSV05'S REGISTERS
6940	100054	110113		42\$:	MOVB	R1,(R3)	;WRITE THE DATA TO REGISTER
6941	100056	000776			BR	42\$	;LOOP UNTIL HALTED
6942							
6943							
6944							
6945							
6946							
6947							
6948	100060	010046		60\$:	MOV	R0,(SP)	;SAVE WORK REGISTER
6949	100062	113700	177562		MOVB	#TTIBR,R0	;GET THE OPERATOR INPUT
6950	100066	042700	000200		BIC	#200,R0	;STRIP OFF PARITY BIT
6951	100072	122700	000015		CMPB	#15,R0	;IS IT A CARRIAGE RETURN ?
6952	100076	001021			BNE	61\$	;JUST EXIT IF NOT
6953	100100	012765	077466' 000002		MOV	#2\$(SP)	;RETURN TO MASTER MENU
6954	100106	005066	000004		CLR	4(SP)	;FORCE PRIORITY ZERO
6955	100112	013737	100160' 000060		MOV	TVECSAV,#TTIVEC	;RESTORE SUPERVISOR VECTOR
6956	100120	013737	100162' 000062		MOV	TPRISAV,#TTIVEC+2	;RESTORE SUPERVISOR PRIORITY
6957	100126	005737	100156'		IST	TTION	;ARE SUPERVISOR INTERRUPTS ENABLED ?
6958	100132	001403			BEQ	61\$	;BRANCH IF YES
6959	100134	042737	000100 177560		BIC	#100,#TTICSR	;TURN OFF TTI INTERRUPTS

B3

TSV5 - HARDWARE TESTS
TEST 12: SCOPE LOOPS

MACRO M1113 01-FEB-84 17:02

SEQ 234

```

6960 100142 012600      61:  MOV    (SP)+,R0      ;RESTORE REGISTER
6961 100144 000002      61:  RTI              ;RETURN FROM INTERRUPT
6962
6963 100146
6964 100146      64:
6964 100146      63:  EXIT    TST              ;EXIT THE TEST
        100146 104432
        100150 000736
6965 100152 000137 000200      65:  JMP     200              ;RETURN TO SUPERVISOR
6966
6967
6968
6969
6970
6971 100156 000000      TTION:      .WORD    0      ;WORD SET IF SUPERVISOR TTI INTER OFF
6972 100160 000000      TVECSAV:   .WORD    0      ;SAVE TTI VECTOR
6973 100162 000000      TPRISAV:   .WORD    0      ;SAVE TTI PRIORITY
6974
6975
6976
6977
6978
6979
6980
6981 100164 100216' 100271' 100317' 6981: .EVEN
6982 100200 100470' 100526' 100574' 6982: .WORD    1,2,3,4,5,6
6983
6984
6985 100216      012      123      105 6985: .ASCIZ  <12>'SELECT SCOPE LOOP FROM FOLLOWING OPTIONS:'
6986 100271      012      011      060 6986: .ASCIZ  <12>' 0      Display This Menu'
6987 100317      011      061      011 6987: .ASCIZ  ' 1      TSBA Read Access'
6988 100343      011      062      011 6988: .ASCIZ  ' 2      TSSR Read Access'
6989 100367      011      063      011 6989: .ASCIZ  ' 3      Initialize (TSSR Write Access)'
6990 100431      011      064      011 6990: .ASCIZ  ' 4      TSDB High Byte Write Access'
6991 100470      011      065      011 6991: .ASCIZ  ' 5      TSDB Low Byte Write Access'
6992 100526      011      066      011 6992: .ASCIZ  ' 6      TSDB Maintenance Mode Write Access'
6993 100574      011      067      011 6993: .ASCIZ  ' 7      TSDBX (TSSR High Byte) Write Access'
6994 100643      011      070      011 6994: .ASCIZ  ' 8      Return to Diagnostic Supervisor'
6995 100706      000
6996 100707      124      171      160 6996: .ASCIZ  'Type RETURN To Stop Scope Loops'
6997 100747      045      116      045 6997: .ASCIZ  '*** Extended Features Switch Not On ***'
6998 101025      123      164      141 6998: .ASCIZ  'Stand-alone Scope Loops Not Executed'
6999 101072      123      143      157 6999: .ASCIZ  'Scope Loops'
7000
7001 101106
7001 101106
7001 101106 104401
7002 101110
7003

```

L10077: TRAP C:ETST

ENDMOD

C3

TSV6 - PARAMETER CODING MACRO M1113 01-FEB-84 17:02
TEST 12: SCOPE LOOPS

SEQ 235

```

1          .TITLE   TSV6 - PARAMETER CODING
7
12
18
19 101110          BGNMOD   TSV6
20 101110          TSV6::
21
22          .SBTTL   HARDWARE PARAMETER CODING SECTION
23
24          ;**
25          ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
26          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
27          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
28          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
29          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
30          ; WITH THE OPERATOR.
31          ;**
32 101110          BGNHRD
33 101110          .WORD   L10100-L$HARD/2
34 101112          L$HARD::
35 101112          GPRMA   HPM1,0,0,160010,177776,YES          ;GET TSBA/TSD8 REGISTER ADDRESS.
36 101112          .WORD   T$CODE
37 101112          .WORD   HPM1
38 101112          .WORD   T$LOLIM
39 101112          .WORD   T$HILIM
40 101112          GPRMA   HPM2,2,0,0,776,YES                ;GET VECTOR ADDRESS.
41 101112          .WORD   T$CODE
42 101112          .WORD   HPM2
43 101112          .WORD   T$LOLIM
44 101112          .WORD   T$HILIM
45 101112          GPRMA   HPM3,4,0,340,0,7,YES              ;GET INTERRUPT PRIORITY.
46 101112          .WORD   HPM3
47 101112          .ENDHRD
48 101112          .EVEN
49 101112          L10100:
50 101112          104      105      126      HPM1:  .ASCIZ  'DEVICE ADDRESS (TSBA/TSD8)'
51 101112          111      116      124      HPM2:  .ASCIZ  'INTERRUPT VECTOR'
52 101112          111      116      124      HPM3:  .ASCIZ  'INTERRUPT PRIORITY'
53 101112          .EVEN
54
55          .SBTTL   SOFTWARE PARAMETER CODING SECTION
56
57          ;**
58          ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
59          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
60          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
61          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
62          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
63          ; WITH THE OPERATOR.
64          ;**
65 101242          BGNSFT
66 101242          .WORD   L10101-L$SOFT/2
67 101244          L$SOFT::
68 101244          GPRML   SPM1,0,-1,YES                      ; GET TRANSPORT TEST FLAG.
69 101244          GPRML   SPM1,2,-1,YES                      ; GET ITERATION CONTROL.
70 101244          .WORD   T$CODE

```

SP)

SP)

D3

TSV6 - PARAMETER CODING MACRO M1117 01-FEB-84 17:02
SOFTWARE PARAMETER CODING SECTION

SEQ 236

```

      101246 101302'
      101250 177777
56      ;      .WORD  SPM4
57      ;      .WORD  -1
58 101252      ;      GPRMD  SPM6,4,D,7777,0,7777,YES      ; GET LOCAL ERROR LIMIT
      ;      GPRMD  SPM7,6,D,7777,0,7777,YES      ; GET GLOBAL ERROR LIMIT
      .ENDSFT
      .EVEN
      101252      L10101:
59
60
61 101252      105      116      101  SPM1:  .ASCIZ  'ENABLE TRANSPORT TESTS '
62 101302      111      116      110  SPM4:  .ASCIZ  'INHIBIT ITERATIONS '
63      ;SPM6:  .ASCIZ  'PER TEST ERROR LIMIT '
64      ;SPM7:  .ASCIZ  'PER UNIT ERROR LIMIT '
65      .SBTTL  PATCH AREA
66
67      ;
68      ; FINALLY A GENEROUS PATCH AREA.
69      ;
70      ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
71      ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
72      ;
73
74 101332      PATCH::
75
76      ;      .BLKW  32.
77 101332      ;      .BLKW  1.
78
79      ;      .IF  NZ,..6377
80      ;      .-..1377+1
81      ;      .ENDC
82 101334      LASTAD      ;SLT LAST USED ADDRESS.
      .EVEN
      .WORD  0
      .WORD  0
      L$LAST::
      ENOMOD
      .SBTTL  HARD CODED P-TABLE
83
84      ;++
85      ;
86      ; DIAGNOSTIC IS PRE-PARAMETERIZED PER THIS TABLE
87      ;--
88 101340      BGNSETUP      1
89 101340      BGNPTAB
      .WORD  0
      .WORD  L10104-./2-1
      L10102:
      .WORD  172522
      .WORD  224
      .WORD  PRI05
      ENDP1AB
      L10104:
      ENOSETUP
90
91
92
93
94 101352
95
96      000001
      .END

```

E3

TSV6 - PARAMETER CODING MACRO M1113 01-FEB-84 17:02
SYMBOL TABLE

SEQ 237

ADDSSR 011736RG	002 C\$AU = 000052	DEVDR0 023146R	002 FREEHI 003130R	002 INTCPC 015760R	002
ADR = 000020 G	C\$AUTO = 000061	DEVNRD 023065R	002 FRESIZ 003126RG	002 INTFLA 015755R	002
AMBTSS 006447R	002 C\$BRK = 000022	DEVNXR 023003R	002 FUSI 004113R	002 INTMAS 015754R	002
ASSEMB = 000010	C\$BSEG = 000004	DEVOML 022733R	002 F\$AU = 000015	INTR 016026RG	002
A1716 = 000003	C\$BSUB = 000002	DEVSUM 022676R	002 F\$AUTO = 000020	INTREC 002224RG	002
BADDAT 003156RG	002 C\$CEFG = 000045	DFPTBL 002156RG	002 F\$BGN = 000040	INTVEC 015756R	002
BADSSR 015510RG	002 C\$CLCK = 000062	DIAGMC = 000000	F\$CLEA = 000007	INTX 004274R	002
BDVPCR = 177520 G	C\$CLEA = 000012	DICEB = 000001	F\$DU = 000016	INVERT 020734RG	002
BENBSW 002230RG	002 C\$CLOS = 000035	DSBINT 016014R	002 F\$END = 000041	IOKCKI = 000200	
BIE = 040000	C\$CLP1 = 000006	DUAD12 004637R	002 F\$HARD = 000004	IOKSTP = 000001	
BIT0 = 000001 G	C\$CVEC = 000036	DUFLG 003112RG	002 F\$HW = 000013	IPRI 002212RG	002
BIT00 = 000001 G	C\$DCLN = 000044	DUMMY 003062R	002 F\$INIT = 000006	ISR = 000100 G	
BIT01 = 000002 G	C\$DODU = 000051	EF.CON = 000036 G	F\$JMP = 000050	IVEC 002210RG	002
BIT02 = 000004 G	C\$DRPT = 000024	EF.NEW = 000035 G	F\$MOD = 000000	IXE = 004000 G	
BIT03 = 000010 G	C\$DU = 000053	EF.PWR = 000034 G	F\$MSG = 000011	I\$AU = 000041	
BIT04 = 000020 G	C\$EDIT = 000003	EF.RES = 000037 G	F\$PROT = 000021	I\$AUTO = 000041	
BIT05 = 000040 G	C\$ERDF = 000055	EF.STA = 000040 G	F\$PWR = 000017	I\$CLN = 000041	
BIT06 = 000100 G	C\$ERHR = 000056	EMAXDU 016611R	002 F\$RPT = 000011	I\$DU = 000041	
BIT07 = 000200 G	C\$ERR0 = 000060	EN = 000000	F\$SEG = 000003	I\$HRD = 000041	
BIT08 = 000400 G	C\$ERSF = 000054	ENAINI 015762R	002 F\$SOFT = 000005	I\$INIT = 000041	
BIT09 = 001000 G	C\$ERS0 = 000057	ENVIRN 020450R	002 F\$SRV = 000010	I\$MOD = 000041	
BIT1 = 000002 G	C\$ESCA = 000010	EPRTSW 002200RG	002 F\$SAB = 000002	I\$MSG = 000041	
BIT10 = 002000 G	C\$ESEG = 000005	EPRT1 006172R	002 F\$SW = 000014	I\$PROT = 000040	
BIT11 = 004000 G	C\$ESUB = 000003	EPRT2 006172R	002 F\$TEST = 000001	I\$PTAB = 000041	
BIT12 = 010000 G	C\$ETST = 000001	ERCM 011543R	002 GDDAT 003160RG	002 I\$PWR = 000041	
BIT13 = 020000 G	C\$EXIT = 000032	ERRHI 002236RG	002 GERRMA 002174RG	002 I\$RPT = 000041	
BIT14 = 040000 G	C\$GETB = 000026	ERRK 016570R	002 GETPAT 020015RG	002 I\$SEG = 000041	
BIT15 = 100000 G	C\$GETW = 000027	ERRLO 002240RG	002 GETSEL 020076RG	002 I\$SETU = 000041	
BIT2 = 000004 G	C\$GMAN = 000043	ERRNO = 002261	G\$CNT0 = 000200	I\$SFT = 000041	
BIT3 = 000010 G	C\$GPHR = 000042	ERRVEC = 000004 G	G\$DELM = 000372	I\$SRV = 000041	
BIT4 = 000020 G	C\$GPL0 = 000030	ERTABE 003376R	002 G\$DISP = 000003	I\$SUB = 000041	
BIT5 = 000040 G	C\$GPRI = 000040	ERTABL 003176R	002 G\$EXCP = 000400	I\$TST = 000041	
BIT6 = 000100 G	C\$INIT = 000011	ESUM 016572R	002 G\$HILI = 000002	J\$JMP = 000167	
BIT7 = 000200 G	C\$INLP = 000020	EVL = 000004 G	G\$LOXI = 000001	KIPAR0 = 172340	
BIT8 = 000400 G	C\$MANI = 000050	EXBCNT = 000010	G\$NO = 000000	KIPAR1 = 172342	
BIT9 = 001000 G	C\$MEN = 000031	EXFMSG 100747R	002 G\$OFFS = 000400	KIPAR2 = 172344	
BOE = 000400 G	C\$MSG = 000023	EXPBRE 015312RG	002 G\$OFSI = 000376	KIPAR3 = 172346	
BRINIT 004453R	002 C\$OPEN = 000034	EXP0 002232RG	002 G\$PRMA = 000001	KIPAR4 = 172350	
BSEL0 = 000000	C\$PNTB = 000014	EXPGOT 004527R	002 G\$PRMD = 000002	KIPAR5 = 172352	
BSEL1 = 000001	C\$PNTF = 000017	EXPGT2 004563R	002 G\$PRML = 000000	KIPAR6 = 172354	
CHKAMB 015654R	002 C\$PNTS = 000016	EXPMSG 002322RG	002 G\$RADA = 000140	KIPAR7 = 172356	
CHKMAN 020320RG	002 C\$PNTX = 000015	EXPREC 015304RG	002 G\$RAD8 = 000000	KIPDR0 = 172300	
CHKTSS 016146R	002 C\$QIO = 000377	EXTA 005604R	002 G\$RADD = 000040	KIPDR1 = 172302	
CKDROP 017014R	002 C\$RDBU = 000007	EXTEND 005602R	002 G\$RADL = 000120	KIPDR2 = 172304	
CKEMAX 016714R	002 C\$REFG = 000047	EXTFEA 002226RG	002 G\$RADO = 000020	KIPDR3 = 172306	
CKMSG 011170RG	002 C\$RESE = 000033	E\$END = 002100	G\$XFER = 000004	KIPDR4 = 172310	
CKMSG2 011310RG	002 C\$REVI = 000003	E\$LOAU = 000035	G\$YES = 000010	KIPDR5 = 172312	
CKRAM 010724RG	002 C\$RFLA = 000021	FATERR = 000060	HIADDR = 001400	KIPDR6 = 172314	
CKRAM2 011034RG	002 C\$RPT = 000025	FATFLG 002222RG	002 HOE = 100000 C	KIPDR7 = 172316	
CHDPKT 021017RG	002 C\$SEFG = 000046	FERCM 011532R	002 HPM1 101132R	002 K TENAB 003134RG	002
CHPMEM 017500R	002 C\$SPRI = 000041	FIFEXP 012000RG	002 HPM2 101166R	002 KTF LG 003132RG	002
CONFIG 017062R	002 C\$SVEC = 000037	FIF1MS 012052R	002 HPM3 101212R	002 K TINIT 020536R	002
COUNT 002310RG	002 C\$TPRI = 000013	FIF2MS 012121R	002 IBE = 010000 G	KTOFF 017106R	002
CSRADD 002206RG	002 DATA 002312RG	FILLME 017234R	002 IDU = 000040 G	KTON 017070R	002
CTAB 003164RG	002 DATASC 020052R	FNOINT 004211R	002 IER = 020000 G	LERRMA 002172RG	002
CTABE 003176RG	002 DEBUGM 011442R	FORCER 002176RG	002 IFALT 004252R	002 LERRNO = 000000	
CTABM 003164RG	002 DEV CNT 002220RG	002 FREE 003124RG	002 INCERK 016656R	002 LISTAL = 000001	

F3

TSV6 - PARAMETER CODING MACRO M1113 01-FEB-84 17:02
SYMBOL TABLE

SEQ 238

LOE	= 040000 G	L\$UNIT	002012RG	002 L10071	056250R	002 NXTU	021572R	002 PRMESS	014052R	002
LOOPCN	002216RG	002 L10000	002164R	002 L10072	057514R	002 OFL	= 000100	PRMNO	002320RG	002
LOOPCO	012736R	002 L10001	002176R	002 L10073	066414R	002 ONEFIL	= 000000	PRMSGC	014362RG	002
LOOPFL	003162RG	002 L10002	005600R	002 L10074	064544R	002 O\$APTS	= 000000	PRMSG0	014542R	002
LOT	= 000010 G	002 L10003	011654R	002 L10075	074256R	002 O\$AU	= 000001	PRMSG1	014607R	002
L\$ACP	002110RG	002 L10004	011672R	002 L10076	077430R	002 O\$BGNR	= 000001	PRMSG2	014645R	002
L\$APT	002036RG	002 L10005	011710R	002 L10077	101106R	002 O\$BGNS	= 000001	PROASC	014230R	002
L\$AU	022122RG	002 L10006	011716R	002 L10100	101132R	002 O\$DU	= 000001	PR1ASC	014275R	002
L\$AUT	002070RG	002 L10007	011734R	002 L10101	101252R	002 O\$ERRT	= 000000	PST32W	003152RG	002
L\$AUTO	022326RG	002 L10010	011752R	002 L10102	101344R	002 O\$GNSW	= 000001	PUNIT	022054R	002
L\$CCP	002106RG	002 L10011	011776R	002 L10104	101352R	002 O\$POIN	= 000001	PW.D11	= 000021	
L\$CLEA	022406RG	002 L10012	012050R	002 MEMADD	013564RG	002 O\$SETU	= 000000	PW.D13	= 000022	
L\$CO	002032RG	002 L10013	012220R	002 MEMCK	021026RG	002 PASRPT	021624R	002 PW.D22	= 000020	
L\$DEPO	002011RG	002 L10014	012734R	002 MENASC	020267R	002 PATCH	101332RG	002 PW.NOP	= 000000	
L\$DESC	003410RG	002 L10015	013562R	002 MENERR	020214R	002 PATDAT	020050R	002 PW.N01	= 000023	
L\$DESP	002076RG	002 L10016	013604R	002 MENRES	020316R	002 PC.ERA	= 002400	PW.RDE	= 000024	
L\$DEVP	002060RG	002 L10017	015310R	002 MIMENU	073142R	002 PC.IER	= 002000	PW.RDR	= 000001	
L\$DISP	002124RG	002 L10020	015316R	002 MMRO	= 170200	PC.N00	= 001000	PW.RDS	= 000005	
L\$DLY	002116RG	002 L10021	015324R	002 MMVEC	= 000250	PC.REL	= 000000	PW.RFI	= 000003	
L\$DTP	002040RG	002 L10022	015336R	002 MSA.FR	= 000006	PC.REW	= 000400	PW.WCT	= 000006	
L\$DTYP	002034RG	002 L10023	015360R	002 MSA.NO	= 000000	PKBCNT	= 000006	PW.WFI	= 000004	
L\$DU	022220RG	002 L10024	015406R	002 MSA.NR	= 000004	PKHI	= 000004	PW.WFM	= 000007	
L\$DUT	002072RG	002 L10025	015546R	002 MSA.VO	= 000002	PKLOW	= 000002	PW.WMI	= 000010	
L\$DVTY	003402RG	002 L10026	016056R	002 MSGEXP	011754RG	002 PKTADD	007366R	002 PW.WNP	= 000011	
L\$EF	002052RG	002 L10030	022052R	002 MSGLOO	012674RG	002 PKTFRM	007330R	002 PW.WTR	= 000002	
L\$ENVI	002044RG	002 L10031	022216R	002 MSGSTA	012160RG	002 PKTGET	011674RG	002 P.ACK	= 000000	
L\$ETP	002102RG	002 L10032	022324R	002 MSGSUB	013552RG	002 PKTIES	011720RG	002 P.CMD	= 000037	
L\$EXP1	002046RG	002 L10033	022404R	002 MS.ATI	= 000006	PKTRAM	004741RG	002 P.CONT	= 000012	
L\$EXP4	002064RG	002 L10034	022432R	002 MS.EXT	= 000200	PKTSSR	011656RG	002 P.CVC	= 040000	
L\$EXP5	002066RG	002 L10035	022674R	002 MS.FSD	= 000001	PNT	= 001000 G	P.FMT	= 000140	
L\$HARD	101112RG	002 L10036	024204R	002 MS.RSF	= 000020	PRAMPK	013606R	002 P.FORM	= 000011	
L\$HIME	002120RG	002 L10037	026176R	002 MS.RST	= 000010	PRASC	014333R	002 P.GETS	= 000017	
L\$HPCP	002016RG	002 L10040	024460R	002 NBA	= 002000	PRBEXP	015300R	002 P.IE	= 000200	
L\$HPTP	002022RG	002 L10041	024724R	002 NEWPAS	021560R	002 PRBMSG	015146R	002 P.INIT	= 000013	
L\$HW	002156RG	002 L10042	031522R	002 NODEV	003114RG	002 PRBREC	015302R	002 P.MODE	= 007400	
L\$ICP	002104RG	002 L10043	026552R	002 NOEXT	027716R	002 PRBTOT	015233R	002 P.OPP	= 020000	
L\$INIT	021326RG	002 L10044	027042R	002 NOINI	004331R	002 PRBYTE	014732RG	002 P.POSI	= 000010	
L\$LADP	002026RG	002 L10045	027340R	002 NOINTR	004215R	002 PRI	= 002000 G	P.READ	= 000001	
L\$LAST	101340RG	002 L10046	027722R	002 NOIIS	002170RG	002 PRIADD	007772R	002 P.SWB	= 010000	
L\$LOAD	002100RG	002 L10047	034312R	002 NOMAN	020354R	002 PRIA0	010042R	002 P.WRIT	= 000005	
L\$LUN	002074RG	002 L10050	032056R	002 NOMEM	005454R	002 PRIBX0	007424RG	002 P.WRTC	= 000004	
L\$MREV	002050RG	002 L10051	032450R	002 NP.IR	= 000200	PRIEQU	007672R	002 P.WRTS	= 000006	
L\$NAME	002000RG	002 L10052	033056R	002 NP.LCO	= 000040	PRIPKT	007202RG	002 QVP	002204RG	002
L\$PRIO	002042RG	002 L10053	040114R	002 NP.OUY	= 000100	PRIRAM	007700R	002 RAMASC	013766R	002
L\$PROT	021316RG	002 L10054	035376R	002 NP.WRP	= 000020	PRITAD	010106R	002 RAMDAT	002242RG	002
L\$PRT	002112RG	002 L10055	036330R	002 NSI	004146R	002 PRITSS	005636R	002 RAMERR	015320RG	002
L\$REPP	002062RG	002 L10056	050226R	002 NSINIT	004403R	002 PRITO	010170R	002 RAMEXP	015340RG	002
L\$REV	002010RG	002 L10057	040420R	002 NUL	004523R	002 PRITI	010233R	002 RAMFOR	007730R	002
L\$RPT	022434RG	002 L10060	041630R	002 NULCR	004524R	002 PRIXOR	007554RG	002 RAMSIZ	002302RG	002
L\$SOFT	101244RG	002 L10061	043170R	002 NXM	= 004000	PRI00	= 000000 G	RAMTAD	015326RG	002
L\$SPC	002056RG	002 L10062	043546R	002 NXMFLG	003116RG	002 PRI01	= 000040 G	RCVHIA	002304RG	002
L\$SPCP	002020RG	002 L10063	045020R	002 NXMHI	003142RG	002 PRI02	= 000100 G	RCVLOA	002306RG	002
L\$STP	002024RG	002 L10064	046064R	002 NXML0	003140RG	002 PRI03	= 000140 G	RDERR	005202R	002
L\$STA	002030RG	002 L10065	051506R	002 NXMTST	021222R	002 PRI04	= 000200 G	RECMSC	002466RG	002
L\$SW	002166RG	002 L10066	062334R	002 NXR	003734R	002 PRI05	= 000240 G	RELV	002234RG	002
L\$TEST	002114RG	002 L10067	053512R	002 NXRERR	005550RG	002 PRI06	= 000300 G	REGSAV	017760R	002
L\$TIML	002014RG	002 L10070	055000R	002 NXRX	003773R	002 PRI07	= 000340 G	RETERR	005366R	002

TSV6 - PARAMETER CODING MACRO M113 01-FEB-84 17:02
SYMBOL TABLE

SEQ 239

REWIND 010624RG	002 S1.I1R= 020000	TSV2 002000RG	002 T##SEG= 010000	T14RST 026124R	002
RMCHBE= 000167	S1.I2R= 040000	TSV3 002176RG	002 T##SOF= 010101	T14SSR 025553R	002
RMCHEN= 000200	S1.PAR= 100000	TSV4 021316RG	002 T##SRV= 010026	T14TSB 025735R	002
RMMSG8= 000215	S2.ATI= 000010	TSV5 023216RG	002 T##SUB= 010074	T142RE 025456R	002
RMMSG8= 000234	S2.BTI= 000004	TSV6 101110RG	002 T##SW= 010001	T15AM2 033672R	002
RMPKTB= 000201	S2.DIM= 000200	TTIBFR= 177562 G	T##TES= 010077	T15AM3 033773R	002
RMPKTE= 000210	S2.ILW= 000100	TTICSR= 177560 G	T1 023216RG	002 T15AM4 034075R	002
RMR= 010000	S2.INR= 000020	TTION= 100156R	002 T10 066416RG	002 T15BFR 033122R	002
RWPACK C10720R	002 S2.OUT= 000040	TTION2 071204R	002 T11 074260RG	002 T15BF2 033610R	002
SC= 100000	S2.UND= 000003	TTIVEC= 000060 G	T12 077432RG	002 T15BS0 033610R	002
SCE= 020000	TBLEND= 003062RG	002 TVECSA 100160R	002 T12BFR 030012R	002 T15BS1 033611R	002
SCHERR 005274R	002 TCOASC 006310R	002 TVSAV2 071206R	002 T12BKG 030467R	002 T15DAT 033110R	002
SCME 005007R	002 TCOCOD 006510R	002 T#ARGC= 000001	T12BLK 030044R	002 T15LO0 031554R	002
SCMENU 100164R	002 TEMP1 003116RG	002 T#CODE= 001130	T12CHA 031460R	002 T15PAC 033100R	002
SDELAY 010470R	002 TEMP2 003120RG	002 T#ERRN= 002261	T12CKR 031240RG	002 T15PK2 033600R	002
SELASC 020262R	002 TERCLS= 000016	T#EXCP= 000000	T12CON 031046R	002 T15RES 034202R	002
SELDAT= 000004	TESTNO= 000014	T#FLAG= 000040	T12DAT 030000R	002 T15RT2 034254R	002
SEL2= 000002	TEXASC 006247R	002 T#FREE= 101352R	002 T12DPR 030646R	002 T15SSR 033616R	002
SETMAP 017130R	002 TFCASC 006351R	002 T#GMAN= 000000	T12GET 030226R	002 T15S2 033612R	002
SETU 021656R	002 TIMEXP 015362RG	002 T#HILI= 000776	T12HIA 030032R	002 T15S3 033614R	002
SFFMSG 011712RG	002 TMSG0 015410R	002 T#LAST= 000001	T12KT 030040R	002 T16BEN 037770R	002
SFHERR 003701R	002 TINERR 011631R	002 T#LOLI= 000000	T12LOA 030034R	002 T16BFR 037742R	002
SFIERR 003646R	002 TMPBFR 002632RG	002 T#LSYM= 010000	T12L00 026226R	002 T16BFS 037762R	002
SFMSG 011644RG	002 TNAM 016516R	002 T#LTNO= 000014	T12MSG 030371R	002 T16CLR 037606R	002
SFPTBL 002166RG	002 TPRISA 100162R	002 T#NEST= 177777	T12NIN 030555R	002 T16DAT 037730R	002
SIFLAG 003154RG	002 TPSAV2 071210R	002 T#NS0= 000000	T12NXM 030737R	002 T16DT2 040010R	002
SIMSG 011576R	002 TRANST 002166RG	002 T#NS1= 000005	T12PAC 027770R	002 T16D01 036362R	002
SKIPT 003400R	002 TSBA= 000000 G	T#NS2= 000002	T12PAR 030036R	002 T16D28 036364P	002
SOFINI 015604RG	002 TSBAH= 000001 G	T#NS3= 000003	T12SET 031416R	002 T16D53 036366H	002
SPACE 010300RG	002 TSDB= 000000 G	T#PCNT= 000000	T12SWR 031350R	002 T16D78 036370R	002
SPM1 101252R	002 TSDBH= 000001 G	T#PTAB= 010103	T12TBE 030176R	002 T16LEN 036712R	002
SPM4 101302R	002 TSFCOD 007050R	002 T#PTHV= 000001	T12WRT 030302R	002 T16L00 034332R	002
SR0= 177572	TSREJ= 000006	T#PTNU= 000001	T121LO 026340R	002 T16PAC 037720R	002
SR1= 177574	TSSDEF 006420R	002 T#SAVL= 177777	T122LO 026674R	002 T16PK2 040000R	002
SR2= 177576	TSSR= 000002 G	T#SEGL= 177777	T123LO 027114R	002 T16REJ 037024R	002
SR3= 172516	TSSRBI 003476RG	002 T#SEKO= 010000	T124LO 027514R	002 T16RES 037540R	002
SSR= 000200	TSSRFO 006227R	002 T#SIZE= 000005	T124TS 030042R	002 T16SEX 037700R	002
STATCO 012222R	002 TSSRH= 000003 G	T#SUBN= 000000	T13BFR 023642R	002 T16SRD 037632R	002
SVCGBL= 000000	TSSX 004014R	002 T#TAGL= 177777	T13DAT 023630R	002 T16SSH 036442R	002
SVCINS= 000000	TSTBLK 002152RG	002 T#TAGN= 010105	T13L00 023234R	002 T16TSB 036610R	002
SVCSUB= 000001	TSTCNT 002214RG	002 T#TEMP= 000000	T13MEM 023735R	002 T16T01 037141R	002
SVCTAG= 000000	TSTEND 016532R	002 T#TEST= 000014	T13NBA 023774R	002 T16T28 037240R	002
SVCTST= 000001	TSTFLA 002314RG	002 T#TSTM= 177777	T13PAC 023620R	002 T16T53 037340R	002
S#LSYM= 010000	TSTL00 016270RG	002 T#TSTS= 000001	T13RES 024136R	002 T16T78 037440R	002
SO.IDB= 000010	TSTPTR 002316RG	002 T#TAU= 010031	T13SSR 024047R	002 T16WMI 037652R	002
SO.IFB= 000002	TSTSET 016322RG	002 T#AUT= 010033	T14BFR 024756R	002 T162SS 036477R	002
SO.IFP= 000001	TST12I 030200R	002 T#CLF= 010034	T14BS0 024750R	002 T163SS 036543R	002
SO.ILD= 000020	TST13I 023662R	002 T#DAT= 010104	T14BS1 024751R	002 T17BEN 050102R	002
SO.ION= 000040	TST14I 026021R	002 T#DU= 010032	T14BS2 024752R	002 T17BFR 047762R	002
SO.IRD= 000100	TST15I 034163R	002 T#HAR= 010100	T14DAT 024750R	002 T17BFS 050002R	002
SO.IRW= 000004	TST16I 036372R	002 T#HW= 010000	T14D*A 025370R	002 T17CLE 047666R	002
SO.ISP= 000200	TST17I 046346R	002 T#INI= 010030	T14L00 024224R	002 T17CLR 047500R	002
S1.ICE= 002000	TST18I 050736R	002 T#MSG= 010025	T14NBA 025402R	002 T17DAT 047750R	002
S1.IEO= 010000	TST19I 057722R	002 T#PC= 000001	T14NIN 025643R	002 T17DT2 050120R	002
S1.IFM= 001000	TST20I 064722R	002 T#PRO= 010027	T14PAC 024740R	002 T17EXE 046244R	002
S1.IHE= 000400	TST39I 077402R	002 T#PTA= 010103	T14PK2 025360R	002 T17EXP 046122R	002
S1.IID= 004000	TST40I 101072R	002 T#RPT= 010035	T14RES 026066R	002 T17EXS 046142R	002

WITCH)

TSV6 - PARAMETER CODING MACRO M1113 01-FEB-84 17:02
SYMBOL TABLE

SEQ 240

T17L00	040144R	002	T19RFI	061606R	002	T203SS	065052R	002	T39DLY	075474R	002	WC,IFE	=	000002		
T17MSK	046116R	002	T19RSF	061540R	002	T204SS	065117R	002	T39DSW	076220R	002	WC,IGO	=	000001		
T17PAC	047740R	002	T19RST	061370R	002	T205SS	065162R	002	T39DTA	076210R	002	WC,IRE	=	000010		
T17PK2	050110R	002	T19SET	061760R	002	T206SS	065226R	002	T39EAI	076216R	002	WC,IRW	=	000004		
T17RFI	047646R	002	T19SEX	061722R	002	T208SS	065271R	002	T39ETN	076421R	002	WC,IOT	=	000100		
T17PSF	047544R	002	T19SNP	061562R	002	T23A	003144RG	002	T39ETS	076332R	002	WC,IIT	=	000040		
T17SET	047706R	002	T19SRD	061520R	002	T23B	003146RG	002	T39MCL	077263R	002	WC,ISR	=	000020		
T17SNP	047566R	002	T19SSR	057763R	002	T3	026200RG	002	T39NBA	076755R	002	WF,IED	=	000010		
T17SRD	047524R	002	T19WCT	061662R	002	T3BFLG	003150RG	002	T39NE	076253R	002	WF,IER	=	000004		
T17SSR	046365R	002	T19WFI	061626R	002	T3.1	026226R	002	T39NFL	076250R	002	WF,IHI	=	000200		
T17WFD	046244R	002	T19WFM	061702R	002	T3.2	026566R	002	T39OFF	077245R	002	WF,IRE	=	000040		
T17WFI	047612R	002	T19WST	061263R	002	T3.3	027056R	002	T39OF2	076514R	002	WF,IWF	=	000020		
T171CM	046705R	002	T191CM	060420R	002	T3.4	027354R	002	T39ON	077254R	002	WF,IWR	=	000100		
T172CM	046767R	002	T192CM	060502R	002	T38ASC	074211R	002	T39ON2	076560R	002	WF,I3R	=	000002		
T172SS	046422R	002	T192SS	060020R	002	T38ASN	074230R	002	T39PAC	075500R	002	WF,I4R	=	000001		
T173CM	047063R	002	T193CM	060576R	002	T38AS0	074106R	002	T39PK2	076200R	002	WRTCHR	=	010472RG	002	
T173SS	046466R	002	T193SS	060064R	002	T38AS1	074153R	002	T39PK3	076230R	002	WRTERR	=	005107R	002	
T174CM	047147R	002	T194SS	060131R	002	T38BFR	071236R	002	T39PK4	076240R	002	WRTMSG	=	005052R	002	
T174SS	046533R	002	T195CM	060664R	002	T38BS0	071230R	002	T39RES	077344R	002	WSMBK	=	021020RG	002	
T175CM	047232R	002	T195SS	060174R	002	T38BS1	071231R	002	T39RL	077340R	002	XFERAS	=	015550R	002	
T175SS	046576R	002	T196CM	060740R	002	T38BS2	071232R	002	T39SBS	077173R	002	XNXM	=	016206R	002	
T176CM	047306R	002	T196SS	060240R	002	T38CNT	074254R	002	T39SFS	077121R	002	XORBFO	=	007506R	002	
T176SS	046642R	002	T197CM	061023R	002	T38DAT	073544R	002	T39SIJ	076246R	002	XORFOR	=	007624R	002	
T177CM	047414R	002	T197SS	060303R	002	T38DLY	071212R	002	T39SSR	077031R	002	XST0	=	000006 G		
T18BFR	051442R	002	T198CM	061111R	002	T38DSW	071740R	002	T39TAD	075510R	002	XST1	=	000010 G		
T18CMP	051143R	002	T198SS	060352R	002	T38DTA	071730R	002	T39WPN	076676R	002	XST2	=	000012 G		
T18DAT	051430R	002	T199CM	061200R	002	T38EAI	071736R	002	T39WR	076242R	002	XST3	=	000014 G		
T18DT2	051500R	002	T2	024206RG	002	T38EB	071712R	002	T39WRT	076623R	002	XST4	=	000016 G		
T18EXP	050706R	002	T2.1	024224R	002	T38ID	073115R	002	T4	031524RG	002	XSOBOT	=	000002		
T18L00	050246R	002	T2.2	024474R	002	T38INT	072506R	002	T4.1	031554R	002	XSOEOT	=	000001		
T18MSK	050702R	002	T20BEN	066272R	002	T38MBP	073604R	002	T4.2	032060R	002	XSOIE	=	000040		
T18PAC	051420R	002	T20BFR	066152R	002	T38MSG	073055R	002	T4.3	032452R	002	XSOILA	=	000400		
T18PK2	051470R	002	T20BFS	066172R	002	T38MS1	072716R	002	T4ONE	101025R	002	XSOILC	=	001000		
T18SET	051310R	002	T20CLE	066054R	002	T38MS2	073011R	002	T5	034314RG	002	XSOLET	=	020000		
T18SMI	051266R	002	T20CLR	065642R	002	T38MS3	072055R	002	T5.1	034332R	002	XSOMOT	=	000200		
T18SRD	051246R	002	T20CWP	065547R	002	T38NBA	072342R	002	T5.2	035412R	002	XSONEF	=	002000		
T18SSR	050775R	002	T20DAT	066140R	002	T38NE	072000R	002	T6	040116RG	002	XSOONL	=	000100		
T18XS	050732R	002	T20DT2	066310R	002	T38OFL	072572R	002	T6.1	040144R	002	XSOPED	=	000010		
T182SS	051032R	002	T20EXE	064722R	002	T38ONL	072536R	002	T6.2	040434R	002	XSORLL	=	010000		
T183SS	051076R	002	T20EXP	064602R	002	T38PAC	071220R	002	T6.3	041644R	002	XSORLS	=	040000		
T19BEN	062212R	002	T20EXS	064622R	002	T38PK2	071720R	002	T6.4	043204R	002	XSOTMK	=	100000		
T19BFC	057536R	002	T20L00	062354R	002	T38PK3	071750R	002	T6.5	043562R	002	XSOVCK	=	000020		
T19BFR	062072R	002	T20MSK	064576R	002	T38PK4	071770R	002	T6.6	045034R	002	XSOULE	=	004000		
T19BFS	062112R	002	T20PAC	066130R	002	T38RES	073546R	002	T7	050230RG	002	XSOULK	=	000004		
T19CLE	061740R	002	T20PK2	066300R	002	T38SIJ	071776R	002	T8	051510R	002	XXCOMM	=	003122RG	002	
T19CLR	061474R	002	T20RFI	065726R	002	T38SSR	072416R	002	T8.1	051526R	002	X1ALWA	=	000000		
T19CNV	062004R	002	T20RSF	065446R	002	T38SST	072526R	002	T8.2	053526R	002	X1FALS	=	000040		
T19DAT	062060R	002	T20SET	066074R	002	T38TAD	071230R	002	T8.3	055014R	002	X1OFFS	=	000400		
T19DT2	062230R	002	T20SEX	066036R	002	T38WLE	072275R	002	T8.4	056264R	002	X1TRUE	=	000020		
T19EXE	057722R	002	T20SRD	065666R	002	T38WR	071772R	002	T9	062336RG	002	X1COR	=	020000		
T19EXP	057602R	002	T20SSR	064751R	002	T38WRL	072234R	002	T9.1	062354R	002	X1DLT	=	100000		
T19EXS	057622R	002	T20SWP	065337R	002	T38WRT	072150R	002	UAM	=	000200 G		X1MBZ	=	017375	
T19L00	051526R	002	T20WFI	065746R	002	T39BFR	075516R	002	UNITN	=	002202RG		X1RBP	=	000400	
T19MSK	057576R	002	T20WFM	066002R	002	T39BS0	075510R	002	UNREC	=	000006		X1SPA	=	040000	
T19PAC	062050R	002	T20WMI	066022R	002	T39BS1	075511R	002	USI	=	004117R		X1UNC	=	000002	
T19PK2	062220R	002	T20WNP	065706R	002	T39BS2	075512R	002	WAITF	=	016060RG		X2BUF	=	000100	
T19PRE	057534R	002	T202SS	065006R	002	T39DAT	077342R	002	WC,IFA	=	000200		X2EXT	=	000200	

TSV6 - PARAMETER CODING MACRO M1113 01-FEB-84 17:02
SYMBOL TABLE

SEQ 241

X2.OFM= 100000	X2.UNI= 000007	X3.MDE= 177400	X3.SPA= 000200	X4.RCE= 040000
X2.RCE= 040000	X2.WCF= 002000	X3.OPI= 000100	X3.TRF= 000020	X4.TSM= 020000
X2.REV= 000077	X3.DCK= 000010	X3.REV= 000040	X4.HSP= 100000	X4.WRC= 000377
X2.SPA= 035400	X3.MBZ= 000006	X3.RIB= 000001	X4.MBZ= 017400	

. ABS. 000000 000
000000 001
ABS 101352 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 31056 WORDS (122 PAGES)
DYNAMIC MEMORY: 20614 WORDS (79 PAGES)
ELAPSED TIME: 00:51:47
CZTSBA,CZTSBA.SEQ/-SP=SVC/ML,TSV1B,TSV22B,TSV3B,TSV4,TSV55B,TSV6
.

TEST 9: READ/WRITE B1
TEST 10: MANUAL INTE....C1
TEST 10: MANUAL INTE....D1
TEST 10: MANUAL INTE....E1
TEST 10: MANUAL INTE....F1
TEST 10: MANUAL INTE....G1
TEST 10: MANUAL INTE....H1
TEST 10: MANUAL INTE....I1
TEST 10: MANUAL INTE....J1
TEST 10: MANUAL INTE....K1
TEST 10: MANUAL INTE....L1
TEST 10: MANUAL INTE....M1
TEST 10: MANUAL INTE....N1

TEST 10: MANUAL INTE....B2
TEST 10: MANUAL INTE....C2
TEST 10: MANUAL INTE....D2
TEST 11: CONFIGURATI....E2
TEST 11: CONFIGURATI....F2
TEST 11: CONFIGURATI....G2
TEST 11: CONFIGURATI....H2
TEST 11: CONFIGURATI....I2
TEST 11: CONFIGURATI....J2
TEST 11: CONFIGURATI....K2
TEST 12: SCOPE LOOPS....L2
TEST 12: SCOPE LOOPS....M2
TEST 12: SCOPE LOOPS....N2

TEST 12: SCOPE LOOPS....B3
TEST 12: SCOPE LOOPS....C3
SOFTWARE PARAMETER C....C3
SYMBOL TABLE E3
SYMBOL TABLE F3
SYMBOL TABLE G3
SYMBOL TABLE H3
SYMBOL TABLE I3