

The image displays a large grid of 140 small tables, arranged in 10 rows and 14 columns. Each small table contains technical data, likely test results or component specifications, with various columns and rows of text and numbers. The text is small and difficult to read, but the layout is consistent across all tables, suggesting a standardized format for the data presented.

The image shows a grid of 48 small, illegible technical diagrams or data tables arranged in 12 rows and 4 columns on the left side of the page. The diagrams appear to be technical drawings or data tables, but the text is too small to read. The right side of the page is mostly blank with some faint, illegible markings.

B1

D
A [C

1

USER DOCUMENTATION

MACRO V05.03 Tuesday 28-Apr-87 08:29 Page 2

SEQ 000

.REM_
IDENTIFICATION

PRODUCT ID: AC-T095E-MC
PRODUCT TITLE: CVTSBE0 TSV05 CTRL PART 2
DEPARTMENT: COMPUTER SPECIAL SYSTEMS/PGG
DATE: JUNE 08, 1987

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

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

COPYRIGHT (C) 1982, 1987 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

TABLE OF CONTENTS

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

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

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

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

1.2 SYSTEM REQUIREMENTS

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

1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

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

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES:

FUNCTIONAL PDP-11/23 CENTRAL PROCESSOR AND MEMORY
FUNCTIONAL CONSOLE TERMINAL
FUNCTIONAL STANDALONE DIAGNOSTIC SUPERVISOR
FUNCTIONAL DIAGNOSTIC LOADER/MONITOR (XXDP+)

1.5 ASSUMPTIONS

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

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

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

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

2.1.1 OPERATOR COMMANDS

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

BOOT THE DIAGNOSTIC MEDIA

```
.R VTSB??  
DIAG. RUN-TIME SERVICES REV D. APR 79  
CVTSB-E-0  
****TSV05 LOGIC DIAGNOSTIC****  
UNIT IS TSV05  
>DR
```

2.2 SWITCHES

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

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

EXAMPLE OF SWITCH USAGE:

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

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

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

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

2.3 FLAGS

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

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

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

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

/FLAGS:LOE:IER:BOE

2.4 HARDWARE QUESTIONS

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

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

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

TSBA/TSDB = 172520, VECTOR = 224

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

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

UNIT 0

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

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

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

2.5 SOFTWARE QUESTIONS

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

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

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

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

2.6 EXTENDED P-TABLE DIALOGUE

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

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

UNITS (0) ? 8<CR>

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

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

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

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

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

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

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

UNIT 8
CSR ADDRESS (0) 160000<CR>
SUB-DEVICE # (0) ? 7<CR>
Q-FACTOR (0) 1 ? <CR>

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

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

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

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

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

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

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

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

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

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

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

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

2.7 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

1. BOOT XXDP+
2. GIVE THE DATE AND ANSWER THE LSI AND 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/BAD DATA. THESE ARE ALWAYS PRINTED UNLESS THE "IER", "IBR" OR "IXR" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

3.2 SPECIFIC ERROR MESSAGES

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

ERROR MESSAGE EXAMPLE 1

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

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

TAPE BUS SIGNALS IN WORD #8: - DESIGNATOR <BIT #>
PARERR<15> IEOT <12> IFMK <9> IRDY<6> IRWD<2>
IRESV2<14> IIDENT<11> IHER <8> IONL<5> IFBY<1>
IRESV1<13> ICER <10> ISPEED<7> ILDP<4> IFPT<0>
TAPE BUS SIGNALS IN WORD #9:
DATMIS<7> ILW<6> OUTRDY<5> INRDY<4>
MESSAGE BUFFER ADDRESS = 047352

MESSAGE BUFFER CONTENTS:
WORD #0 EXPD: 100020 RECV: 100020 XOR: 000000
WORD #1 EXPD: 000012 RECV: 000012 XOR: 000000
WORD #2 EXPD: 000000 RECV: 000000 XOR: 000000
WORD #3 EXPD: 000010 RECV: 000010 XOR: 000000
WORD #4 EXPD: 000000 RECV: 000000 XOR: 000000
WORD #5 EXPD: 000000 RECV: 000000 XOR: 000000
WORD #6 EXPD: 000000 RECV: 000000 XOR: 000000
WORD #7 EXPD: 000000 RECV: 000000 XOR: 000000
WORD #8 EXPD: 070217 RECV: 070217 XOR: 000000
WORD #9 EXPD: 000074 RECV: 000034 XOR: 000040

ERROR MESSAGE EXAMPLE 2

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

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

ERROR MESSAGE EXAMPLE 3

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

CVTSB 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-11/23)

DR>STA/FLA:PNT:HOE:UAM

UNITS (D) ? 1

UNIT 0

DEVICE ADDRESS (0) 172520 ? <CR>

VECTOR (0) 224 ? <CR>

CHANGE SW (L) ? N<CR>

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

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

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

0 ERRORS

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

PROGRAM RUN TIMES

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

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

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

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

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

5.0 DEVICE INFORMATION TABLES

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

CHANGE HW (L) ?

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

UNIT 0

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

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

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

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

CHANGE SW (L) ?

6.0 TEST SUMMARIES

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

TEST DESCRIPTION:

This test verifies that a Hardware Initialize command invoked after a Write Characteristics command sets up the Command, Message and Characteristic image blocks in the controller ram correctly.

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

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

TEST 3: DMA MEMORY ADDRESSING

This test verifies that the controller can properly address and access all available CPU memory (other than that occupied by the diagnostic and diagnostic supervisor code) for both reading (DATI) and writing (DATO). Verified are the LSI-11 Bus drivers for all available address lines. Up to this point only 16 bits have been used for DMA transfers.

CAUTION

The LSI BUS drivers for all available address lines are only checked when running on a 11/23B system with more than 128K words of memory!

TEST 4: RAM EXERCISER TEST

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

TEST 5: EXTENDED FEATURES SWITCH AND TIMERS A,B

TEST DESCRIPTION:

This test verifies the Invert Extended Features function can logically invert the Extended features switch and that the internal timers A and B operate correctly.

TEST 6: FIFO EXERCISER

TEST DESCRIPTION:

This test uses the Write Subsystem Memory command to verify the controller's FIFO and associated status and control logic.

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

TEST DESCRIPTION:

Write to TSSR register to soft initialize the controller
Do WRITE CHARACTERISTICS to check for Extended Features Switch
If Extended Features Hardware Switch Clear then:
Do Write Subsystem Write Miscellaneous to Set Extended Features.
Do WRITE CHARACTERISTICS to select reserved unit 7
Do a Write Subsystem READ STATUS
If any transport interface signals are asserted then Print Error

TEST 8: TRANSPORT BUS INTERFACE LOOPBACK TEST

TEST DESCRIPTION:

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

TEST 9: READ/WRITE DATA PARITY TEST

TEST DESCRIPTION:

This test verifies that the Write Data Parity generator and the Read Data Parity checker operate properly. The Transport Bus signal loopback mode is enabled and a Set Wrong parity function is executed. Then various Write Subsystem Memory functions are performed to write data to and from the FIFO in loopback mode. The program then checks to insure a Read Data parity error occurred.
A Reset FIFO is done and the Read Data parity error bit is again tested to insure it cleared.
Finally a Clear wrong parity function is done and it is verified the data word can pass in loopback mode without setting Read Data parity error.

TEST 10: MANUAL INTERVENTION

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

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

TEST 11: CONFIGURATION TYPEOUT

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

- 1.0 STATE OF THE EXTENDED FEATURES SWITCH ON THE M7196: ON (EXTENDED FEATURES ENABLED) OR OFF (EXTENDED FEATURES DISABLED).
- 2.0 STATE OF THE BUFFERING ENABLE SWITCH ON THE M7196: ON (BUFFERING ENABLED) OR OFF (BUFFERING DISABLED).
- 3.0 MICROCODE REVISION LEVEL OF THE M7196.
- 4.0 NUMBER OF TAPE TRANSPORTS CONNECTED TO THE CONTROLLER.
- 5.0 UNIT SELECT CODE AND STATE (ONLINE/OFFLINE, WRITE ENABLED/PROTECTED) OF EACH CONNECTED TRANSPORT. IN ADDITION, THE PROGRAM WILL INDICATE, FOR EACH ON-LINE TRANSPORT, WHETHER OR NOT IT IS EQUIPPED WITH THE EXTENDED TAPE STATUS READOUT FEATURE.

TEST 12: SCOPE LOOPS

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

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

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

7.0 MAINTENANCE HISTORY

REVISION A - MARCH 1982

REVISION B - APRIL 1983

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

REVISION C - JUNE 1984

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

REVISION D - JULY 1985

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

H2

REVISION E - APRIL 1987

CHANGES MADE TO ALLOW DIAGNOSTICS TO WORK WITH
THE NEW TSV05 MICROCODE (REVISION 2). THE NEW
TSV05 MICROCODE ALWAYS IN EXTENDED FEATURE MODE.

```

936          .TITLE  TSV2 - PROGRAM HEADER
937          .SBTTL  PROGRAM HEADER
938
944          .MCALL  SVC
945 000000   SVC          ; INITIALIZE SUPERVISOR MACROS
946          .ENABLE LC
947          .NLIST  BEX,CND
953 000000   .ENABL  ABS,AMA
954          =2000
955 002000   BGNMOD  TSV2
          002000
956
957          TSV2::
958          ;**
959          ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
960          ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
961          ;--
962 002000   POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
963 002000   HEADER  CVTSB,E,0,655,0
          002000   L$NAME::          ;DIAGNOSTIC NAME
          002000   .ASCII  /C/
          002001   .ASCII  /V/
          002002   .ASCII  /T/
          002003   .ASCII  /S/
          002004   .ASCII  /B/
          002005   .BYTE   0
          002006   .BYTE   0
          002007   .BYTE   0
          002010   L$REV::          ;REVISION LEVEL
          002010   .ASCII  /E/
          002011   L$DEPO::        ;0
          002011   .ASCII  /0/
          002012   L$UNIT::        ;NUMBER OF UNITS
          002012   .WORD    0
          002014   L$TIML::        ;LONGEST TEST TIME
          002014   .WORD    655.
          002016   L$HPCP::        ;PTR. TO H.W. QUES.
          002016   .WORD    L$HARD
          002020   L$SPCP::        ;PTR. TO S.W. QUES.
          002020   .WORD    L$SOFT
          002022   L$HPTP::        ;PTR. TO DEF. H.W. PTABLE
          002022   .WORD    L$HW
          002024   L$SPTP::        ;PTR. TO S.W. PTABLE
          002024   .WORD    L$SW
          002026   L$LADP::        ;DIAG. END ADDRESS
          002026   .WORD    L$LAST
          002030   L$STA::         ;RESERVED FOR APT STATS
          002030   .WORD    0
          002032   L$CO::         ;DIAGNOSTIC TYPE
          002032   .WORD    0
          002034   L$DTYP::        ;APT EXPANSION
          002034   .WORD    0
          002036   L$APT::         ;PTR. TO DISPATCH TABLE
          002036   .WORD    0
          002040   L$DTP::         ;DIAGNOSTIC RUN PRIORITY
          002040   .WORD    L$DISPATCH
          002042

```

J2

PROGRAM HEADER

002042	000000			
002044		L\$ENVI::	.WORD 0	;FLAGS DESCRIBE HOW IT WAS SETUP
002044	000000			
002046		L\$EXP1::	.WORD 0	;EXPANSION WORD
002046	000000			
002050		L\$MREV::	.WORD 0	;SVC REV AND EDIT #
002050	003			
002051	003			
002052		L\$EF::	.BYTE C\$REVISION	
002052	000000			
002054	000000			
002056		L\$SPC::	.WORD 0	
002056	000000			
002060		L\$DEVP::	.WORD 0	; POINTER TO DEVICE TYPE LIST
002060	003404			
002062		L\$REPP::	.WORD L\$DVTYP	;PTR. TO REPORT CODE
002062	022254			
002064		L\$EXP4::	.WORD L\$RPT	
002064	000000			
002066		L\$EXP5::	.WORD 0	
002066	000000			
002070		L\$AUT::	.WORD 0	;PTR. TO ADD UNIT CODE
002070	021742			
002072		L\$DUT::	.WORD L\$AU	;PTR. TO DROP UNIT CODE
002072	022040			
002074		L\$LUN::	.WORD L\$DU	;LUN FOR EXERCISERS TO FILL
002074	000000			
002076		L\$DESP::	.WORD 0	;PTR. TO DIAG. DESCRIPTION
002076	003412			
002100		L\$LOAD::	.WORD L\$DESC	;GENERATE SPECIAL AUTOLOAD EMT
002100	104035			
002102			EMT E\$LOAD	
002102	000000	L\$ETP::	.WORD 0	;PTR. TO ERR TBL
002104		L\$ICP::	.WORD 0	;PTR. TO INIT CODE
002104	021146			
002106		L\$CCP::	.WORD L\$INIT	;PTR. TO CLEAN-UP CODE
002106	022226			
002110		L\$ACP::	.WORD L\$CLEAN	;PTR. TO AUTO CODE
002110	022146			
002112		L\$PRT::	.WORD L\$AUTO	;PTR. TO PROTECT TABLE
002112	021136			
002114		L\$TEST::	.WORD L\$PROT	;TEST NUMBER
002114	000000			
002116		L\$DLY::	.WORD 0	;DELAY COUNT
002116	000000			
002120		L\$HIME::	.WORD 0	;PTR. TO HIGH MEM
002120	000000			

K2

DISPATCH TABLE

965
966
967
968
969
970
971
972 002122 000014
002122 023036
002124 024016
002126 026010
002130 031404
002132 034174
002134 037766
002136 050100
002140 051360
002142 062206
002144 066256
002146 074114
002150 077624

.SBTTL DISPATCH TABLE

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

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

L2

DEFAULT HARDWARE P-TABLE

```
974                                     .SBTTL  DEFAULT HARDWARE P-TABLE
975
976                                     ;**
977                                     ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
978                                     ; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
979                                     ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
980                                     ;--
981 002154                               BGNHW   DFPTBL   ;DEFAULT HARD-P-TABLE
    002154 000003                       .WORD   L10000-L$HW/2
    002156                               L$HW::
    002156                               DFPTBL::
982
983 002156 172520                       .WORD   172520       ; 1ST (OF 2) REGISTERS.
984 002160 000224                       .WORD   224          ; INTERRUPT VECTOR
985 002162 000200                       .WORD   PRI04       ; INTERRUPT PRIORITY.
986 002164                               ENDHW
    002164                               L10000:
```

SOFTWARE P-TABLE

```

988                .SBTTL  SOFTWARE P-TABLE
989
990                ;**
991                ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
992                ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
993                ;--
994 002164          BGNSW  SFPTBL
                   002164 000004  .WORD  L10001-L$SW/2
                   002166
                   002166
995
996 002166          TRANSTST::  .WORD  0      ; ENABLE TEST OF TRANSPORT(S) IF =1
997 002170          NOITS::    .WORD  0      ; INHIBIT ITERATION OPTION.
998
999                ; ... 0 = ITERATE.
                   ; ...NZ = INHIBIT ITERATE.
1000 002172         LERRMAX::   .WORD  15.    ; LOCAL (PER TEST) ERROR LIMIT
1001 002174         GERRMAX::   .WORD  200.   ; GLOBAL (PER UNIT) ERROR LIMIT
1002 002176         ENDSW
                   L10001:
1003
1004 002176         ENDMOD

```

SOFTWARE P-TABLE

1014
 1015
 1020
 1026
 1027 002176
 002176
 1028
 1029
 1030
 1031
 1032
 1033
 1034
 1035
 1039 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

; BIT9== BIT09
 ; BIT8== BIT08
 ; BIT7== BIT07
 ; BIT6== BIT06
 ; BIT5== BIT05
 ; BIT4== BIT04
 ; BIT3== BIT03
 ; BIT2== BIT02
 ; BIT1== BIT01
 ; BIT0== BIT00

; EVENT FLAG DEFINITIONS
 ; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

000040	EF.START== 32.	; START COMMAND WAS ISSUED
000037	EF.RESTART== 31.	; RESTART COMMAND WAS ISSUED
000036	EF.CONTINUE== 30.	; CONTINUE COMMAND WAS ISSUED
000035	EF.NEW== 29.	; A NEW PASS HAS BEEN STARTED
000034	EF.PWR== 28.	; A POWER-FAIL/POWER-UP OCCURRED

; PRIORITY LEVEL DEFINITIONS

GLOBAL EQUATES SECTION

```

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

```

; OPERATOR FLAG BITS

```

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

```

1040
1041 002176

; DEFINE MEMORY MANAGEMENT REGISTERS

```

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

```

```

000250
177572
177574
177576
172516

```

C3

MEMORY MANAGEMENT DEFINITIONS

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

MEMORY MANAGEMENT DEFINITIONS

```

SDPAR4= 172270
SDPAR5= 172272
SDPAR6= 172274
SDPAR7= 172276
.ENDC
.ENDC
;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
172300 KIPDR0= 172300
172302 KIPDR1= 172302
172304 KIPDR2= 172304
172306 KIPDR3= 172306
172310 KIPDR4= 172310
172312 KIPDR5= 172312
172314 KIPDR6= 172314
172316 KIPDR7= 172316
      .IF NB
;*KERNEL "D" PAGE
DESCRIPTOR REGISTERS
      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

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

1046                      .SBTTL TSV05 REGISTER AND PACKET DEFINITIONS
1047
1048                      ;
1049                      ; SOME GENERAL EQUATES.
1050                      ;
1051
1052                      000004 ERRVEC==      4           ; POINTER TO ERROR VECTOR FOR BUS TIME OUT.
1053                      000060 TTIVEC==     60           ; INTERRUPT VECTOR FOR CONSOLE INPUT
1054                      177560 TTICSR==    177560        ; BUS ADDRESS OF CONSOLE INPUT
1055                      177562 TTIBFR==    177562        ; CONSOLE INPUT DATA BUFFER
1056                      177520 BDVPCR==    177520        ; BDV11 PAGE CONTROL REGISTER
1057
1058                      ;*
1059                      ;BIT DEFINITIONS FOR TSSR REGISTER
1060                      ;-
1061
1062                      100000 SC=          BIT15         ;SPECIAL CONDITION
1063                      040000 BIE=          BIT14         ;BUS INTERFACE ERROR
1064                      020000 SCE=          BIT13         ;SANITY CHECK ERROR
1065                      010000 RMR=          BIT12         ;MODIFICATION REFUSED
1066                      004000 NXM=          BIT11         ;NONEXISTANT MEMORY ERROR
1067                      002000 NBA=          BIT10         ;NEED BUFFER ADDRESS
1068                      001400 HIADDR=     BIT9:BIT8       ;EXTENDED ADDRESS BITS
1069                      000200 SSR=          BIT7          ;SUB SYSTEM READY
1070                      000100 OFL=          BIT6          ;OFF LINE BIT
1071                      000060 FATERR=     BIT4:BIT5       ;FATAL TERMINATION ERROR CODES
1072                      000016 TERCLS=     BIT3:BIT2:BIT1  ;TERMINATION CODES
1073
1074                      ;*
1075                      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
1076                      ;(XST0)
1077                      ;
1078                      ;-
1079
1080
1081                      100000 XSOTMK=     BIT15         ;TAPE MARK DETECTED
1082                      040000 XSORLS=     BIT14         ;RECORD LENGTH SHORT
1083                      020000 XSOLET=     BIT13         ;LOGICAL END OF TAPE
1084                      010000 XSORLL=     BIT12         ;RECORD LENGTH LONG
1085                      004000 XSOMLE=     BIT11         ;WRITE LOCK ERROR
1086                      002000 XSONEF=     BIT10         ;NON EXECUTABLE FUNCTION
1087                      001000 XSOILC=     BIT9          ;ILLEGAL COMMAND
1088                      000400 XSOILA=     BIT8          ;ILLEGAL ADDRESS
1089                      000200 XSOMOT=     BIT7          ;TAPE IN MOTION
1090                      000100 XSOONL=     BIT6          ;TRANSPORT ON LINE
1091                      000040 XSOIE=      BIT5          ;INTERRUPT ENABLE
1092                      000020 XSOVCK=     BIT4          ;VOLUME CHECK BIT
1093                      000010 XSOPED=     BIT3          ;PHASE ENCODED DRIVE
1094                      000004 XSOWLK=     BIT2          ;WRITE LOCKED
1095                      000002 XSOBOT=     BIT1          ;BEGINNING OF TAPE
1096                      000001 XSOEOT=     BIT0          ;END OF TAPE

```


TSV05 REGISTER AND PACKET DEFINITIONS

```

1098
1099      ;*
1100      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
1101      ;(XST1)
1102      ;-
1103      100000 X1.DLT = BIT15      ;DATA LATE
1104      040000 X1.SPARE= BIT14      ;NOT USED
1105      020000 X1.COR = BIT13      ;CORRECTABLE DATA ERROR
1106      017375 X1.MBZ = BIT12-BIT11-BIT10-BIT9-BIT7-BIT6-BIT5-BIT4-BIT3-BIT2-BIT0 ;ALWAYS 0
1107      000400 X1.RBP = BIT8      ;READ BUS PARITY ERROR
1108      000002 X1.UNC = BIT1      ;UNCORRECTABLE DATA OR HARD ERROR
1109
1110      ;*
1111      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
1112      ;(XST2)
1113      ;-
1114      100000 X2.OPM = BIT15      ;OPERATION IN PROGRESS (TAPE MOVING)
1115      040000 X2.RCE = BIT14      ;RAM CHECKSUM ERROR
1116      035400 X2.SPARE= BIT13-BIT12-BIT11-BIT9-BIT8 ;NOT USED BY TSV05 (ALWAYS=0)
1117      002000 X2.WCF = BIT10      ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
1118      000200 X2.EXTF = BIT7      ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
1119      000100 X2.BUFE = BIT6      ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
1120      000077 X2.REV = 000077 ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
1121      000007 X2.UNIT = BIT2-BIT1-BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.
1122
1123      ;*
1124      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
1125      ;(XST3)
1126      ;-
1127      177400 X3.MDE = 177400 ;MICRO-DIAGNOSTIC ERROR CODE
1128      000200 X3.SPARE= BIT7      ;NOT USED BY TSV05
1129      000100 X3.OPI = BIT6      ;OPERATION INCOMPLETE
1130      000040 X3.REV = BIT5      ;REVERSE
1131      000020 X3.TRF = BIT4      ;TRANSPORT RESPONSE FAILURE
1132      000010 X3.DCK = BIT3      ;DENSITY CHECK
1133      000006 X3.MBZ =BIT2-BIT1 ;NOT USED ALWAYS 0
1134      000001 X3.RIB = BIT0      ;REVERSE INTO BOT
1135
1136      ;*
1137      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
1138      ;(XST4)
1139      ;-
1140      100000 X4.HSP = BIT15      ;HIGH SPEED
1141      040000 X4.RCE = BIT14      ;RETRY COUNT EXCEEDED
1142      020000 X4.TSM = BIT13      ;TRANSPORT SPECIAL MODE
1143      017400 X4.MBZ = BIT12-BIT11-BIT10-BIT9-BIT8 ;NOT USED ALWAYS 0
1144      000377 X4.WRC = 000377 ;WRITE RETRY COUNT FIELD
1145
1146      ;*
1147      ;TSSR TERMINATION CODES (BIT 0-2)
1148      ;
1149      ;-
1150
1151      000006 TSREJ= 3*2 ;COMMAND REJECTED
1152      000006 UNREC= 6 ;UNRECOVERABLE ERROR

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

1154      ;+
1155      ;
1156      ;DEVICE REGISTER OFFSETS
1157      ;
1158      ;-
1159
1160      000000      TSBA== 0
1161      000000      TSDB== 0      ;TSDB/TSBA REGISTER
1162      000001      TSBAH== 1
1163      000001      TSDBH== 1      ;TSDB/TSBA REGISTER HIGH BYTE
1164      000002      TSSR== 2      ;TSSR REGISTER
1165      000003      TSSRH== 3      ;TSSR REGISTER HIGH BYTE
1166
1167      ;+
1168      ; TSDB ADDRESS BIT DEFINITIONS
1169      ;-
1170      000003      A1716 = BIT1+BIT0      ;ADDRESS BITS 17:16 ARE IN 1:0
1171
1172      ;+
1173      ; COMMAND DEFINITIONS
1174      ;-
1175      000017      P.GETSTAT      = 17      ;GET STATUS
1176      000013      P.INIT        = 13      ;INITIALIZE
1177      000012      P.CONTROL     = 12      ;CONTROL COMMANDS
1178      000011      P.FORMAT      = 11      ;FORMAT
1179      000010      P.POSITION    = 10      ;POSITION
1180      000006      P.WRTSUB      = 6       ;SUBSYSTEM WRITE
1181      000005      P.WRITE       = 5       ;WRITE
1182      000004      P.WRTCHAR    = 4       ;WRITE CHARACTERISTICS
1183      000001      P.READ        = 1       ;READ
1184
1185      ;+
1186      ; COMMAND PACKET HEADER WORD BIT DEFINITIONS
1187      ;-
1188      100000      P.ACK      = BIT15      ;BUFFER AVAIL FOR CONTROLLER
1189      040000      P.CVC      = BIT14      ;CLEAR VOLUME CHECK
1190      020000      P.OPP      = BIT13      ;REVERSE SEQUENCE OF DATA BITS
1191      010000      P.SWB      = BIT12      ;SWAP BYTES IN MEMORY
1192      007400      P.MODE     = BIT11:BIT10:BIT9:BIT8 ;EXTENDED COMMAND MODE FIELD
1193      000200      P.IE       = BIT7       ;INTERRUPT ENABLE
1194      000140      P.FMT= BIT6:BIT5      ;PACKET HEADER TYPE (ALWAYS=0)
1195      000037      P.CMD      = 37        ;MAJOR COMMAND FIELD
1196
1197      ;+
1198      ; CONTROL COMMAND MODE CODES
1199      ;-
1199      000000      PC.RELEASE   = 0*256.   ;RELEASE BUFFER
1200      000400      PC.REWIND   = 1*256.   ;REWIND
1201      001000      PC.NOOP     = 2*256.   ;NO-OP
1202      002000      PC.IEREW    = 4*256.   ;REWIND IMMEDIATE INTERRUPT
1203      002400      PC.ERASE    = 5*256.   ;SECURITY ERASE

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

1205
1206      ;*
1207      ; CONTROLLER RAM DEFINITIONS
1208      ;-
1208      000167      RMCHBEG = 167      ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
1209      000200      RMCHEND = 200      ;CHARACTERISTICS IO DATA END RAM ADDRESS
1210      000201      RMPKTBE= 201      ;COMMAND PACKET BEGIN RAM ADDRESS
1211      000210      RMPKTEND= 210      ;COMMAND PACKET END RAM ADDRESS
1212      000215      RMMSGBE= 215      ;MESSAGE BUFFER BEGIN RAM ADDRESS
1213      000234      RMMSGEND= 234      ;MESSAGE BUFFER END RAM ADDRESS
1214
1215      ;*
1216      ; REGISTER DEFINITIONS IN THE MESSAGE BUFFER
1217      ;
1218      ;-
1219
1220      000006      XST0== 6      ;EXTENDED STATUS REGISTER 0 (WORD 4)
1221      000010      XST1== 8      ;EXTENDED STATUS REGISTER 1 (WORD 5)
1222      000012      XST2== 10     ;EXTENDED STATUS REGISTER 2 (WORD 6)
1223      000014      XST3== 12     ;EXTENDED STATUS REGISTER 3 (WORD 7)
1224      000016      XST4== 14     ;EXTENDED STATUS REGISTER 4 (WORD 8)
1225
1226      ;*
1227      ;
1228      ; OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
1229      ;
1230      ;-
1231
1232      000002      PKLOW  = 2      ;LOW ORDER CHARACTERISTIC DATA POINTER
1233      000004      PKHI   = 4      ;HIGH ORDER CHARACTERISTIC DATA POINTER
1234      000006      PKBCNT = 6      ;NUMBER OF BYTES IN DATA PACKET
1235
1236      000010      EXBCNT=10      ;NUMBER OF BYTES IN EXTENDED DATA PACKET
1237
1238      ;*
1239      ; DATA PACKET OFFSETS FOR WRITE SUBSYSTEM COMMAND
1240      ;-
1241      000000      BSELO  = 0      ;BYTE 0
1242      000001      BSEL1  = 1      ;BYTE 1
1243      000002      SEL2   = 2      ;WORD 2
1244      000004      SELDATA = 4      ;WORD 3

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

1246
1247      ;*
1248      ;BSEL0 SELECT CODES FOR WRITE SUBSYSTEM COMMAND
1249      ;-
1249      000000 PW.NOP          = 0          ;NO-OP
1250      000001 PW.RDRAM       = 1          ;READ RAM
1251      000002 PW.WTRAM       = 2          ;WRITE RAM
1252      000003 PW.RFIFO       = 3          ;READ FIFO
1253      000004 PW.WFIFO       = 4          ;WRITE FIFO
1254      000005 PW.RDSTAT      = 5          ;READ STATUS
1255      000006 PW.WCTL        = 6          ;WRITE TAPE CONTROL
1256      000007 PW.WFMT        = 7          ;WRITE TAPE FORMAT
1257      000010 PW.WMISC       = 10         ;WRITE MISCELLANEOUS
1258      000011 PW.WNPR        = 11         ;WRITE NPR CONTROL
1259      000020 PW.D22         = 20         ;DO MICROTEST 22
1260      000021 PW.D11         = 21         ;DO MICROTEST 11
1261      000022 PW.D13         = 22         ;DO MICROTEST 13
1262      000023 PW.NO1311     = 23         ;DISABLE MICROTEST 11 AND 13
1263      000024 PW.RDXT        = 24         ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS)
1264
1265      ;*
1266      ;BSEL1 CODES FOR WRITE TAPE CONTROL
1267      ;-
1268      000200 WC.IFAD         = BIT7        ;IFAD - FORMATTER ADDRESS
1269      000100 WC.IOTAD        = BIT6        ;ITADO - TRANSPORT ADDRESS BIT 0
1270      000040 WC.I1TAD        = BIT5        ;ITAD1 - TRANSPORT ADDRESS BIT 1
1271      000020 WC.I5RESV      = BIT4        ;IRESV5 - RESERVED #5
1272      000010 WC.IREW        = BIT3        ;IREW - REWIND
1273      000004 WC.IRWU        = BIT2        ;IRWU - REWIND AND UNLOAD
1274      000002 WC.IFEN        = BIT1        ;IFEN - FORMATTER ENABLE
1275      000001 WC.IGO         = BIT0
1276
1277      ;*
1278      ;BSEL1 CODES FOR WRITE FORMAT
1279      ;-
1280      000200 WF.IHISP        = BIT7        ;IHISP - HIGH SPEED
1281      000100 WF.IWRT        = BIT6        ;IWRT - WRITE
1282      000040 WF.IREV        = BIT5        ;IREV - REVERSE
1283      000020 WF.IWFM        = BIT4        ;IWFM - WRITE FILE MARK
1284      000010 WF.IEDIT       = BIT3        ;IEDIT - EDIT
1285      000004 WF.IERASE      = BIT2        ;IERASE - ERASE
1286      000002 WF.I3RESV     = BIT1        ;IRESV3 - RESERVED #3
1287      000001 WF.I4RESV     = BIT0        ;IRESV4 - RESERVED #4
1288
1289      ;*
1290      ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
1291      ;-
1292      000200 MS.EXT          = BIT7        ;INVERT SENSE OF EXTENDED FEATURES SWITCH
1293      000020 MS.RSFIFO      = BIT4        ;RESET FIFO AND INPUT PARITY ERRORR
1294      000010 MS.RSTAPE      = BIT3        ;RESET TAPE STATUS IN 2 FLIP-FLOPS
1295      000006 MS.ATTN        = BIT2:BIT1   ;ATTENTION TRIGGER FIELD
1296      000001 MS.RSD         = BIT0        ;RESET TIMER A,B THEN DELAY TIMES IN SEL2

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

1298
1299
1300
1301      000000
1302      000002
1303      000004
1304      000006
1305
1306
1307
1308      000200
1309      000100
1310      000040
1311      000020
1312
1313
1314
1315
1316      000200
1317      000100
1318      000040
1319      000020
1320      000010
1321      000004
1322      000003
1323      100000
1324      040000
1325      020000
1326      010000
1327      004000
1328      002000
1329      001000
1330      000400
1331      000200
1332      000100
1333      000040
1334      000020
1335      000010
1336      000004
1337      000002
1338      000001

; *
; MS.ATTN SUBCODES
; -
MSA.NOP = 0*2      ;NO-OP (NOTHING TRIGGERED)
MSA.VOL = 1*2      ;SIMULATE ON-LINE/OFF-LINE TRANSISTION
MSA.NRAM= 2*2      ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
MSA.FRAME= 3*2     ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)

; *
; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
; -
NP.IR      = BIT7      ;INTERRUPT REQUEST (0-1 TRANSITION)
NP.OUT     = BIT6      ;TAPE DATA DIRECTION OUT (0= IN)
NP.LOOP    = BIT5      ;ENABLE TRANSPORT LOOPBACK
NP.WRP     = BIT4      ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)

; *
; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
; -
S2.DIM     = BIT7      ;WORD #9 BYTE 2 DATA IN MISS
S2.ILW     = BIT6      ;      ILW H
S2.OURDY   = BIT5      ;      OUT RDY H
S2.INRDY   = BIT4      ;      IN RDY H
S2.ATIMR   = BIT3      ;      TIMER A FLAG H
S2.BTIMR   = BIT2      ;      TIMER B FLAG H
S2.UNDEF   = BIT1-BIT0 ;(UNDEFINED)

S1.PARIN   = BIT15     ;WORD #8 BYTE 1 PARIN H
S1.I2RESV  = BIT14     ;      IRESV2
S1.I1RESV  = BIT13     ;      IRESV1
S1.IEOT    = BIT12     ;      IEOT L
S1.IIDENT  = BIT11     ;      IIDENT H
S1.ICER    = BIT10     ;      ICER H
S1.IFMK    = BIT9      ;      IFMK H
S1.IHER    = BIT8      ;      IHER H

S0.ISPEED  = BIT7      ;WORD #8 BYTE 0 ISPEED H
S0.IRDY    = BIT6      ;      IRDY L
S0.IONL    = BIT5      ;      IONL L
S0.ILDPL   = BIT4      ;      ILDP L
S0.IDBY    = BIT3      ;      IDBY L
S0.IRWD    = BIT2      ;      IRWD L
S0.IFBL    = BIT1      ;      IFBL L
S0.IFPT    = BIT0      ;      IFPT L

```

K3

SPECIAL MACROS AND OPDEFS.

```

1340             .SBTTL SPECIAL MACROS AND OPDEFS.
1341
1342             ;*
1343             ;SAVE GENERAL REGS 1 TO 5
1344             ;-
1345
1346             .MACRO SAVREG
1347             JSR   R5,REGSAV
1348             .ENDM
1349
1350             ;*
1351             ; MACRO TO FORCE AN ERROR
1352             ;-
1353             .MACRO FORCERROR TAG,NOTSSR
1354             .NLIST
1355             .IIF NDF LISTALL, .NLIST
1356             .LIST
1357             .IF B NOTSSR
1358             MOV   TSSR(R5),R1      ;READ TSSR
1359             .ENDC
1360             MOV   FORCER,FORCER    ;IS FORCER SET? (LEAVE C BIT ALONE)
1361             BNE   TAG              ;BR IF YES
1362             .NLIST
1363             .IIF NDF LISTALL, .LIST
1364             .LIST
1365             .ENDM
1366
1367             ;*
1368             ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
1369             ; WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
1370             ; SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
1371             ; FORCER TO 17777
1372             ; TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
1373             ;-
1374             .MACRO FORCEEXIT TAG
1375             .NLIST
1376             .IIF NDF LISTALL, .NLIST
1377             .LIST
1378             MOV   FORCER,FORCER    ;IS FORCER NEGATIVE?
1379             BMI   TAG              ;BR IF YES
1380             .NLIST
1381             .IIF NDF LISTALL, .LIST
1382             .LIST
1383             .ENDM
1384             ;*
1385             ; MACRO TO INCREMENT ERROR COUNTS
1386             ;-
1387             .MACRO NEXT.ERRNO
1388             .NLIST
1389             ;;;.IIF NDF LISTALL, .NLIST
1390             ERRNO=ERRNO+1
1391             ;;;.IIF NDF LISTALL, .LIST
1392             .LIST
1393             .ENDM

```

L3

SPECIAL MACROS AND OPDEFS.

```
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406      000000
1407
1408
1409
1410
1411
1412
1413
1414 002176 000000
1415
1416
```

```

;
;MACRO TO PERFORM XOR
;-
      .MACRO XOR A,B
      MOV A,-(SP)
      BIC B,(SP)
      BIC A,B
      BIS (SP)+,B
      .ENDM
EN=0      ; INITIALIZE ERROR NUMBER
.SBTTL FORCER - FORCE ERROR FLAG

;
; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
;
FORCER:: 0      ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED -
; - BY THE MACRO "IFERROR"). AN ERROR NEED NOT -
; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.
```

GLOBAL DATA SECTION

.SBTTL GLOBAL DATA SECTION

```

1418
1419
1420
1421      ;**
1422      ;THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1423      ;IN MORE THAN ONE TEST.
1424      ;--
1425
1426      ;
1427      ;THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
1428      ;SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.
1429      ;
1429 002200 000000  EPRTSW::      .WORD 0      ;PRINT SWITCH
1430 002202 000000  UNITN::      .WORD 0      ;UNIT # UNDER TEST.
1431 002204 000000  QVP::      .WORD 0      ;QUICK VERIFY FLAG.
1432 002206 000000  CSRADDR::   .WORD 0      ;ADDRESS OF CSR FOR CURRENT DEVICE
1433 002210 000224  IVEC::      .WORD 224    ;INTERRUPT VECTOR
1434 002212 000200  IPRI::      .WORD PRI04  ;INTERRUPT PRIORITY.
1435 002214 000000  TSTCNT::   .WORD 0      ;NUMBER OF TESTS RUN IN THIS PASS
1436 002216 000000  LOOPCNT::  .WORD 0      ;REMAINING ITERATION COUNT FOR TEST
1437 002220 000000  DEVCNT::   .WORD 0      ;NUMBER OF DEVICE UNDER TEST
1438 002222 000000  FATFLG::   .WORD 0      ;SET IF FATAL ERROR IS DETECTED IN TEST
1439 002224 000000  INTRECV::  .WORD 0      ;SET IF TAPE INTERRUPT WAS RECEIVED
1440 002226 000000  EXTFEA::   .WORD 0      ;EXTENDED FEATURES SOFTWARE SW 0=OFF;1=ON
1441 002230 000000  REV::      .WORD 0      ;REVISION LEVEL
1442 002232 000000  BENBSW::   .WORD 0      ;BUFFER ENABLE SWITCH SW 0=OFF;1=ON
1443 002234 000000  EXPD::     .WORD 0      ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
1444 002236 000000  RECV::     .WORD 0      ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
1445 002240 000000  ERRHI::    .WORD 0      ;HIGH ADDRESS MEMORY ERROR
1446 002242 000000  ERRLO::    .WORD 0      ;LOW ADDRESS MEMORY ERROR
1447 002244 000000  RAMDATA::  .BLKW 16.    ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
1448 002304 000000  RAMSIZ::   .WORD 0      ;RAM DATA SIZE FOR PRAMPKT ROUTINE
1449 002306 000000  RCVHIADD::.WORD 0      ;RECEIVED BUFFER HIGH ADDRESS
1450 002310 000000  RCVLOADD::.WORD 0      ;RECEIVED BUFFER LOW ADDRESS
1451 002312 000000  COUNT::    .WORD 0      ;TEST COUNT PATTERN
1452 002314 000000  DATA::    .WORD 0      ;TEST DATA
1453 002316 000000  TSTFLAG::  .WORD 0      ;TEST FLAG WORD
1454 002320 000000  TSTPTR::   .WORD 0      ;TSTBLK POINTER
1455 002322 000000  PRMNO::    .WORD 0      ;PRINT ROUTINE TEMP
1456 002324 000000  EXPMSG::   .BLKB 100.   ;EXPECTED MESSAGE BUFFER DATA
1457 002470 000000  RECMSG::   .BLKB 100.   ;RECEIVED MESSAGE BUFFER DATA
1458 002634 000000  TMPBFR::   .BLKB 80.    ;TEMPORARY STORAGE FOR PRINT

```


N3

TSTBLK - TEST DATA TABLE

.SBTTL TSTBLK - TEST DATA TABLE

1460
 1461
 1462
 1463
 1464
 1465
 1466
 1467
 1468
 1469
 1470
 1471
 1472
 1473
 1474
 1475
 1476 002754
 1477 002754 000000
 1478 002756 177777
 1479 002760 000001
 1480 002762 000002
 1481 002764 000004
 1482 002766 000010
 1483 002770 000020
 1484 002772 000040
 1485 002774 000100
 1486 002776 000200
 1487 003000 000400
 1488 003002 001000
 1489 003004 002000
 1490 003006 004000
 1491 003010 010000
 1492 003012 020000
 1493 003014 040000
 1494 003016 100000
 1495 003020 177776
 1496 003022 177775
 1497 003024 177773
 1498 003026 177767
 1499 003030 177757
 1500 003032 177737
 1501 003034 177677
 1502 003036 177577
 1503 003040 177377
 1504 003042 176777
 1505 003044 175777
 1506 003046 173777
 1507 003050 167777
 1508 003052 157777
 1509 003054 137777
 1510 003056 077777
 1511 003060 125252
 1512 003062 052525
 1513 003064

```

;*
; 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
;
; -

```

```

TSTBLK::
.WORD 0 ;ALL ZEROS
.WORD 177777 ;ALL ONES
.WORD BIT0 ;DATA FOR WALKING ONES
.WORD BIT1
.WORD BIT2
.WORD BIT3
.WORD BIT4
.WORD BIT5
.WORD BIT6
.WORD BIT7
.WORD BIT8
.WORD BIT9
.WORD BIT10
.WORD BIT11
.WORD BIT12
.WORD BIT13
.WORD BIT14
.WORD BIT15
.WORD †CBIT0 ;DATA FOR WALKING ZEROS
.WORD †CBIT1
.WORD †CBIT2
.WORD †CBIT3
.WORD †CBIT4
.WORD †CBIT5
.WORD †CBIT6
.WORD †CBIT7
.WORD †CBIT8
.WORD †CBIT9
.WORD †CBIT10
.WORD †CBIT11
.WORD †CBIT12
.WORD †CBIT13
.WORD †CBIT14
.WORD †CBIT15
.WORD 125252 ;ALTERNATING ONES, ZEROS
.WORD 052525 ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE

TBLEND==.

```

GLOBAL ENVIRONMENT STORAGE

```

1515          .SBTTL GLOBAL ENVIRONMENT STORAGE
1516          ;
1517          ; STORAGE FOR DEVICE REGISTERS
1518          ;
1519 003064 000000 100000 000000 DUMMY: 0,100000,0,0 ; DUMMY DEVICE REGISTERS...
1520 003074 000000 000000 000000      0,0,0,0,0,0,0,0 ; ...FOR MULTI-UNIT CHECKOUT.
1521          ;
1522          ;
1523 003114 000000 DUFLG::          .WORD 0          ; "DROPPED UNIT" FLAG.
1524          ; INHIBITS CODE IN "CLEAN-UP".
1525 003116 000000 NODEV::          .WORD 0          ; FLAG TO SAY NO DEVICE.
1526          ;
1527 003120 000000 TEMP1::          .WORD 0          ; SOME TEMP LOCATIONS.
1528 003122 000000 TEMP2::          .WORD 0
1529 003124 000000 XXCOMM::          .WORD 0          ; XXDP* COMM BLOCK POINTER.
1530 003126 000000 FREE::          .WORD 0          ; 1ST FREE MEMORY ADDRESS...
1531 003130 000000 FRESIZ::          .WORD 0          ; ...AND SIZE (IN WORDS).
1532 003132 000000 FREEHI: .WORD 0          ; LAST WORD IN FREE SPACE
1533 003134 000000 KTFLG::          .WORD 0          ; KT11, MEM AVAIL FLAG -
1534          ; - .WORD 0 = <24K OR NO KT -
1535          ; - NZ = >24K AND KT.
1536 003136 000000 KTENABLE::          .WORD 0          ; SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
1537 003140 000000 NXMFLG::          .WORD 0          ; SET IF WE CAN TEST CLEARED OTHERWISE
1538 003142 000000 NXMLO::          .WORD 0          ; NXM LO ADDRESS BITS
1539 003144 000000 NXMHI::          .WORD 0          ; NXM HI ADDRESS BITS FOR DAL'S 16-21
1540 003146 000000 T23A::          .WORD 0          ; 11/23A FLAG
1541 003150 000000 T23B::          .WORD 0          ; 11/23B FLAG
1542 003152 000000 T3BFLG::          .WORD 0          ; TEST 3B FLAG +0
1543 003154 002000 PST32W::          .WORD 2000          ; 32W BLOCK ADDRESS FOR 32K START
1544 003156 000000 SIFLAG::          .WORD 0
1545 003160 000000 BADDAT::          .WORD 0          ; ACTUAL DATA
1546 003162 000000 GDDAT::          .WORD 0          ; EXPECTED DATA
1547 003164 000000 LOOPFL::          .WORD 0
1548 003166          CTAB::          ; CONFIGURATION TABLES.
1549 003166 000000 CTABM::          .WORD 0          ; CONFIG WORK.
1550 003170          .WORD 0
1551 003172          .WORD 0
1552 003174          .WORD 0
1553 003176 177777          .WORD -1          ; END OF MEM TABLE.
1554 003200          CTABE::
1555          ; ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
1556          ;
1557          ; 0 = UNIT NOT TESTED
1558          ; 100000 = UNIT ONLINE, NO ERRORS
1559          ; 10XXXX = UNIT ONLINE, ENCOUNTERED XXXX ERRORS
1560          ; 160000 = UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
1561          ; 160001 = UNIT DROPPED, NOT IDLE AT START
1562          ; 14XXXX = UNIT DROPPED, ENCOUNTERED XXXX ERRORS
1563          ;
1564 003200          ERTABL:          .BLKW 64.
1565 003400 000000          ERTABE:          .WORD 0
1566          ;
1567 003402 000000          SKIPT:          .WORD 0          ; 1=SKIP SUBTEST 0=NO SKIP OF SUBTEST

```

GLOBAL TEXT MESSAGES

1569
 1570
 1571
 1572
 1573
 1574
 1575
 1576
 1577
 1578 003404
 003404
 003404 124 123 126

1579
 1587
 1588
 1589
 1590 003412
 003412
 003412 052 052 052

1591
 1605
 1606
 1607
 1608
 1609 003504 003544 003547 003553
 1610 003524 003605 003611 003615
 1611 003544 123 103 000
 1612 003547 102 111 105
 1613 003553 123 103 105
 1614 003557 122 115 122
 1615 003563 116 130 115
 1616 003567 116 102 101
 1617 003573 102 111 124
 1618 003600 102 111 124
 1619 003605 123 123 122
 1620 003611 117 106 114
 1621 003615 102 111 124
 1622 003622 102 111 124
 1623 003627 102 111 124
 1624 003634 102 111 124
 1625 003641 102 111 124
 1626 003646 102 111 124
 1627
 1628 003654 124 123 123
 1629 003707 124 123 123
 1630 003742 040 040 116
 1631 004001 045 101 040
 1632 004022 045 101 040
 1633 004062 045 101 040
 1634 004121 045 116 045
 1635 004125 040 040 125
 1636 004154 040 040 111
 1637 004217 045 116 045
 1638 004223 040 040 116
 1639 004260 040 040 111

```

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

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

; *
; BIT TO ASCII CONVERSION FOR TSSR REGISTER
; -
TSSRBIT::
      .WORD 1$,2$,3$,4$,5$,6$,7$,8$
      .WORD 9$,10$,11$,12$,13$,14$,15$,16$
1$: .ASCIZ 'SC'
2$: .ASCIZ 'BIE'
3$: .ASCIZ 'SCE'
4$: .ASCIZ 'RMR'
5$: .ASCIZ 'NXM'
6$: .ASCIZ 'NBA'
7$: .ASCIZ 'BIT9'
8$: .ASCIZ 'BIT8'
9$: .ASCIZ 'SSR'
10$: .ASCIZ 'OFL'
11$: .ASCIZ 'BIT5'
12$: .ASCIZ 'BIT4'
13$: .ASCIZ 'BIT3'
14$: .ASCIZ 'BIT2'
15$: .ASCIZ 'BIT1'
16$: .ASCIZ 'BIT0'
      .EVEN
SFIERR: .ASCIZ 'TSSR ERROR AFTER SOFT INIT'
SFHERR: .ASCIZ 'TSSR ERROR AFTER BUS RESET'
NXR: .ASCIZ / NON-EXISTANT DEVICE REGISTER/
NXRX: .ASCIZ /#A ADDRESS: #06/
TSSX: .ASCII /#A TSBA,TSSR EXP'D: #06#A,#06#N/
      .ASCIZ /#A TSBA,TSSR REC'D: #06#A,#06#N/
FUSI: .ASCII /#N#A/
USI: .ASCIZ / UNEXPECTED INTERRUPT/
NSI: .ASCIZ / INTERRUPT EXPECTED, NOT RECEIVED/
FNOINTR: .ASCII /#N#A/
NOINTR: .ASCIZ / NO INTERRUPT WAS GENERATED/
IFALT: .ASCIZ / INTERRUPT FAULT/

```

GLOBAL TEXT MESSAGES

```

1640 004302 045 101 040 INTX: .ASCIZ /%A CPU PC: %06%A TSBA: %06/
1641 004337 040 040 042 NOINIT: .ASCIZ / "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
1642 004411 040 040 042 NSINIT: .ASCIZ / "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
1643 004461 040 040 042 BRINIT: .ASCIZ / "BUS-RESET" DIDN'T INITIALIZE THE DPU/
1644 004531 000 040 042 NUL: .ASCIZ //
1645 004532 045 116 000 NULCR: .ASCIZ /%N/
1646 004535 045 101 040 EXPGOT: .ASCIZ /%A EXP'D: %06%A, REC'D: %06/
1647 004571 045 116 045 EXPGT2: .ASCIZ /%N%A EXP'D: %06%A, %06%N%A REC'D: %0%A, %06/
1648 004645 045 101 040 DUAD12: .ASCIZ /%A REG(W) WRITTEN TO: %06%A REG(R) READ; EXP'D: %06%A, REC'D: %06/
1649 004747 122 101 115 PKTRAM: .ASCIZ 'RAM Contents Do Not Match Packet Sent'
1650 005015 040 040 103 SCME: .ASCIZ / CONFIG DOESN'T MATCH MFG. MASTER/
1651 005060 127 122 111 WRTMSG: .ASCIZ 'WRITE CHARACTERISTICS Failed'
1652 005115 124 123 123 WRTERR: .ASCIZ 'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
1653 005210 124 123 123 RDERR: .ASCIZ 'TSSR Incorrect After READ Command, More Bits Set Than SSR'
1654 005302 106 101 124 SCHERR: .ASCIZ 'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
1655 005374 105 122 122 RETERR: .ASCIZ 'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
1656 005462 045 116 045 NOMEM: .ASCIZ '%N%A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****N'
1657 005556 045 116 045 M8186: .ASCIZ '%N%A ***** 11/23A SYSTEM *****N'
1658 005647 045 116 045 M8189: .ASCIZ '%N%A ***** 11/23B SYSTEM *****N'

```

```

1659 .EVEN
1660 .SBTTL GLOBAL ERROR REPORT SECTION
1661
1662
1663
1664
1665
1666

```

```

; **
; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
; CALLS THAT ARE USED IN MORE THAN ONE TEST.
; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
; --

```

```

1667 005740
005740
1668 005740
005740 013746 003116
005744 012746 004001
005750 012746 000002
005754 010600
005756 104415
005760 062706 000006
1669 005764 004737 005772
1670 005770
005770
005770 104423
1671
1672
1673
1674
1675 005772 005727
1676 005774 000000
1677 005776 001402
1678 006000 004777 177770
1679 006004
006004 012746 004532
006010 012746 000001
006014 010600
006016 104415
006020 062706 000004
1680 006024 000207

```

```

BGNMSG NXRRERR ;NON-EXISTANT DEVICE REGISTER.
NXRRERR: PRINTX #NXRX,NODEV ;NODEV = NEXM ADDRESS.
MOV NODEV,-(SP)
MOV #NXRX,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #6,SP
JSR PC,EXTEND ; PRINT EXTENSION IF REQUIRED.
ENDMSG

```

```

L10002: TRAP C#MSG
;
; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
; TO ANY OF THE ABOVE ERROR SIGNATURES.
;

```

```

EXTEND: TST (PC)+
EXTA: 0 ; 0 = NO EXTENSION.
BEQ 14
JSR PC,@EXTA ; APPEND EXTENSION TEXT.
14: PRINTX #NULCR ; PRINT A BLANK LINE
MOV #NULCR,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #4,SP
RTS PC

```

PRITSSR - PRINT TSSR CONTENTS

```

1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700 006026
1701 006026
1702 006032 010104
1703 006034
      006034 010446
      006036 012746 006417
      006042 012746 000002
      006046 010600
      006050 104414
      006052 062706 000006
1704 006056 010400
1705 006060 004737 016104
1706 006064 103410
1707 006066
      006066 012746 006637
      006072 012746 000001
      006076 010600
      006100 104415
      006102 062706 000004
1708 006106 010403
1709 006110 042703 001476
1710 006114 001434
1711 006116 012702 002634
1712 006122 012701 003504
1713 006126 005703
1714 006130 001413
1715 006132 000241
1716 006134 006103
1717 006136 103006
1718 006140 011100
1719 006142 112022
1720 006144 001376
1721 006146 112762 000054 177777
1722 006154 005721
1723 006156 000763
1724 006160 105042
1725 006162
      006162 012746 002634
      006166 012746 006610

```

.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
  ADD @6,SP
  MOV R4,R0             ;GET TSSR BACK FOR CHKAMB
  JSR PC,CHKAMB        ;ARE CONTENTS AMBIGUOUS ?
  BCS 5$               ;BRANCH IF NOT
                       ;SHOW CONTENTS ARE AMBIGUOUS
  PRINTX @AMBTSSR
  MOV @AMBTSSR,-(SP)
  MOV @1,-(SP)
  MOV SP,R0
  TRAP C#PNTX
  ADD @4,SP
5$: MOV R4,R3           ;CONTENTS OF TSSR
   BIC @HIADDR!FATERR!TERCLS,R3 ;CLEAR ALL MULTIPLE BIT FIELDS
   BEQ 20$             ;NO BITS ARE SET
   MOV @TMPBFR,R2     ;TEMPORARY ASCII BUFFER
   MOV @TSSRBIT,R1   ;ASCII EQUIVALENT OF BITS
10$: TST R3           ;REMAINING BITS TO CONVERT
   BEQ 15$           ;BRANCH WHEN ALL ARE DONE
   CLC               ;CLEAR CARRY FOR SHIFT
   ROL R3            ;SHIFT NEXT BIT TO CARRY
   BCC 13$          ;BRANCH IF BIT NOT SET
   MOV (R1),R0      ;POINTER TO BIT DEFINITION
11$: MOVB (R0),-(R2) ;MOVE ASCII TO BUFFER
   BNE 11$          ;MOVE ALL BITS
   MOVB @'-1(R2)    ;INSERT A COMMA TO TERMINATE
13$: TST (R1)       ;POINT TO NEXT DESCRIPTION
   BR 10$          ;GET THE REMAINING BITS
15$: CLRB -(R2)    ;TERMINATE THE LINE
   PRINTX @TSSDEF,@TMPBFR ;PRINT THE BIT DEFINITIONS
   MOV @TMPBFR,-(SP)
   MOV @TSSDEF,-(SP)

```

PRITSSR - PRINT TSSR CONTENTS

```

006172 012746 000002      MOV      #2,-(SP)
006176 010600      MOV      SP,R0
006200 104415      TRAP    C#PNTX
006202 062706 000006      ADD     #6,SP
1726
1727 006206 010403      20$:   MOV     R4,R3           ;GET THE TSSR CONTENTS
1728 006210 042703 177761      BIC     #1CTERCLS,R3    ;CLEAR ALL BUT TERMINATION
1729 006214 016303 006700      MOV     TCOCOD(R3),R3   ;GET THE TERMINATION CODE MEANING
1730 006220      PRINTX #TCOASC,R3      ;PRINT THE TERMINATION CODE
      006220 010346      MOV     R3,-(SP)
      006222 012746 006500      MOV     #TCOASC,-(SP)
      006226 012746 000002      MOV     #2,-(SP)
      006232 010600      MOV     SP,R0
      006234 104415      TRAP    C#PNTX
      006236 062706 000006      ADD     #6,SP
1731 006242 010403      MOV     R4,R3           ;TSSR CONTENTS AGAIN
1732 006244 042703 177717      BIC     #1CFATERR,R3   ;CLEAR ALL BUT FATAL TERMINATION
1733 006250 001416      BEQ     25$             ;DON'T PRINT IF ZERO
1734 006252 006203      ASR     R3
1735 006254 006203      ASR     R3
1736 006256 006203      ASR     R3           ;ALINE TERMINATION CODE FOR INDEX
1737 006260 016303 007240      MOV     TSFCOD(R3),R3  ;GET THE FATAL TERMINATION CODE
1738 006264      PRINTX #TFCASC,R3     ;PRINT THE FATAL TERMINATION CODE
      006264 010346      MOV     R3,-(SP)
      006266 012746 006541      MOV     #TFCASC,-(SP)
      006272 012746 000002      MOV     #2,-(SP)
      006276 010600      MOV     SP,R0
      006300 104415      TRAP    C#PNTX
      006302 062706 000006      ADD     #6,SP
1739 006306 042704 176377      25$:   BIC     #1CHIADDR,R4   ;CLEAR ALL BUT EXTENDED ADDRESS
1740 006312 001411      BEQ     30$             ;DON'T PRINT IF ZERO
1741 006314      PRINTX #TEXASC,R4     ;PRINT THE EXTENDED ADDRESS BITS
      006314 010446      MOV     R4,-(SP)
      006316 012746 006437      MOV     #TEXASC,-(SP)
      006322 012746 000002      MOV     #2,-(SP)
      006326 010600      MOV     SP,R0
      006330 104415      TRAP    C#PNTX
      006332 062706 000006      ADD     #6,SP
1742 006336 013703 002200      30$:   MOV     EPRTSW,R3      ;PRINT MESSAGE BUFFER ADDRESS
1743 006342      PRINTX R3             ;PRINT PROPER MESSAGE
      006342 010346      MOV     R3,-(SP)
      006344 012746 000001      MOV     #1,-(SP)
      006350 010600      MOV     SP,R0
      006352 104415      TRAP    C#PNTX
      006354 062706 000004      ADD     #4,SP
1744 006360 000207      RTS     PC             ;RETURN TO CALLER

```

PRITSSR - PRINT TSSR CONTENTS

1751	006362				EPRT2:				
1752	006362	045	116	045	EPRT1:	.ASCIZ	'#N#A *****REPLACE M7196*****'		
1753									
1763	006417	045	116	045	TSSRFOR:	.ASCIZ	'#N#A TSSR = #06'		
1764	006437	045	116	045	TEXASC:	.ASCIZ	'#N#A Extended Address Bits = #06'		
1765	006500	045	116	045	TCOASC:	.ASCIZ	'#N#A Termination Class Code = #T'		
1766	006541	045	116	045	TFCASC:	.ASCIZ	'#N#A Fatal Termination Class Code = #T'		
1767	006610	045	116	045	TSSDEF:	.ASCIZ	'#N#A TSSR Bits Set: #T'		
1768	006637	045	116	045	AMBTSSR:	.ASCIZ	'#N#A TSSR Contents Are Ambiguous'		
1769									
1770	006700	006720	006743	006771	TCOCOD:	.WORD	1#,2#,3#,4#,5#,6#,7#,8#		
1771	006720	116	157	162	1#:	.ASCIZ	'Normal Termination'		
1772	006743	124	145	162	2#:	.ASCIZ	'Termination Condition'		
1773	006771	124	141	160	3#:	.ASCIZ	'Tape Status Alert'		
1774	007013	106	165	156	4#:	.ASCIZ	'Function Reject'		
1775	007033	122	145	143	5#:	.ASCIZ	'Recoverable Error - Tape Position One Record Down'		
1776	007115	122	145	143	6#:	.ASCIZ	'Recoverable Error - Tape Was Not Moved'		
1777	007164	125	156	162	7#:	.ASCIZ	'Unrecoverable Error'		
1778	007210	106	141	164	8#:	.ASCIZ	'Fatal Controller Error'		
1779						.EVEN			
1780									
1781	007240	007250	007304	007315	TSFCOD:	.WORD	1#,2#,3#,4#		
1782	007250	111	156	164	1#:	.ASCIZ	'Internal Diagnostic Failure'		
1783	007304	122	145	163	2#:	.ASCIZ	'Reserved'		
1784	007315	102	165	163	3#:	.ASCIZ	'Bus Interface or Sanity Check Error'		
1785	007361	122	145	163	4#:	.ASCIZ	'Reserved'		
1786						.EVEN			

PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

1788 .SBTTL PRIPKT -PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

1789 ;*
1790 ;THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
1791 ;THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
1792 ;
1793 ;INPUT:

1794 ;
1795 ;
1796 ; R0 NUMBER OF WORDS IN PACKET
1797 ; R3 HIGH ORDER COMMAND PACKET ADDRESS
1798 ; R4 ADDRESS OF COMMAND PACKET
1799 ;
1800 ; NOTE: R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
1801 ;-

1802
1803 007372 PRIPKT::
1804 007372 SAVREG ;SAVE THE REGISTERS
1805 007376 010005 MOV R0,R5 ;SAVE NO. OF WORDS IN PACKET
1806 007400 005737 003136 TST KTENABLE ;ABOVE 28K UNDER TEST?
1807 007404 001001 BNE 10\$;BR IF YES
1808 007406 005003 CLR R3 ;SET HIGH ORDER ADDRESS TO 0
1809 007410 010301 10\$: MOV R3,R1 ;COPY HIGH ORDER ADDRESS
1810 007412 010400 MOV R4,R0 ;GET LOWER ADDRESS
1811 007414 006100 ROL R0 ;SHIFT BIT 15 INTO C BIT
1812 007416 006101 ROL R1 ;AND INTO HIGH ORDER.
1813 007420 PRINTB #PKTADD,R1,R4 ;PRINT PACKET ADDRESS
1814 007420 010446 MOV R4,-(SP)
1815 007422 010146 MOV R1,-(SP)
1816 007424 012746 007556 MOV #PKTADD,-(SP)
1817 007430 012746 000003 MOV #3,-(SP)
1818 007434 010600 MOV SP,R0
1819 007436 104414 TRAP C#PNTB
1820 007440 062706 000010 ADD #10,SP
1821 007444 010300 15\$: MOV R3,R0 ;GET HIGH ORDER ADDRESS
1822 007446 001404 BEQ 20\$;BR IF NOT ABOVE 28K.
1823 007450 010401 MOV R4,R1 ;GET LOW ORDER ADDRESS
1824 007452 004737 017356 JSR PC,SETMAP ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
1825 007456 010004 MOV R0,R4 ;GET RETURNED PAR6 ADDRESS BIAS
1826 007460 005001 20\$: CLR R1 ;SAVE WORD NUMBER
1827 007462 012402 25\$: MOV (R4),R2 ;GET PACKET CONTENTS
1828 007464 PRINTB #PKTFRM,R1,R2 ;PRINT THE DATA
1829 007464 010246 MOV R2,-(SP)
1830 007466 010146 MOV R1,-(SP)
1831 007470 012746 007520 MOV #PKTFRM,-(SP)
1832 007474 012746 000003 MOV #3,-(SP)
1833 007500 010600 MOV SP,R0
1834 007502 104414 TRAP C#PNTB
1835 007504 062706 000010 ADD #10,SP
1836 007510 005201 INC R1 ;NEXT WORD NUMBER
1837 007512 020105 CMP R1,R5 ;DONE ALL PACKET WORDS?
1838 007514 002762 BLT 25\$;LOOP TILL ALL DONE
1839 007516 000207 RTS PC ;RETURN
1840 007520 045 116 045 PKTFRM: .ASCIZ '#N#A Packet Word #D1#A = #06#'
1841 007556 045 116 045 PKTADD: .ASCIZ '#N#A Packet Address = #01#05#'
1842 .EVEN

PRIBXOR - PRINT EXPD, RECV AND XOR BYTE

```

1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846 007614
1847 007614
1848 007620 010203
1849 007622
1850 007632 012700 177400
1851 007636 040001
1852 007640 040002
1853 007642 040003
1854 007644
    007644 010346
    007646 010146
    007650 010246
    007652 012746 007676
    007656 012746 000004
    007662 010600
    007664 104414
    007666 062706 000012
1855 007672 010300
1856 007674 000207

```

.SBTTL PRIBXOR - PRINT EXPD, RECV AND XOR BYTE

```

;*
;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.

```

;INPUTS:

```

;      R1      RECEIVED DATA
;      R2      EXPECTED DATA

```

;OUTPUT:

```

;      R0      XOR OF EXPECTED/RECEIVED DATA

```

PRIBXOR::

```

    SAVREG                ;SAVE THE REGISTERS
    MOV     R2,R3          ;EXPECTED DATA
    XOR     R1,R3          ;FORM THE EXCLUSIVE OR
    MOV     #1C<377>,R0    ;BYTE MASK
    BIC     R0,R1          ;SAVE LOW BYTE RECV
    BIC     R0,R2          ;SAVE LOW BYTE EXPD
    BIC     R0,R3          ;SAVE LOW BYTE XOR
    PRINTB @XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
    MOV     R3,-(SP)
    MOV     R1,-(SP)
    MOV     R2,-(SP)
    MOV     @XORBFOR,-(SP)
    MOV     #4,-(SP)
    MOV     SP,R0
    TRAP   C#PNTB
    ADD    #12,SP
    MOV    R3,R0
    RTS    PC              ;RO HAS XOR ON RETURN
                          ;RETURN TO CALLER

```

```

1858 007676 045 116 045 XORBFOR: .ASCIZ '#N#A EXPD: #03#A RECV: #03#A XOR: #03'

```

```

.EVEN
.SBTTL PRIBXOR - PRINT EXPD, RECV AND XOR

```

```

;*
;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.

```

;INPUTS:

```

;      R1      RECEIVED DATA
;      R2      EXPECTED DATA

```

;OUTPUT:

```

;      R0      XOR OF EXPECTED/RECEIVED DATA

```

PRIBXOR::

```

    SAVREG                ;SAVE THE REGISTERS
    MOV     R2,R3          ;EXPECTED DATA
    XOR     R1,R3          ;FORM THE EXCLUSIVE OR
    PRINTB @XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE

```

```

1875 007744
1876 007744
1877 007750 010203
1878 007752
1879 007762

```

J4

PRIXOR - PRINT EXPD, RECV AND XOR

```

007762 010346            MOV    R3,-(SP)
007764 010146            MOV    R1,-(SP)
007766 010246            MOV    R2,-(SP)
007770 012746 010014     MOV    @XORFOR,-(SP)
007774 012746 000004     MOV    @4,-(SP)
010000 010600            MOV    SP,R0
010002 104414            TRAP   C$PNTB
010004 062706 000012     ADD    @12,SP
1880 010010 010300       MOV    R3,R0                ;R0 HAS XOR ON RETURN
1881 010012 000207       RTS    PC                 ;RETURN TO CALLER
1882
1883 010014        045       116       045 XORFOR: .ASCIZ 'N%A EXPD: #06%A RECV: #06%A XOR: #06'
1884                .EVEN

```

PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT

1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899

.SBTTL PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT

;*
; ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
; THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
; INPUTS:
; R0 OCTAL VALUE TO CONVERT
; R1 TABLE OF POINTERS TO ASCII EQUIVALENT
;-

1900 010062
1901 010062
1902 010066 000207
1903

PRIEQU: SAVREG ;SAVE THE REGISTERS
RTS PC ;RETURN TO CALLER

1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914

.SBTTL PRIRAM - PRINT RAM ADDRESS

;*
; PRINT CONTROLLER RAM ADDRESS.
; THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
; INPUTS:
; R4 RAM ADDRESS
;-

1915 010070
1916 010070
1917 010074
010074 010446
010076 012746 010120
010102 012746 000002
010106 010600
010110 104414
010112 062706 000006
1918 010116 000207

PRIRAM: SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
PRINTB #RAMFOR,R4 ;PRINT RAM ADDRESS IN ERROR
MOV R4,-(SP)
MOV #RAMFOR,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C#PNTB
ADD #6,SP
RTS PC ;RETURN

1919
1920 010120 045 116 045 RAMFOR: .ASCIZ '#N#A CONTROLLER RAM ADDRESS = #06'
1921 .EVEN

PRIADD - PRINT MEMORY ERROR ADDRESS

```

1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935 010162
1936 010162
1937 010166 013700 002240
1938 010172 013701 002242
1939 010176 010102
1940 010200 006101
1941 010202 006100
1942 010204
      010204 010246
      010206 010046
      010210 012746 010232
      010214 012746 000003
      010220 010600
      010222 104414
      010224 062706 000010
1943 010230 000207
1944
1945 010232 045 116
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960 010276
1961 010276
1962 010302 013702 002240
1963 010306 013701 002242
1964
1965
1966
1967 010312
      010312 010146
      010314 012746 010360
      010320 012746 000002
      010324 010600
      010326 104414

```

```

.SBTTL PRIADD - PRINT MEMORY ERROR ADDRESS
;*
;PRINT MEMORY ADDRESS
;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
;
; IMPLICIT INPUTS
;
; ERRHI - HIGH ORDER ADDRESS
; ERRLO - LOW ORDER ADDRESS
;
;-
PRIADD:
  SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
  MOV ERRHI,R0                          ;GET HIGH ADDRESS
  MOV ERRLO,R1                          ;GET LOW ADDRESS
  MOV R1,R2                             ;COPY LOW ADDRESS
  ROL R1                                 ;SHIFT BIT 15 TO C BIT
  ROL R0                                 ;SHIFT INTO HIGH ORDER
  PRINTB @PRIA0,R0,R2                   ;PRINT MEMORY ADDRESS IN ERROR
  MOV R2,-(SP)
  MOV R0,-(SP)
  MOV @PRIA0,-(SP)
  MOV #3,-(SP)
  MOV SP,R0
  TRAP C#PNTB
  ADD #10,SP
  RTS PC                                ;RETURN

```

```

045 PRIA0: .ASCIZ '#N#A MEMORY ERROR ADDRESS = #01#05'
          .EVEN

```

.SBTTL PRITADD - PRINT MEMORY TEST ADDRESS

```

1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960 010276
1961 010276
1962 010302 013702 002240
1963 010306 013701 002242
1964
1965
1966
1967 010312
      010312 010146
      010314 012746 010360
      010320 012746 000002
      010324 010600
      010326 104414

```

```

;*
;PRINT MEMORY ADDRESS
;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
;
; IMPLICIT INPUTS
;
; ERRHI - HIGH ORDER ADDRESS
; ERRLO - LOW ORDER ADDRESS
;
;-
PRITADD:
  SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
  MOV ERRHI,R2                          ;GET HIGH ADDRESS
  MOV ERRLO,R1                          ;GET LOW ADDRESS
  MOV R1,R2                             ;COPY LOW ADDRESS
  ROL R1                                 ;SHIFT BIT 15 TO C BIT
  ROL R0                                 ;SHIFT INTO HIGH ORDER
  PRINTB @PRIT0,R1                      ;PRINT MEMORY ADDRESS LOW IN ERROR
  MOV R1,-(SP)
  MOV @PRIT0,-(SP)
  MOV #2,-(SP)
  MOV SP,R0
  TRAP C#PNTB

```

M4

PRITADD - PRINT MEMORY TEST ADDRESS

```

1968 010330 062706 000006      ADD    #6,SP
      010334                PRINTB  #PRIT1,R2      ;PRINT MEMORY ADDRESS HIGH IN ERROR
      010334 010246      MOV    R2,-(SP)
      010336 012746 010423      MOV    #PRIT1,-(SP)
      010342 012746 000002      MOV    #2,-(SP)
      010346 010600      MOV    SP,R0
      010350 104414      TRAP  C#PNTB
1969 010352 062706 000006      ADD    #6,SP
      010356 000207      RTS    PC      ;RETURN
1970
1971 010360      045      116      045 PRIT0: .ASCIZ '#N#A MEMORY TEST ADDRESS LOW = #06'
1972 010423      045      116      045 PRIT1: .ASCIZ '#N#A MEMORY TEST ADDRESS HIGH = #06'
1973                .EVEN

```

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

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

1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009

```

;*
;ROUTINE TO ISSUE A SPACE RECORDS
;COMMAND (FORWARD OR REVERSE)
;INPUT:
;
;   R3   NUMBER OF RECORDS TO BE SPACED OVER
;        BIT15 CONTROLS DIRECTION
;        BIT15 = 0 IS FORWARD
;        BIT15 = 1 IS REVERSE
;   R5   FIRST DEVICE UNIBUS ADDRESS
;
;   REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
;OUTPUT:
;
;   CARRY SET - SPACE RECORDS COMMAND OK
;         CLR - SPACE RECORDS FAILED
;
;   R0   THE CONTENTS OF R4 IS MOVED TO R0
;
;IMPLICIT OUTPUT:
;
;   TAPE HAS BEEN MOVED
;SIDE EFFECTS:
;
;-
SPACE::
    SAVREG                                ;SAVE THE GENERAL REGISTERS
    MOV     #500.,SDELAY                  ;SET UP DELAY
    MOV     #140010,80#                   ;SET UP COMMAND, SPACE FORWARD
    TST     R3                             ;CHECK FOR DIRECTION
    BMI     5#                             ;BR, IF REVERSE INDICATED
    MOV     R3,90#                         ;LOAD UP NUMBER OF RECORDS TO SPACE
    BR     10#                             ;GO DO COMMAND
    BIC     #BIT15,R3                      ;CLEAR DIRECTION BIT
    MOV     R3,90#                         ;LOAD UP NUMBER OF RECORDS TO SPACE
    BIS     #BIT8,80#                      ;SET REVERSE BIT IN COMMAND PACKET
    MOV     #80#,R4                        ;SET UP R4 WITH PACKET ADDRESS
    MOV     R4,TSDB(R5)                   ;SEND OUT COMMAND
    JSR     PC,WAITF                       ;WAIT FOR SSR
    BCS     20#                             ;BR, IF SSR IS SET AND OK
    DELAY   250                            ;DELAY ABOUT .25 SECONDS
    MOV     #250,(PC)-
    .WORD   0
    MOV     L#DLY,(PC)-
    .WORD   0
    DEC     -6(PC)
    BNE     .-4

```

2010	010470			
2011	010470			
2012	010474	012737	000764	010660
2013	010502	012737	140010	010650
2014	010510	005703		
2015	010512	100403		
2016	010514	010337	010652	
2017	010520	000407		
2018	010522	042703	100000	
2019	010526	010337	010652	
2020	010532	052737	000400	010650
2021	010540	012704	010650	
2022	010544	010465	000000	
2023	010550	004737	016310	
2024	010554	103420		
2025	010556			
	010556	012727	000250	
	010562	000000		
	010564	013727	002116	
	010570	000000		
	010572	005367	177772	
	010576	001375		

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

```

010600 005367 177756          DEC    -22(PC)
010604 001367                BNE    .-20
2026 010606 005337 010660    DEC    SDELAY                ;BUMP DELAY COUNTER DOWN
2027 010612 001356                BNE    15$                   ;BR, IF MORE DELAY
2028 010614 000411                BR     60$                   ;BR IF TROUBLE CARRY = CLEAR
2029 010616 016501 000002    20$:  MOV    TSSR(R5),R1      ;READ TSSR
2030 010622 012702 000200    MOV    #SSR,R2              ;SET UP EXPECTED
2031 010626 020201    25$:  CMP    R2,R1            ;AKE THEY OK
2032 010630 001401                BEQ    40$                   ;BR, IF EQUAL = OK
2033 010632 000402                BR     60$                   ;TROUBLE EXIT
2034 010634 000261    40$:  SEC                    ;SET CARRY NO TROUBLE
2035 010636 000401                BR     70$                   ;EXIT
2036 010640 000241    60$:  CLC                    ;CARRY CLEAR = ERROR
2037 010642    70$:
2038 010642 010400                MOV    R4,R0                ;PASS PACKET ADDRESS
2039 010644 000207                RTS     PC                   ;RETURN
2040
2041
2042
2043    ;
2044    ;PACKET FOR SPACE COMMAND
2045    ;
2046    ;      .=<.10>&177770
2047
2048    ;
2049    ;COMMAND WORD
2050 010650 000000    80$:  .WORD
2051    ;NUMBER OF RECORDS TO BE SPACED OVER WORD
2052 010652 000000    90$:  .WORD
2053 010654 000000                .WORD
2054 010656 000000                .WORD
2055 010660 000000    SDELAY: .WORD    0                ;DELAY COUNTER
2056                .EVEN
2057                .SBTTL  WRCHR - WRITE CHARACTERISTICS COMMAND

```

WRTCHR - WRITE CHARACTERISTICS COMMAND

```

2059
2060 ;*
2061 ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
2062 ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
2063 ;
2064 ;INPUT:
2065 ; R4 ADDRESS OF PACKET FROM TEST
2066 ; R5 FIRST DEVICE UNIBUS ADDRESS
2067 ; REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
2068 ;
2069 ;OUTPUT:
2070 ; R0 TSSR CONTENTS
2071 ; CARRY SET - WRITE CHARACTERISTICS COMMAND OK
2072 ; CLR - WRITE CHARACTERISTICS FAILED
2073 ;
2074 ;IMPLICIT OUTPUT:
2075 ;
2076 ; MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
2077 ; SOFTWARE SWITCHES SET AS FOLLOWS:
2078 ; EXTFEA = EXTENDED FEATURES PRESENT
2079 ; BENBSW = BUFFER ENABLE SWITCH ON OR OFF
2080 ;
2081 ;SIDE EFFECTS:
2082 ;-
2083 WRTCHR::
2084 SAVREG ;SAVE THE GENERAL REGISTERS
2085 CLR BENBSW ;CLEAR BUFFER ENABLE SWITCH
2086 CLR EXTFEA ;CLEAR EXTENDED FEATURES SW SWITCH
2087 10$: MOV R4,TSDB(R5) ;SEND OUT COMMAND
2088 JSR PC,CHKTSSR ;WAIT FOR SSR
2089 BCS 20$ ;BR, IF SSR IS SET AND OK
2090 BR 60$ ;BR IF TROUBLE CARRY = CLEAR
2091 20$: MOV TSSR(R5),R1 ;READ TSSR
2092 MOV #SSR,R2 ;SET UP EXPECTED
2093 BIT #OFL,R1 ;WAS OFF LINE SET IN TSSR
2094 BEQ 25$ ;BR, IF NO OFL SET
2095 BIS #OFL,R2 ;MAKE THEM LOOK ALIKE
2096 25$: CMP R2,R1 ;ARE THEY OK
2097 BEQ 40$ ;BR, IF EQUAL = OK
2098 BR 60$ ;TROUBLE EXIT
2099 40$: ADD #8,R4 ;POINT TO WRT CHARA DATA PACKET
2100 MOV (R4),R3 ;GET ADDRESS OF MESSAGE BUFFER
2101 BIT #X2.EXTF,XST2(R3) ;EXTENDED FEATURES BIT SET?
2102 BEQ 45$ ;BR IF NO
2103 INC EXTFEA ;SET EXTENDED FEATURES SW SWITCH
2104 45$: BIT #X2.BUFE,XST2(R3) ;BUFFER ENABLE SWITCH SET
2105 BEQ 50$ ;BR, IF SWITCH NOT SET
2106 INC BENBSW ;SET SOFTWARE SWITCH FOR ENABLED
2107 50$: MOV XST2(R3),REV ;MICROCODE REV LEVEL
2108 BIC #17700,REV ;CLEAR UNWANTED BITS
2109 CMP #1,REV ;IS IT A NEW MICROCODE
2110 BEQ 55$ ;NO BR
2111 MOV #1,EXTFEA ;ALWAY EXTENDED FEATURE FOR NEW
2112 ;MICROCODE
2113 BIS #X2.EXTF,XST2(R3) ;EXTENDED FEATURE ALWAYS SET IN
2114 ;MICROCODE
2115

```


REWIND - POSITION TAPE (REWIND) COMMAND

```

2122                .SBTTL REWIND - POSITION TAPE (REWIND) COMMAND
2123                ;*
2124                ;
2125                ; THIS ROUTINE WILL REWIND THE SELECTED TAPE.
2126                ;
2127                ; CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
2128                ; TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
2129                ; SSR TO SET IN THE TSSR
2130                ;
2131                ;
2132                ; CALLING SEQUENCE:
2133                ;
2134                ; DO A SOFT INIT
2135                ; DO A WRITE CHARACTERISTICS
2136                ; JSR PC,REWIND
2137                ;
2138                ; INPUT:
2139                ;
2140                ; R5 FIRST DEVICE UNIBUS ADDRESS
2141                ;
2142                ;
2143                ; OUTPUT
2144                ;
2145                ; R0 THE CONTENTS OF R4 IS PASSED TO R0
2146                ;
2147                ;
2148                ;-
2149                REWIND::
2150                SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
2151                MOV #RWPACK,R4                        ;GET PACKET ADDRESS
2152                MOV R4,TSDB(R5)                       ;SEND PACKET ADDRESS TO EXECUTE
2153                MOV #360.,R3                          ;ENOUGH TIME FOR 2400' REEL TO REWIND
2154                JSR PC,WAITF                          ;WAIT FOR SSR TO SET
2155                BCS 20$                               ;LEAVE WHEN SSR IS SET
2156                DELAY 250.                            ;WAIT FOR .25 SECONDS
                MOV #250.,(PC)+
                .WORD 0
                MOV L#DLY,(PC)+
                .WORD 0
                DEC -6(PC)
                BNE -4
                DEC -22(PC)
                BNE -20
                DEC R3                                  ;BUMP COUNTER DOWN
                BNE 10$                                  ;KEEP GOING
                CLC                                     ;CLEAR CARRY TO SET ERROR
                MOV R4,R0                               ;PASS THE PACKET ADDRESS
                RTS PC                                  ;RETURN
                .=<..10>E177770
2164                RWPACK:
2166                .WORD 102010
2167                .WORD 0
2168                ;POSTION COMMAND (REWIND)
                ;NOT USED
2149 011054
2150 011054
2151 011060 012704 011150
2152 011064 010465 000000
2153 011070 012703 000550
2154 011074 004737 016310
2155 011100 103417
2156 011102
    011102 012727 000372
    011106 000000
    011110 013727 002116
    011114 000000
    011116 005367 177772
    011122 001375
    011124 005367 177756
    011130 001367
2157 011132 005303
2158 011134 001357
2159 011136 000241
2160 011140 010400
2161 011142 000207
2162
2164 011150
2166 011150
2167 011150 102010
2168 011152 000000

```

CKRAM - COMPARE RAM TO I/O PACKET

```

2170 .SBTTL CKRAM - COMPARE RAM TO I/O PACKET
2171 ;*
2172 ;
2173 ;ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
2174 ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
2175 ;
2176 ;INPUT:
2177 ;
2178 ; R4 ADDRESS OF THE COMMAND PACKET
2179 ; R5 FIRST DEVICE UNIBUS ADDRESS
2180 ;
2181 ;OUTPUT:
2182 ;
2183 ; CARRY SET - RAM MATCHES PACKET
2184 ; CLR - RAM DOES NOT MATCH PACKET
2185 ;
2186 ;IMPLICIT OUTPUT:
2187 ;
2188 ; THE TABLE RAMDATA IS FILLED WITH THE
2189 ; DATA HELD IN RAM.
2190 ; RAMSIZ IS SET TO 8. FOR PRAMPKT ROUTINE
2191 ;
2192 ;SIDE EFFECTS:
2193 ;
2194 ; THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
2195 ;
2196 ;-
2197 ;

```

```

2198 011154 CKRAM:: SAVREG ;SAVE THE GENERAL REGISTERS
2199 011154 MOV #RAMDATA,R1 ;ADDRESS TO SAVE THE RAM DATA
2200 011160 012701 002244 MOV #RMPKTBEGR2 ;BYTE ADDRESS OF FIRST RAM DATA
2201 011164 012702 000201 CLR R3 ;CLEAR THE ERROR FLAG
2202 011170 005003 JSR PC,CHKTSSR ;WAIT FOR SSR
2203 011172 004737 016376 000000 000000 MOVB #0,TSDB(R5) ;SET MAINTENANCE MODE
2204 011176 112765 000000 10$: JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
2205 011204 004737 016376 MOV R2,TSDB(R5) ;SELECT NEXT RAM ADDRESS
2206 011210 010265 000000 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
2207 011214 004737 016376 MOVVB TSBA(R5),(R1) ;READ THE RAM DATA
2208 011220 116511 000000 CMPB (R1)+,(R4)+ ;COMPARE TO EXPECTED
2209 011224 122124 BEQ 20$ ;BRANCH IF OK
2210 011226 001401 INC R3 ;SET ERROR FLAG
2211 011230 005203 INC R2 ;ADDRESS OF NEXT RAM LOCATION
2212 011232 005202 20$: INC R2 ;REACHED END YET ?
2213 011234 020227 000210 CMP R2,#RMPKTEND ;BRANCH TILL ALL READ
2214 011240 003761 BLE 10$ ;WAS AN ERROR FOUND ?
2215 011242 005703 TST R3 ;BRANCH IF NOT
2216 011244 001402 BEQ 30$ ;CLEAR CARRY TO SHOW ERROR
2217 011246 000241 CLC ;AND EXIT
2218 011250 000401 BR 50$ ;SHOW GOOD COMPARE
2219 011252 000261 SEC ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
2220 011254 012737 000010 002304 50$: MOV #8.,RAMSIZ
2221 011262 000207 RTS PC ;RETURN

```

CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA

```

2223          .SBTTL CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA
2224          ;*
2225          ;
2226          ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
2227          ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
2228          ;
2229          ;INPUT:
2230          ;
2231          ;       R4      ADDRESS OF THE CHARACTERISTICS DATA
2232          ;       R5      FIRST DEVICE UNIBUS ADDRESS
2233          ;
2234          ;OUTPUT:
2235          ;
2236          ;       CARRY   SET - RAM MATCHES PACKET
2237          ;              CLR - RAM DOES NOT MATCH PACKET
2238          ;
2239          ;IMPLICIT OUTPUT:
2240          ;
2241          ;       THE TABLE RAMDATA IS FILLED WITH THE
2242          ;       DATA HELD IN RAM.
2243          ;       RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKI ROUTINE
2244          ;
2245          ;SIDE EFFECTS:
2246          ;
2247          ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
2248          ;
2249          CKRAM2::
2250          SAVREG          ;SAVE THE GENERAL REGISTERS
2251          MOV             #RAMDATA,R1      ;ADDRESS TO SAVE THE RAM DATA
2252          MOV             #RMCHBEG,R2     ;BYTE ADDRESS OF FIRST RAM DATA
2253          CLR            R3              ;CLEAR THE ERROR FLAG
2254          JSR            PC,CHKTSSR      ;WAIT FOR SSR
2255          MOVB           #0,TSDB(R5)     ;SET MAINTENANCE MODE
2256          JSR            PC,CHKTSSR      ;WAIT FOR SSR TO SET
2257          MOV            R2,TSDB(R5)     ;SELECT NEXT RAM ADDRESS
2258          JSR            PC,CHKTSSR      ;WAIT FOR SSR TO SET
2259          MOVB           TSBA(R5),(R1)   ;READ THE RAM DATA
2260          CMPB           (R1)*,(R4)*    ;COMPARE TO EXPECTED
2261          BEQ            20$            ;BRANCH IF OK
2262          INC            R3              ;SET ERROR FLAG
2263          INC            R2              ;ADDRESS OF NEXT RAM LOCATION
2264          MOV            #8.,RAMSIZ      ;ASSUME EXTFEA NOT SET
2265          TST            EXTFEA         ;IS THE SOFTWARE EXTENDED FEATURES SET
2266          BEQ            25$            ;BR, IF NOT SET
2267          MOV            #10.,RAMSIZ    ;SET RAMSIZ FOR EXTEND FEATURES
2268          CMP            R2,#RMCHEND    ;AT END OF EXTENDED BUFFER
2269          BLE            10$            ;BR, IF NOT AT END YET
2270          BR            27$            ;AT END BRANCH
2271          CMP            R2,#RMCHEND-2  ;REACHED END YET ?
2272          BLE            10$            ;BRANCH TILL ALL READ
2273          TST            R3              ;WAS AN ERROR FOUND ?
2274          BEQ            30$            ;BRANCH IF NOT
2275          CLC              ;CLEAR CARRY TO SHOW ERROR
2276          BR            50$            ;AND EXIT
2277          SEC              ;SHOW GOOD COMPARE
2278          RTS             PC            ;RETURN

```

CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS

```

2280 .SBTTL CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS
2281 ;*
2282 ;
2283 ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
2284 ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
2285 ;ERROR PRINT ROUTINES.
2286 ;
2287 ;INPUT:
2288 ;
2289 ; R0 RECV MESSAGE BUFFER HIGH ORDER ADDRESS
2290 ; R1 RECV MESSAGE BUFFER LOW ORDER ADDRESS
2291 ; R2 EXPD MESSAGE BUFFER ADDRESS
2292 ;OUTPUT:
2293 ;
2294 ; CARRY SET - MESSAGE BUFFERS MATCH
2295 ; CLR -MESSAGE BUFFERS DON'T MATCH
2296 ;
2297 ;IMPLICIT OUTPUT:
2298 ;
2299 ; EXPMSG BUFFER IS SET TO EXPD DATA
2300 ; RECMMSG BUFFER IS SET TO RECV DATA
2301 ; RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
2302 ; RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
2303 ;
2304 ;-
2305 CKMSG::
2306 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2307 MOV R0,RCVHIADD ;SAVE RECV HIGH ADDRESS
2308 MOV R1,RCVLOADD ;SAVE RECV LOW ADDRESS
2309 TST KTENABLE ;TESTING ABOVE 28K?
2310 BEQ 10$ ;BR IF NO
2311 JSR PC,SETMAP ;RETURN ADDRESS BIASED TO PAR6 IN R0
2312 MOV R0,R1 ;GET RETURNED ADDRESS BIASED TO PAR6
2313 10$: CLR R4 ;WORD IN BUFFER
2314 CLR R3 ;CLEAR ERROR SEEN FLAG
2315 MOV R2,R5 ;GET EXPD BUFFER ADDRESS
2316 15$: MOV (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
2317 MOV (R1),RECMMSG(R4) ;SAVE RECV FOR ERROR REPORT
2318 CMP (R2)+,(R1)+ ;EXPD EQUAL RECV?
2319 BEQ 25$ ;BR IF YES
2320 INC R3 ;SET ERROR SEEN FLAG
2321 25$: ADD #2,R4 ;POINT TO NEXT WORD ADDRESS
2322 CMP R4,#14 ;DONE FIRST 7 WORDS?
2323 BLE 15$ ;BR IF NO
2324 BIT #X2.EXTF,XST2(R5) ;IS EXTENDED FEATURES SET IN EXPD?
2325 BEQ 50$ ;BR IF NO
2326 CMP R4,#16 ;DONE EXTENDED FEATURES WORD?
2327 BLE 15$ ;BR IF NO
2328 50$: TST R3 ;ANY ERRORS SEEN?
2329 BEQ 55$ ;BR IF NO
2330 CLC ;SET FAILURE
2331 BR 60$ ;
2332 55$: SEC ;SET SUCCESS
2333 60$: RTS PC ;RETURN

```

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

2335 .SBTTL CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS
2336
2337 ;*
2338 ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
2339 ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
2340 ;ERROR PRINT ROUTINES.
2341 ;
2342 ;INPUT:
2343 ; R0 RECV MESSAGE BUFFER HIGH ORDER ADDRESS
2344 ; R1 RECV MESSAGE BUFFER LOW ORDER ADDRESS
2345 ; R2 EXPD MESSAGE BUFFER ADDRESS
2346 ; R3 NUMBER OF BYTES TO COMPARE
2347 ;
2348 ;OUTPUT:
2349 ; CARRY SET - MESSAGE BUFFERS MATCH
2350 ; CLR - MESSAGE BUFFERS DON'T MATCH
2351 ;
2352 ;IMPLICIT OUTPUT:
2353 ; EXPMSG BUFFER IS SET TO EXPD DATA
2354 ; RECMG MSG BUFFER IS SET TO RECV DATA
2355 ; RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
2356 ; RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
2357 ;-
2358 CKMSG2::
2359 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2360 CMP R3,#RECMG MSG-EXPMSG ;@D IS COUNT ABOVE MAX ALLOWED?
2361 BLE 5# ;@D BR IF NO
2362 MOV #RECMG MSG-EXPMSG,R3 ;@D
2363 PRINTF #DEBUGMSG ;@D
2364 MOV #DEBUGMSG,-(SP)
2365 MOV #1,-(SP)
2366 MOV SP,R0
2367 TRAP C#PNTF
2368 ADD #4,SP
2369 5# : MOV R0,RCVHIADD ;SAVE RECV HIGH ADDRESS
2370 MOV R1,RCVLOADD ;SAVE RECV LOW ADDRESS
2371 TST KTENABLE ;TESTING ABOVE 28K?
2372 BEQ 10# ;BR IF NO
2373 JSR PC,SETMAP ;RETURN ADDRESS BIASED TO PAR6 IN R0
2374 MOV R0,R1 ;GET RETURNED ADDRESS BIASED TO PAR6
2375 10# : CLR R4 ;WORD IN BUFFER
2376 CLR R5 ;CLEAR ERROR SEEN FLAG
2377 15# : MOVB (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
2378 MOVB (R1),RECMG MSG(R4) ;SAVE RECV FOR ERROR REPORT
2379 CMPB (R2)*,(R1)* ;EXPD EQUAL RECV?
2380 BEQ 25# ;BR IF YES
2381 INC R5 ;SET ERROR SEEN FLAG
2382 25# : ADD #1,R4 ;POINT TO NEXT BYTE
2383 CMP R4,R3 ;DONE ALL BYTES?
2384 BGE 50# ;BR IF YES
2385 BR 15# ;DO NEXT BYTE
2386 50# : TST R5 ;ANY ERRORS SEEN?
2387 BEQ 55# ;BR IF NO
2388 CLC ;SET FAILURE
2389 BR 60# ;
2390 55# : SEC ;SET SUCCESS
2391 60# : RTS PC ;RETURN

```

J5

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

2387 011672      120      122      117 DEBUGMSG: .ASCIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-' ;@aD
2388 011762      045      116      045 FERCM:  .ASCII /%N%A ***/
2389 011773      040      040      124 ERCM:  .ASCIZ / TSSR ERROR CODE REC'D = /
2390 012026      056      056      056 SIMSG:  .ASCIZ /... AFTER DOING SOFT INIT/
2391 012061      124      105      123 TINERR: .ASCIZ /TEST: .../
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407 012074      004737 006026 BGNMSG SFMSG
      012074
2408 012074      004737 006026 SFMSG:: JSR PC,PRITSSR ;PRINT CONTENTS OF TSSR REGISTER
2409 012100      004737 017242 JSR PC,CKDROP ;DROP UNIT, IF ALLOWED
2410 012104
      012104
      012104 104423 ENDMSG
      L10003: TRAP C$MSG

2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423 012106      004737 006026 BGNMSG PKTSSR
      012106
2424 012106      004737 006026 PKTSSR:: JSR PC,PRITSSR ;PRINT THE CONTENTS OF TSSR REGISTER
2425 012112      012700 000004 MOV #4,RO ;NO. OF WORDS IN PACKET
2426 012116      004737 007372 JSR PC,PRIPKT ;PRINT THE CONTENTS OF COMMAND PACKET
2427 012122
      012122
      012122 104423 ENDMSG
      L10004: TRAP C$MSG

```

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

2429
2430      ;*
2431      ;PRINT ROUTINE TO PRINT THE CONTENTS OF
2432      ;TSSR AND A GET STATUS COMMAND PACKET.
2433      ;
2434      ;INPUTS:
2435      ;
2436      ;       R1       TSSR CONTENTS
2437      ;       R4       ADDRESS OF COMMAND PACKET
2438      ;
2438 012124      BGNMSG  PKTGETS
2439 012124      PKTGETS::
2439 012124 004737 006026      JSR       PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
2440 012130 012700 000002      MOV       #2,R0          ;NO. OF WORDS IN GET STATUS PACKET
2441 012134 004737 007372      JSR       PC,PRIPKT     ;PRINT THE CONTENTS OF COMMAND PACKET
2442 012140      ENDMSG
2443 012140      L10005:
2443 012140 104423      TRAP      C#MSG
2444
2445      ;*
2446      ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
2447      ;
2448      ;INPUTS:
2449      ;
2450      ;       R1       TSSR CONTENTS
2450      ;       R4       ADDRESS OF COMMAND PACKET
2451 012142      BGNMSG  SFFMSG
2451 012142 004737 006026      SFFMSG::
2452 012146      JSR       PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
2453 012146      ENDMSG
2453 012146 104423      L10006:
2454      .SBTTL  TRAP      C#MSG
2455      .SBTTL  .SBTTL  PKTMES - PRINT TSSR AND MESSAGE BUFFER
2456
2457      ;*
2458      ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
2459      ;BUFFER FOR ERROR REPORTS
2460      ;
2461      ;INPUTS:
2462      ;
2463      ;       R1       CONTENTS OF TSSR
2464      ;       R2       LOW ORDER MESSAGE BUFFER
2465      ;       R3       HIGH ORDER MESSAGE BUFFER ADDRESS
2466      ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
2467 012150      BGNMSG  PKTMES
2467 012150 004737 006026      PKTMES::
2468 012154 010200      JSR       PC,PRITSSR      ;PRINT CONTENTS OF TSSR
2469 012156 010301      MOV       R2,R0          ;LOW ORDER ADDRESS
2470 012160 004737 014302      MOV       R3,R1          ;HIGH ORDER ADDRESS
2470 012164      JSR       PC,PRMESS     ;PRINT THE MESSAGE BUFFER
2470 012164      ENDMSG
2470 012164 104423      L10007:
2470 012164      TRAP      C#MSG

```


ADDSSR - PRINT TEST ADDRESS AND TSSR

```

2472                                     .SBTTL ADDSSR - PRINT TEST ADDRESS AND TSSR
2473                                     ;*
2474                                     ;PRINT ROUTINE TO PRINT THE CONTENTS OF
2475                                     ;TSSR AND A MEMORY TEST ADDRESS
2476                                     ;
2477                                     ;INPUTS:
2478                                     ;
2479                                     ;       R5       FIRST DEVICE UNIBUS ADDRESS
2480                                     ;       ERRHI     HIGH ORDER MEMORY TEST ADDRESS
2481                                     ;       ERRLO     LOW ORDER MEMORY TEST ADDRESS
2482                                     ;
2483                                     ;-
2484 012166                               BGNMSG ADDSSR
2485 012166                               ADDSSR::
2486 012166 004737 010276                 JSR     PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
2487 012172 016501 000002                 MOV     TSSR(R5),R1    ;GET CURRENT TSSR
2488 012176 004737 006026                 JSR     PC,PRITSSR    ;PRINT THE CONTENTS OF TSSR REGISTER
2489 012202                               ENDMMSG
2490 012202                               L10010:
2491 012202 104423                         TRAP   C#MSG
2492                                     .SBTTL MSGEXP - PRINT WRITE CHAR. EXPD-RCV MESSAGE BUFFERS
2493                                     ;*
2494                                     ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
2495                                     ;
2496                                     ;IMPLICIT INPUTS:
2497                                     ;
2498                                     ;       EXPMSG   - EXPECTED MESSAGE BUFFER
2499                                     ;       RECMMSG  - RECEIVED MESSAGE BUFFER
2500                                     ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2501                                     ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2502                                     ;-
2503 012204                               BGNMSG MSGEXP
2504 012204                               MSGEXP::
2505 012204 012700 000007                 MOV     #7,R0         ;ASSUME NO EXT FEATURES
2506 012210 005737 002226                 TST     EXTFEA        ;EXT FEATURES SET?
2507 012214 001402                         BEQ     5$            ;BR IF NO
2508 012216 012700 000010                 MOV     #8.,R0        ;EXT FEATURE BUFFER IS 8 WORDS
2509 012222 004737 014612                 JSR     PC,PRMSGEXP   ;PRINT EXPD/RCV MESSAGE BUFFERS
2510 012226                               ENDMMSG
2511 012226                               L10011:
2512 012226 104423                         TRAP   C#MSG

```

FIFEXP - PRINT FIFO EXP/RECV DATA

```

2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522 012230
      012230
2523 012230
      012230 010146
      012232 012746 012302
      012236 012746 000002
      012242 010600
      012244 104415
      012246 062706 000006
2524 012252
      012252 012746 012351
      012256 012746 000001
      012262 010600
      012264 104415
      012266 062706 000004
2525 012272 010100
2526 012274 004737 015162
2527 012300
      012300
      012300 104423
2528 012302 045 116
2529 012351 045 116
2530

```

```

.SBTTL FIFEXP - PRINT FIFO EXP/REC DATA
;*
;PRINT ROUTINE TO PRINT FIFO EXP/REC DATA
;
; R1 - BYTE COUNT
;IMPLICIT INPUTS:
;
; EXPMSG - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
; RECVMSG - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
;-
BGNMSG FIFEXP
FIFEXP:
PRINTX #FIF1MSG,R1 ;PRINT BYTES TRANSFERRED
MOV R1,-(SP)
MOV #FIF1MSG,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #6,SP
PRINTX #FIF2MSG ;PRINT HEADER MSG
MOV #FIF2MSG,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #4,SP
MOV R1,R0 ;GET BYTE COUNT
JSR PC,PRBYTEXP ;PRINT FIFO BYTES IN ERROR
ENDMSG
L10012:
TRAP C#MSG
.FIF1MSG: .ASCIZ '#N#A NUMBER OF BYTES TRANSFERRED = #D2'
.FIF2MSG: .ASCIZ '#N#A FIFO DATA BYTES IN ERROR:'
.EVEN

```

MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS

```

2532          .SBTTL MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
2533          ;*
2534          ;
2535          ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
2536          ;
2537          ;
2538          ;IMPLICIT INPUTS:
2539          ;
2540          ;     EXPMSG - EXPECTED MESSAGE BUFFER
2541          ;     RECMSG - RECEIVED MESSAGE BUFFER
2542          ;     RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2543          ;     RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2544          ;
2545          ;-
2546          BGNMSG MSGSTAT
2547          MSGSTAT:
2548          10$: MOV     #STATCOD,R1      ;ASCII ADDRESS TABLE
2549          MOV     (R1)+,R0          ;DONE ALL MSG LINES?
2550          BEQ     20$,              ;BR IF YES
2551          PRINTX R0                 ;PRINT STATUS BIT NAMES
2552          MOV     R0,-(SP)
2553          MOV     #1,-(SP)
2554          MOV     SP,R0
2555          TRAP   C#PNTX
2556          ADD     #4,SP
2557          BR     10$                ;DO ANOTHER MSG LINE
2558          20$: MOV     #10,,R0      ;NUMBER OF WORDS IN A READ STATUS BUFFER
2559          JSR     PC,PRMSGEXP        ;PRINT EXPD/RECV MESSAGE BUFFERS
2560          ENDMSG
2561          L10013: TRAP   C#MSG
2562          ;
2563          ;
2564          ;
2565          ;
2566          ;
2567          ;
2568          ;
2569          ;
2570          ;
2571          ;
2572          ;
2573          ;
2574          ;
2575          ;
2576          ;
2577          ;
2578          ;
2579          ;
2580          ;
2581          ;
2582          ;
2583          ;
2584          ;
2585          ;
2586          ;
2587          ;
2588          ;
2589          ;
2590          ;
2591          ;
2592          ;
2593          ;
2594          ;
2595          ;
2596          ;
2597          ;
2598          ;
2599          ;
2600          ;
2601          ;
2602          ;
2603          ;
2604          ;
2605          ;
2606          ;
2607          ;
2608          ;
2609          ;
2610          ;
2611          ;
2612          ;
2613          ;
2614          ;
2615          ;
2616          ;
2617          ;
2618          ;
2619          ;
2620          ;
2621          ;
2622          ;
2623          ;
2624          ;
2625          ;
2626          ;
2627          ;
2628          ;
2629          ;
2630          ;
2631          ;
2632          ;
2633          ;
2634          ;
2635          ;
2636          ;
2637          ;
2638          ;
2639          ;
2640          ;
2641          ;
2642          ;
2643          ;
2644          ;
2645          ;
2646          ;
2647          ;
2648          ;
2649          ;
2650          ;
2651          ;
2652          ;
2653          ;
2654          ;
2655          ;
2656          ;
2657          ;
2658          ;
2659          ;
2660          ;
2661          ;
2662          ;
2663          ;
2664          ;
2665          ;
2666          ;
2667          ;
2668          ;
2669          ;
2670          ;
2671          ;
2672          ;
2673          ;
2674          ;
2675          ;
2676          ;
2677          ;
2678          ;
2679          ;
2680          ;
2681          ;
2682          ;
2683          ;
2684          ;
2685          ;
2686          ;
2687          ;
2688          ;
2689          ;
2690          ;
2691          ;
2692          ;
2693          ;
2694          ;
2695          ;
2696          ;
2697          ;
2698          ;
2699          ;
2700          ;
2701          ;
2702          ;
2703          ;
2704          ;
2705          ;
2706          ;
2707          ;
2708          ;
2709          ;
2710          ;
2711          ;
2712          ;
2713          ;
2714          ;
2715          ;
2716          ;
2717          ;
2718          ;
2719          ;
2720          ;
2721          ;
2722          ;
2723          ;
2724          ;
2725          ;
2726          ;
2727          ;
2728          ;
2729          ;
2730          ;
2731          ;
2732          ;
2733          ;
2734          ;
2735          ;
2736          ;
2737          ;
2738          ;
2739          ;
2740          ;
2741          ;
2742          ;
2743          ;
2744          ;
2745          ;
2746          ;
2747          ;
2748          ;
2749          ;
2750          ;
2751          ;
2752          ;
2753          ;
2754          ;
2755          ;
2756          ;
2757          ;
2758          ;
2759          ;
2760          ;
2761          ;
2762          ;
2763          ;
2764          ;
2765          ;
2766          ;
2767          ;
2768          ;
2769          ;
2770          ;
2771          ;
2772          ;
2773          ;
2774          ;
2775          ;
2776          ;
2777          ;
2778          ;
2779          ;
2780          ;
2781          ;
2782          ;
2783          ;
2784          ;
2785          ;
2786          ;
2787          ;
2788          ;
2789          ;
2790          ;
2791          ;
2792          ;
2793          ;
2794          ;
2795          ;
2796          ;
2797          ;
2798          ;
2799          ;
2800          ;
2801          ;
2802          ;
2803          ;
2804          ;
2805          ;
2806          ;
2807          ;
2808          ;
2809          ;
2810          ;
2811          ;
2812          ;
2813          ;
2814          ;
2815          ;
2816          ;
2817          ;
2818          ;
2819          ;
2820          ;
2821          ;
2822          ;
2823          ;
2824          ;
2825          ;
2826          ;
2827          ;
2828          ;
2829          ;
2830          ;
2831          ;
2832          ;
2833          ;
2834          ;
2835          ;
2836          ;
2837          ;
2838          ;
2839          ;
2840          ;
2841          ;
2842          ;
2843          ;
2844          ;
2845          ;
2846          ;
2847          ;
2848          ;
2849          ;
2850          ;
2851          ;
2852          ;
2853          ;
2854          ;
2855          ;
2856          ;
2857          ;
2858          ;
2859          ;
2860          ;
2861          ;
2862          ;
2863          ;
2864          ;
2865          ;
2866          ;
2867          ;
2868          ;
2869          ;
2870          ;
2871          ;
2872          ;
2873          ;
2874          ;
2875          ;
2876          ;
2877          ;
2878          ;
2879          ;
2880          ;
2881          ;
2882          ;
2883          ;
2884          ;
2885          ;
2886          ;
2887          ;
2888          ;
2889          ;
2890          ;
2891          ;
2892          ;
2893          ;
2894          ;
2895          ;
2896          ;
2897          ;
2898          ;
2899          ;
2900          ;
2901          ;
2902          ;
2903          ;
2904          ;
2905          ;
2906          ;
2907          ;
2908          ;
2909          ;
2910          ;
2911          ;
2912          ;
2913          ;
2914          ;
2915          ;
2916          ;
2917          ;
2918          ;
2919          ;
2920          ;
2921          ;
2922          ;
2923          ;
2924          ;
2925          ;
2926          ;
2927          ;
2928          ;
2929          ;
2930          ;
2931          ;
2932          ;
2933          ;
2934          ;
2935          ;
2936          ;
2937          ;
2938          ;
2939          ;
2940          ;
2941          ;
2942          ;
2943          ;
2944          ;
2945          ;
2946          ;
2947          ;
2948          ;
2949          ;
2950          ;
2951          ;
2952          ;
2953          ;
2954          ;
2955          ;
2956          ;
2957          ;
2958          ;
2959          ;
2960          ;
2961          ;
2962          ;
2963          ;
2964          ;
2965          ;
2966          ;
2967          ;
2968          ;
2969          ;
2970          ;
2971          ;
2972          ;
2973          ;
2974          ;
2975          ;
2976          ;
2977          ;
2978          ;
2979          ;
2980          ;
2981          ;
2982          ;
2983          ;
2984          ;
2985          ;
2986          ;
2987          ;
2988          ;
2989          ;
2990          ;
2991          ;
2992          ;
2993          ;
2994          ;
2995          ;
2996          ;
2997          ;
2998          ;
2999          ;
3000          ;

```

MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS

```

2565                                     .SBTTL MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
2566                                     ;*
2567                                     ;
2568                                     ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
2569                                     ;
2570                                     ;IMPLICIT INPUTS:
2571                                     ;
2572                                     ;     EXPMSG - EXPECTED MESSAGE BUFFER
2573                                     ;     RECMSG - RECEIVED MESSAGE BUFFER
2574                                     ;     RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2575                                     ;     RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2576                                     ;
2577 013124                                BGNMSG MSGLOOP
013124                                MSGLOOP::
2578 013124 012701 013166                10$: MOV     #LOOPCOD,R1      ;ASCII ADDRESS TABLE
2579 013130 012100                        MOV     (R1)+,R0      ;DONE ALL MSG LINES?
2580 013132 001410                        BEQ     20$,        ;BR IF YES
2581 013134                                PRINTX  R0           ;PRINT STATUS BIT NAMES
013134                                MOV     R0,-(SP)
013136 012746 000001                    MOV     #1,-(SP)
013142 010600                            MOV     SP,R0
013144 104415                            TRAP   C#PNTX
013146 062706 000004                    ADD     #4,SP
2582 013152 000766                        BR      10$         ;DO ANOTHER MSG LINE
2583 013154 012700 000012                20$: MOV     #10,R0   ;NUMBER OF WORDS IN A READ STATUS BUFFER
2584 013160 004737 014612                JSR     PC,PRMSGEXP ;PRINT EXPD/RCV MESSAGE BUFFERS
2585 013164                                ENDMSG
013164                                L10014:
013164 104423                            TRAP   C#MSG
2585
2587 013166 013206 013261 013360        LOOPCOD: .WORD 1$,2$,3$,4$,5$,6$,7$,0
2588 013206 045 116 045 1$: .ASCIZ '#N#A Tape Bus Loopback Signals in Word #8:'
2589 013261 045 116 045 2$: .ASCIZ '#N#A PARERR<15> IRESV2<14> IRESV1<13>'
2590 013360 045 116 045 3$: .ASCIZ '#N#A IHISP=>IEOT<12> IWRT=>IIDENT<11> IREV =>ICER <10>'
2591 013457 045 116 045 4$: .ASCIZ '#N#A IWFM =>IFMK<09> IEDIT=>IHER <08> IFAD =>ISPEED<07>'
2592 013556 045 116 045 5$: .ASCIZ '#N#A ITADO=>IRDY<06> ITAD1=>IONL <05> IERASE=>ILDP <04>'
2593 013655 045 116 045 6$: .ASCIZ '#N#A IREW =>IDBY<03> IRWU =>IRWD <02> IFEN =>IFBY <01>'
2594 013754 045 116 045 7$: .ASCIZ '#N#A IGO =>IFPT<00>'
2595                                     .EVEN

```

C6

MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER

```

2597          .SBTTL MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
2598          ;*
2599          ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
2600          ;
2601          ;
2602          ;IMPLICIT INPUTS:
2603          ;
2604          ;
2605          ;   EXPMSG - EXPECTED MESSAGE BUFFER
2606          ;   RECMSG - RECEIVED MESSAGE BUFFER
2607          ;   RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2608          ;   RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2609          ;-
2610          BGNMSG MSGSUB
2611          MSGSUB::
2612          MOV     #10,R0          ;SIZE OF WRITE SUBSYSTEM BUFFER
2613          JSR     PC,PRMSGEXP    ;PRINT EXPD/RCV MESSAGE BUFFERS
2614          ENDMSG
2615          L10015:
2616          TRAP    C#MSG
2617          .SBTTL MEMADD - PRINT MEMORY ADDRESS DATA ERROR
2618          ;*
2619          ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
2620          ;
2621          ;IMPLICIT INPUTS:
2622          ;
2623          ;   ERRHI - MEMORY ERROR HIGH ORDER ADDRESS
2624          ;   ERRLO - MEMORY ERROR LOW ORDER ADDRESS
2625          ;   EXP   - EXPECTED DATA
2626          ;   RECV  - RECEIVED DATA
2627          ;-
2628          BGNMSG MEMADD
2629          MEMADD::
2630          JSR     PC,PRIADD      ;PRINT MEMORY ADDRESS IN ERROR
2631          MOV     EXPD,R1        ;GET EXPD DATA
2632          MOV     RECV,R2        ;GET RECEIVED DATA
2633          JSR     PC,PRIXOR      ;PRINT EXPD/RCV
2634          ENDMSG
2635          L10016:
2636          TRAP    C#MSG

```

```

2610 014002
2611 014002 012700 000012
2612 014006 004737 014612
2613 014012
2614 014012 104423
2627 014014
2628 014014 004737 010162
2629 014020 013701 002234
2630 014024 013702 002236
2631 014030 004737 007744
2632 014034
2633 014034 104423

```

D6

PRAMPKT - PRINT RAM AND PACKET DATA

```

2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650 014036
2651 014036
2652 014042 012701 002244
2653 014046 005002
2654 014050 122124
2655 014052 001005
2656 014054
2657 014064 000436
2658 014066 116105 177777
2659 014072 116403 177777
2660 014076
2661 014106 042703 177400
2662 014112 116137 177777 002236
2663 014120 116437 177777 002234
2664 014126
    014126 010346
    014130 013746 002234
    014134 013746 002236
    014140 010246
    014142 012746 014216
    014146 012746 000005
    014152 010600
    014154 104414
    014156 062706 000014
2665 014162 005202
2666 014164 005737 002304
2667 014170 001404
2668 014172 020237 002304
2669 014176 003724
2670 014200 000403
2671 014202 020227 000010
2672 014206 002720
2673 014210 005037 002304
2674 014214 000207
2675
2676 014216 045 116 045 RAMASC: .ASCIZ 'N#A BYTE: #D2#A RAM: #03#A Packet: #03#A XOR:#03'
2677

```

```

.SBTTL PRAMPKT - PRINT RAM AND PACKET DATA
;*
;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
;WHEN THE RAM DATA DOES NOT MATCH.
;
;INPUTS:
;
;      R4      POINTER TO COMMAND PACKET
;IMPLICIT INPUTS:
;      RAMDATA  DATA AS READ FROM THE RAM
;      RAMSIZ   NUMBER OF BYTES IN PACKET
;               IF RAMSIZ=0 THEN DEFAULT TO 8.
;
;IMPLICIT OUTPUTS:
;      RAMSIZ   SET TO 0
;-
PRAMPKT:
    SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
    MOV     #RAMDATA,R1                   ;DATA FROM THE RAM
    CLR     R2                             ;INIT BYTE NUMBER
    5$:    CMPB  (R1),.(R4)                 ;COMPARE EXPECTED, RECEIVED
    BNE     7$                             ;BR IF NO MATCH
    FORCERROR 7$,NOTSSR
    BR     10$
    7$:    MOVB -1(R1),R5                   ;GET RECV RAM DATA
    MOVB -1(R4),R3                         ;GET EXPD PACKET DATA
    XOR     R5,R3                          ;XOR EXPD/RECV
    BIC     #177400,R3                     ;LOW BYTE ONLY
    MOVB -1(R1),RECV                       ;GET RECEIVED RAM DATA
    MOVB -1(R4),EXPD                       ;GET EXPECTED RAM DATA
    PRINTB #RAMASC,R2,RECV,EXPD,R3
    MOV     R3,-(SP)
    MOV     EXPD,-(SP)
    MOV     RECV,-(SP)
    MOV     R2,-(SP)
    MOV     #RAMASC,-(SP)
    MOV     #5,-(SP)
    MOV     SP,R0
    TRAP   C#PNTB
    ADD    #14,SP
    10$:   INC     R2                       ;UPDATE BYTE COUNT
    TST    RAMSIZ                          ;DEFAULT TO 8.?
    BEQ    15$                              ;BR IF YES
    CMP    R2,RAMSIZ                       ;DONE ALL BYTES?
    BLE    5$                              ;BR IF NO
    ;
    15$:   CMP    R2,#8.                   ;DONE DEFAULT NUMBER OF BYTES?
    BLT    5$                              ;BR IF NO
    20$:   BLT    5$
    25$:   CLR    RAMSIZ                   ;SET DEFAULT RAMSIZ
    RTS    PC                              ;RETURN

```

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```

2679          .SBTTL PRMESS - PRINT CONTENTS OF MESSAGE BUFFER
2680          ;*
2681          ;THIS ROUTINE PRINTS THE CONTENTS OF
2682          ;THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE TSV-05.
2683          ;
2684          ;INPUT:
2685          ;      R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
2686          ;      R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
2687          ;      NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
2688          ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
2689          ;-
2690          PRMESS: SAVREG          ;SAVE THE REGISTERS
2691          MOV      R0,R5          ;SAVE LOW ORDER ADDRESS
2692          TST      KTENABLE       ;ADDRESS ABOVE 28K?
2693          BNE     10$            ;BR IF YES
2694          CLR     R1              ;SET HIGH ORDER ADDRESS TO 0
2695          10$:  MOV     R1,R3      ;SAVE HIGH ORDER ADDRESS
2696          ROL     R0              ;SHIFT BIT15 TO C BIT
2697          ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
2698          PRINTX  @PROASC,R1,R5  ;PRINT MESSAGE BUFFER ADDRESS
2699          MOV     R5,-(SP)
2700          MOV     R1,-(SP)
2701          MOV     @PROASC,-(SP)
2702          MOV     #3,-(SP)
2703          MOV     SP,R0
2704          TRAP   C$PNTX
2705          ADD     #10,SP
2706          PRINTX  @PRIASC          ;PRINT HEADER FOR CONTENTS
2707          MOV     @PRIASC,-(SP)
2708          MOV     #1,-(SP)
2709          MOV     SP,R0
2710          TRAP   C$PNTX
2711          ADD     #4,SP
2712          CLR     R4              ;NUMBER OF THE NEXT WORD
2713          MOV     R5,R1          ;COPY LOW ORDER ADDRESS
2714          MOV     R3,R0          ;COPY HIGH ORDER ADDRESS
2715          BEQ    20$            ;BR IF NOT ABOVE 28K
2716          JSR    PC,SETMAP      ;SETUP PAR ADDRESS IN R0
2717          MOV     R0,R5          ;GET PAR FORMAT ADDRESS ABOVE 28K
2718          20$:  PRINTX  @PRASC,R4,(R5)+ ;PRINT THE CONTENTS OF MEMORY BUFFER
2719          MOV     (R5),-(SP)
2720          MOV     R4,-(SP)
2721          MOV     @PRASC,-(SP)
2722          MOV     #3,-(SP)
2723          MOV     SP,R0
2724          TRAP   C$PNTX
2725          ADD     #10,SP
2726          INC     R4              ;NUMBER OF THE NEXT
2727          CMP     R4,#7          ;DONE ALL YET ?
2728          BGT    50$            ;BRANCH IF ALL DONE
2729          BLT    20$            ;PRINT FIRST 7 WORDS
2730          BIT     @X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
2731          BNE    20$            ;PRINT EXTENDED STATUS WORD
2732          50$:  RTS     PC          ;RETURN
2733          PROASC: .ASCIZ  '%N%A Message Buffer Address = %01%05'
2734          PRIASC: .ASCIZ  '%N%A Message Buffer Contents:'
2735          PRASC:  .ASCIZ  '%N%A Word%D1%A: %0'

```

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```

2718 .EVEN
2719 .SBTTL PRMSGEXP - PRINT EXPD/RCV MESSAGE BUFFERS
2720
2721 ;*
2722 ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
2723 ; RO - NUMBER OF WORDS IN BUFFER
2724 ;IMPLICIT INPUTS:
2725 ; EXPMSG - EXPECTED MESSAGE BUFFER
2726 ; RECMMSG - RECEIVED MESSAGE BUFFER
2727 ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
2728 ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
2729
2729 014612 PRMSGEXP::
2730 014612 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2731 014616 010005 MOV RO,R5 ;SAVE NUMBER OF WORDS
2732 014620 013700 002310 MOV RCVLOADD,RO ;GET RECV LOW ADDRESS
2733 014624 010004 MOV RO,R4 ;COPY LOW ADDRESS
2734 014626 013701 002306 MOV RCVHIADD,R1 ;GET RECV HIGH ADDRESS
2735 014632 006100 ROL RO ;SHIFT BIT15 TO C BIT
2736 014634 006101 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
2737 014636 PRINTX #PRMSG0,R1,R4 ;PRINT MESSAGE BUFFER ADDRESS
2738 014636 010446 MOV R4,-(SP)
2739 014640 010146 MOV R1,-(SP)
2740 014642 012746 014772 MOV #PRMSG0,-(SP)
2741 014646 012746 000003 MOV #3,-(SP)
2742 014652 010600 MOV SP,RO
2743 014654 104415 TRAP C#PNTX
2744 014656 062706 000010 ADD #10,SP
2745 014662 PRINTX #PRMSG1 ;PRINT HEADER FOR CONTENTS
2746 014662 012746 015037 MOV #PRMSG1,-(SP)
2747 014666 012746 000001 MOV #1,-(SP)
2748 014672 010600 MOV SP,RO
2749 014674 104415 TRAP C#PNTX
2750 014676 062706 000004 ADD #4,SP
2751 014702 005004 CLR R4 ;NUMBER OF THE CURRENT WORD
2752 014704 012701 002324 MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
2753 014710 012702 002470 MOV #RCMMSG,R2 ;GET RECV BUFFER ADDRESS
2754 014714 011100 20$: MOV (R1),RO ;GET EXPD
2755 014716 011203 MOV (R2),R3 ;GET RECV
2756 014720 XOR RO,R3 ;XOR EXPD/RCV
2757 014730 PRINTX #PRMSG2,R4,(R1)*,(R2)*,R3
2758 014730 010346 MOV R3,-(SP)
2759 014732 012246 MOV (R2)*,-(SP)
2760 014734 012146 MOV (R1)*,-(SP)
2761 014736 010446 MOV R4,-(SP)
2762 014740 012746 015075 MOV #PRMSG2,-(SP)
2763 014744 012746 000005 MOV #5,-(SP)
2764 014750 010600 MOV SP,RO
2765 014752 104415 TRAP C#PNTX
2766 014754 062706 000014 ADD #14,SP
2767 014760 005204 INC R4 ;NUMBER OF THE NEXT
2768 014762 020405 CMP R4,R5 ;DONE ALL YET?
2769 014764 002001 BGE 50$ ;BR IF YES
2770 014766 000752 BR 20$ ;DO ANOTHER
2771 014770 000207 RTS PC ;RETURN
2772 014772 045 116 045 PRMSG0: .ASCIZ '#N#A Message Buffer Address = #01#05'
2773 015037 045 116 045 PRMSG1: .ASCIZ '#N#A Message Buffer Contents:'
2774 015075 045 116 045 PRMSG2: .ASCIZ '#N#A WORD #D2#A EXPD: #06#A RECV: #06#A XOR: #06'

```


PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS

```

2755 .EVEN
2756 .SBTTL PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
2757 ;*
2758 ;
2759 ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
2760 ; ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
2761 ;
2762 ; R0 - NUMBER OF BYTES IN BUFFER
2763 ;
2764 ;IMPLICIT INPUTS:
2765 ;
2766 ; EXPMSG - EXPECTED MESSAGE BUFFER
2767 ; RECMMSG - RECEIVED MESSAGE BUFFER
2768 ;-
2769 015162 PRBYTEXP::
2770 015162 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2771 015166 010005 MOV R0,R5 ;SAVE NUMBER OF BYTES
2772 015170 005037 002322 CLR PRMNO ;INIT ERROR COUNT
2773 015174 005004 CLR R4 ;NUMBER OF THE CURRENT BYTE
2774 015176 012701 002324 MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
2775 015202 012702 002470 MOV #RECMMSG,R2 ;GET RECV BUFFER ADDRESS
2776 015206 111100 20$: MOVB (R1),R0 ;GET EXPD BYTE
2777 015210 042700 177400 BIC #1C<377>,R0 ;CLEAR UPPER BYTE
2778 015214 110037 015530 MOVB R0,PRBEXP ;SAVE FOR ERROR REPORT
2779 015220 111203 MOVB (R2),R3 ;GET RECV BYTE
2780 015222 042703 177400 BIC #1C<377>,R3 ;CLEAR UPPER BYTE
2781 015226 110337 015532 MOVB R3,PRBREC ;FOR ERROR REPORT
2782 015232 XOR R0,R3 ;XOR EXPD/RECV
2783 015242 122122 CMPB (R1)+,(R2)+ ;EXPD = RECV?
2784 015244 001431 BEQ 30$ ;BR IF YES
2785 015246 005237 002322 INC PRMNO ;UPDATE ERROR COUNT
2786 015252 023727 002322 000010 CMP PRMNO,#8. ;PRINTED 8?
2787 015260 101023 BHI 30$ ;BR IF YES
2788 015262 27$: PRINTX #PRBMSG,R4,PRBEXP,PRBREC,R3
015262 010346 MOV R3,-(SP)
015264 013746 015532 MOV PRBREC,-(SP)
015270 013746 015530 MOV PRBEXP,-(SP)
015274 010446 MOV R4,-(SP)
015276 012746 015376 MOV #PRBMSG,-(SP)
015302 012746 000005 MOV #5,-(SP)
015306 010600 MOV SP,R0
015310 104415 TRAP C#PNTX
015312 062706 000014 ADD #14,SP
2789 015316 FORCEXIT 50$ ;@@D
2790 015326 000404 BR 35$ ;@D
2791 015330 30$: FORCERROR 27$,NOTSSR ;@D
2792 015330 ;@D
2793 015340 35$: INC R4 ;NUMBER OF THE NEXT
2794 015340 005204 CMP R4,R5 ;DONE ALL YET?
2795 015342 020405 BGE 50$ ;BR IF YES
2796 015344 002001 BR 20$ ;DO ANOTHER
2797 015346 000717 50$: PRINTX #PRBTOT,PRMNO ;PRINT TOTAL ERROR COUNT
2798 015350 MOV PRMNO,-(SP)
015350 013746 002322 MOV #PRBTOT,-(SP)
015354 012746 015463 MOV #2,-(SP)
015360 012746 000002 MOV SP,R0
015364 010600

```

H6

PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

```

015366 104415 TRAP C$PNTX
015370 062706 000006 ADD #6,SP
2799 015374 000207 RTS PC ;RETURN
2800
2801 015376 045 116 045 PRBMSG: .ASCIZ 'N%A BYTE #D2%A EXPD: #03%A RECV: #03%A XOR: #03'
2802 015463 045 116 045 PRBTOT: .ASCIZ 'N%A NUMBER OF BYTES IN ERROR = #D2'
2803 .EVEN
2804 015530 000000 PRBEXP: .WORD 0 ;EXPD
2805 015532 000000 PRBREC: .WORD 0 ;RECV
2806 .SBTTL EXPREC - PRINT EXPD/RECV WORD DATA
2807 ;*
2808 ;
2809 ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2810 ;
2811 ;INPUTS:
2812 ;
2813 ; R1 RECEIVED DATA
2814 ; R2 EXPECTED DATA
2815 ;
2816 ;-
2817
2818 015534 BGNMSG EXPREC
015534
2819 015534 004737 007744 EXPREC: JSR PC,PRIXOR ;PRINT THE DATA
2820 015540 ENDMSG
015540
L10017: TRAP C$MSG
015540 104423 .SBTTL EXPBREC - PRINT EXPD/RECV BYTE DATA
2821 ;*
2822 ;
2823 ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
2824 ;
2825 ;
2826 ;INPUTS:
2827 ;
2828 ; R1 RECEIVED DATA BYTE
2829 ; R2 EXPECTED DATA BYTE
2830 ;
2831 ;-
2832
2833
2834 015542 BGNMSG EXPBREC
015542
2835 015542 004737 007614 EXPBREC: JSR PC,PRIBXOR ;PRINT THE DATA
2836 015546 ENDMSG
015546
L10020: TRAP C$MSG
2837 015546 104423 .SBTTL RAMERR - PRINT RAM AND PACKET DATA
2838 ;*
2839 ;
2840 ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
2841 ;
2842 ;INPUTS:
2843 ;
2844 ; R4 POINTER TO COMMAND PACKET
2845 ;
2846 ;
2847 ;

```

RAMERR - PRINT RAM AND PACKET DATA

```

2848 ;IMPLICIT INPUTS:
2849 :
2850 :   RAMDATA      DATA AS READ FROM THE RAM
2851 :   RAMSIZ      NUMBER OF BYTES IN PACKET
2852 :               IF RAMSIZ=0 THEN DEFAULT TO 8.
2853 :
2854 ;IMPLICIT OUTPUTS:
2855 :
2856 :   RAMSIZ SET TO 0
2857 :-
2858 :
2859 015550 BGNMSG RAMERR
      015550
2860 015550 004737 014036 RAMERR:: JSR PC,PRAMPKT ;PRINT RAM/PACKET DATA
2861 015554 ENDMSG
      015554
      015554 104423 L10021: TRAP C$MSG
2862 :
2863 :   .SBTTL RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
2864 :
2865 ;*
2866 ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
2867 :
2868 ;INPUTS:
2869 :
2870 :   R4          POINTER TO COMMAND PACKET
2871 :
2872 ;IMPLICIT INPUTS:
2873 :
2874 :   RAMDATA      DATA AS READ FROM THE RAM
2875 :   RAMSIZ      NUMBER OF BYTES IN PACKET
2876 :               IF RAMSIZ=0 THEN DEFAULT TO 8.
2877 :   ERRHI       HIGH ORDER TEST ADDRESS
2878 :   ERRLO       LOW ORDER TEST ADDRESS
2879 :
2880 ;IMPLICIT OUTPUTS:
2881 :
2882 :   RAMSIZ SET TO 0
2883 :-
2884 :
2885 015556 BGNMSG RAMTADD
      015556
2886 015556 004737 010276 RAMTADD:: JSR PC,PRITADD ;PRINT TEST ADDRESS
2887 015562 004737 014036 JSR PC,PRAMPKT ;PRINT RAM/PACKET DATA
2888 015566 ENDMSG
      015566
      015566 104423 L10022: TRAP C$MSG
2889 :
2890 :   .SBTTL RAMEXP - PRINT RAM EXPD/RECV DATA
2891 :
2892 ;*
2893 ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2894 :
2895 ;INPUTS:
2896 :
2897 :   R1          RECEIVED DATA
2898 :   R2          EXPECTED DATA

```

RAMEXP - PRINT RAM EXPD/RECV DATA

```

2899          :      R4      CONTROLLER RAM ADDRESS
2900          :-
2901
2902 015570          BGNMSG  RAMEXP
          015570
2903 015570 042701 177400  RAMEXP::
2904 015574 042702 177400      BIC      #1C<377>,R1      ;SAVE EXPD RAM DATA BYTE
2905 015600 004737 010070      BIC      #1C<377>,R2      ;SAVE EXPD RAM DATA BYTE
2906 015604 004737 007744      JSR      PC,PRIRAM      ;PRINT THE RAM ADDRESS
2907 015610          JSR      PC,PRIXOR      ;PRINT THE DATA
          015610
          015610 104423  L10023:
2908          TRAP      C#MSG
2909          .SBTTL  TIMEXP - PRINT TIMER A,B AND EXP/REC
2910          ;*
2911          ;
2912          ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
2913          ;AND TIMER A,B HEADER MESSAGE
2914          ;
2915          ;INPUTS:
2916          ;
2917          ;      R1      RECEIVED DATA
2918          ;      R2      EXPECTED DATA
2919          ;
2920          :-
2921          BGNMSG  TIMEXP
          015612
          015612
2922 015612          TIMEXP::
          015612 012746 015640      PRINTX  #TIMSGO          ;PRINT HEADER
          015616 012746 000001      MOV      #TIMSGO,-(SP)
          015622 010600          MOV      #1,-(SP)
          015624 104415          MOV      SP,R0
          015626 062706 000004      TRAP    C#PNTX
2923 015632 004737 007744      ADD      #4,SP
2924 015636          JSR      PC,PRIXOR          ;PRINT THE DATA
          015636          ENDMMSG
          015636 104423  L10024:
2925          TRAP      C#MSG
2926 015640          045      116      045  TIMSGO: .ASCIZ  '#N#A TIMER A STATUS IS IN BIT 3#N#A TIMER B STATUS IS IN BIT 2'
2927          .EVEN
2928          .SBTTL  BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS
2929          ;*
2930          ;
2931          ;PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
2932          ;
2933          ;INPUTS:
2934          ;
2935          ;      R1      CONTENTS OF TSSR
2936          ;      R2      DATA WRITTEN (8 BITS)
2937          ;
2938          ;
2939          ;
2940          :-
2941          BGNMSG  BADSSR
          015740
          015740
2942 015740 010246          BADSSR::
2943 015742 042702 177400      MOV      R2,-(SP)          ;SAVE DATA TRANSFERRED
          BIC      #177400,R2      ;GET JUST ONE BYTE

```

K6

BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

```

2944 015746          PRINTB  #XFERASC,R2
      015746 010246    MOV      R2,-(SP)
      015750 012746 016000 MOV      #XFERASC,-(SP)
      015754 012746 000002 MOV      #2,-(SP)
      015760 010600    MOV      SP,R0
      015762 104414    TRAP    C$PNTB
      015764 062706 000006 ADD      #6,SP
2945 015770 012602    MOV      (SP),R2          ;RESTORE R2
2946 015772 004737 006026 JSR      PC,PRITSSR      ;DECODE TSSR CONTENTS
2947 015776          ENDMSG
      015776          L10025:
      015776 104423    TRAP    C$MSG
2948 016000 045 116 045 XFERASC: .ASCIZ  '#N#A Data Transferred = #03'

```

GLOBAL SUBROUTINES SECTION

```

2950      .SBTTL  GLOBAL SUBROUTINES SECTION
2951
2952      ;**
2953      ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2954      ; THAT ARE USED IN MORE THAN ONE TEST.
2955      ;--
2956      .SBTTL  SOFINIT - SOFT INITIALIZE OF CONTROLLER
2957
2958      ;*
2959      ;
2960      ;ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
2961      ;BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
2962      ;THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
2963      ;DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
2964      ;
2965      ;INPUTS:
2966      ;
2967      ;      R5      ADDRESS OF FIRST REGISTER
2968      ;
2969      ;OUTPUTS:
2970      ;
2971      ;      R0      CONTENTS OF TSSR, IF ERROR
2972      ;      CARRY   SET IF INIT WAS OKAY
2973      ;              CLEAR IF FATAL ERROR
2974      ;
2975      ;CALLING SEQUENCE:
2976      ;
2977      ;      MOV      #ADDRESS,R5
2978      ;      JSR      PC,SOFINIT
2979      ;      BCS      CONTINUE
2980      ;      ERRDF
2981      ;              ;REPORT FATAL ERROR
2982      ;
2983      ;-
2984      016034      SOFINIT::
2985      016034      SAVREG      ; SAVE THE REGISTERS
2986      016040      012765      000000      000002      MOV      #0,TSSR(R5)      ; DO THE INIT.
2987      016046      004737      016310      JSR      PC,WAITF      ; WAIT FOR SSR
2988      016052      016500      000002      MOV      TSSR(R5),R0      ;GET THE TSSR REGISTER
2989      016056      010004      MOV      R0,R4      ;TSSR CONTENTS
2990      016060      042704      176277      BIC      #+C<HIADDR:OFL>,R4
2991      016064      052704      002200      BIS      #SSR!NBA,R4      ;R4 HAS EXPECTED CONTENTS
2992      016070      020400      CMP      R4,R0      ;ONLY EXPECTED BITS SET ?
2993      016072      001402      BEQ      5$      ;BRANCH IF OKAY
2994      016074      000241      CLC      ;CLEAR THE CARRY FOR ERROR
2995      016076      000401      BR      10$      ;GO TO EXIT
2996      016100      000261      5$:      SEC      ;SET THE CARRY BIT
2997      016102      000207      10$:      RTS      PC      ;RETURN TO CALLER

```


ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS

```

3044 .SBTTL ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS
3045 ;
3046 ; DEFAULT DISPLAY INTERRUPT HANDLERS.
3047 ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
3048 ; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
3049 ;
3050 ;
3051 ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
3052 ;
3053 ; IOKCKIN=BIT7 ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
3054 ; IOKSTP=BIT0 ; EXPECT "STOP" INTERRUPT.
3055 ;
3056 ; INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
3057 016204 000 INTMASK: .BYTE 0
3058 ; INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
3059 016205 000 INTFLAG: .BYTE 0
3060 ;
3061 ; SAVED INTERRUPT VECTOR:
3062 016206 000000 INTVEC: .WORD 0
3063 ; SAVE CPU PC
3064 016210 000000 INTCPC: .WORD 0
3065 ;
3066 ; SUBROUTINE TO ENABLE INTERRUPTS:
3067 016212 010046 ENAINT: MOV RO,-(SP) ;SAVE RO
3068 016214 013700 002210 MOV IVEC,RO ;GET POINTER TO VECTORS
3069 016220 012720 016256 MOV #INTR,(RO)+ ;SET UP INTERRUPT VECTOR
3070 016224 012720 000340 MOV #PRI07,(RO)+
3071 016230 012600 MOV (SP)+,RO ;RESTORE RO
3072 016232 011646 MOV (SP),-(SP)
3073 016234 012766 000000 000002 MOV #0,2(SP) ;SET CPU TO LEVEL 0
3074 016242 000002 RTI
3075 ;
3076 ; SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
3077 016244 011646 DSBINT: MOV (SP),-(SP)
3078 016246 012766 000340 000002 MOV #PRI07,2(SP)
3079 016254 000002 RTI
3080 .SBTTL INTR - INTERRUPT HANDLERS
3081 ;
3082 016256 BGNSRV INTR ;DEFINE INTERRUPT ENTRY
3083 016256 012737 000001 002224 INTR:: MOV #1,INTRECV ;SET FLAG TO SHOW INTERRUPT RECEIVED
3084 016264 105037 016205 CLR B INTFLAG ;CLEAR FLAG TO SAY WE GOT INTERRUPT
3085 016270 132737 000001 016204 BIT B #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
3086 016276 001003 BNE 1$ ;BR IF YES
3087 016300 152737 000001 016205 BIS B #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.
3088 ;
3089 ; SAVE REGISTERS, MSG BUFFER, ETC.
3090 016306 1$: ENDSRV
3091 016306 L10026:
016306 000002 RTI

```


WAITF - WAIT FOR SUBSYSTEM READY

```

3093                                     .SBTTL WAITF - WAIT FOR SUBSYSTEM READY
3094                                     ;
3095                                     ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
3096                                     ;
3097                                     ; INPUTS:
3098                                     ;
3099                                     ; R5 ADDRESS OF FIRST DEVICE REGISTER
3100                                     ;
3101                                     ; OUTPUTS:
3102                                     ;
3103                                     ; R0 CONTENTS OF LAST TSSR READ
3104                                     ; CARRY SET - READY BIT SET
3105                                     ; CLR - TIMEOUT WAITING FOR READY
3106                                     ;
3107 016310 000401 WAITF:: BR 1$ ;NOP WHEN SUPER FIXED
3108 016312 104422 BREAK ; DO A SUPVSR BREAK FIRST.
      016312 104422 TRAP C$BRK
3109 016314 012746 011000 1$: MOV #11000,-(SP) ;25-APRIL-83 REV B - 1100 MSEC TIMER
3110 016320 016500 000002 2$: MOV TSSR(R5),R0 ;READ THE TSSR REGISTER
3111 016324 105700 TSTB R0 ;TEST FOR READY BIT SET
3112
3113 016326 100420 BMI 3$ ; EXIT ON STOP FLAG.
3114 016330 DELAY 1 ; WAIT 100 USEC
      016330 012727 000001 MOV #1,(PC)+
      016334 000000 .WORD 0
      016336 013727 002116 MOV L$DLY,(PC)+
      016342 000000 .WORD 0
      016344 005367 177772 DEC -6(PC)
      016350 001375 BNE .-4
      016352 005367 177756 DEC -22(PC)
      016356 001367 BNE .-20
3115 016360 005316 DEC (SP) ;REDUCE DELAY COUNT
3116 016362 001356 BNE 2$ ;RETRY UNTIL TIMER EXPIRES
3117 016364 000241 CLC ; C = 0, CONTROLLER STILL RUNNING...
3118 016366 000401 BR 4$ ;...OR HUNG-UP AFTER 300 MSEC.
3119 016370 000261 3$: SEC ; C = 1, CONTROLLER IS STOPPED.
3120 016372 005326 4$: DEC (SP)+ ;RESTORE STACK WITHOUT CHANGING CARRY BIT
3121 016374 000207 RTS PC

```

CHKTSSR - CHECK TSSR FOR READY

```

3123 .SBTTL  CHKTSSR - CHECK TSSR FOR READY
3124 ;*
3125 ;THIS ROUTINE WAITS FOR READY IN THE TSSR
3126 ;AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
3127 ;
3128 ;INPUT:
3129 ;       R5      ADDRESS OF CSR REGISTERS
3130 ;
3131 ;OUTPUT:
3132 ;       R0      CONTENTS OF TSSR
3133 ;       CARRY   SET - OKAY
3134 ;             CLR - NOT READY AMBIGUOUS, OR SC SET
3135 ;-
3136 CHKTSSR:
3137 016376 004737 016310 JSR  PC, WAITF          ;WAIT FOR READY
3138 016402 103014 BCC  20$              ;BRANCH IF TIME OUT
3139 016404 004737 016104 JSR  PC, CHKAMB        ;TSSR AMBIGUOUS?
3140 016410 103006 BCC  10$              ;BR IF YES
3141 016412 032700 100000 BIT  #SC, R0           ;SPECIAL CONDITION SET?
3142 016416 001405 BEQ  15$              ;BR IF NO
3143 016420 032700 074000 BIT  #<SCE!BIE!RMR!NXM>, R0 ;ANY ERROR BITS SET?
3144 016424 001402 BEQ  15$              ;BR IF NO
3145 016426 000241 10$: CLC                          ;SET FAILURE
3146 016430 000401 BR    20$              ;
3147 016432 000261 15$: SEC                          ;SET SUCCESS
3148 016434 000207 20$: RTS  PC                ;RETURN TO CALLER
3149 .SBTTL  XNXM - CHECK FOR NONEXISTENT MEMORY
3150 ;*
3151 ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
3152 ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
3153 ; "C" = 0, ALL ADDRESSES OK.
3154 ;
3155 ;CALL:  MOV  ADR1, R1
3156 ;       MOV  ADR2, R2
3157 ;       JSR  PC, NXM
3158 ;       RETURN
3159 016436 012737 016470 000004 XNXM: MOV  #2$, @#4      ;TEST "C" AND PROCEED.
3160 016444 012737 000200 000006 MOV  #PRI04, @#6      ; SET BUSERR VECTOR.
3161 016452 005003 CLR  R3              ;FLAG.
3162 016454 005711 1$: TST  (R1)          ;TEST THE ADDRESS(ES).
3163 ;IF ANY TRAP, CONTINUE AT 2$.
3164 016456 020102 CMP  R1, R2          ;OTHERWISE, CONTINUE HERE.
3165 016460 001407 BEQ  3$              ;BR IF FINISHED (NO NEXM'S).
3166 016462 062701 000002 ADD  #2, R1          ;SET NEXT ADDRESS...
3167 016466 000772 BR   1$              ;... AND CONTINUE.
3168 016470 005103 2$: COM  R3              ;GOT ONE, SET FLAG...
3169 016472 012716 016500 MOV  #3$, (SP)
3170 016476 000002 RTI
3171 016500 3$: CLRVEC #4          ;... AND DISMISS INTERRUPT...
3172 016500 012700 000004 MOV  #4, R0          ;... AND GIVE BACK THE VECTOR.
3173 016504 104436 TRAP C#CVEC
3174 016506 005703 TST  R3              ;DID WE CATCH ONE ??
3175 016510 001401 BEQ  .+4            ;NO, "C" = 0, SKIP NEXT.
3176 016512 000261 SEC
3177 016514 000207 RTS  PC                ;YES, "C" = 1, (R1) = NEXM ADDR.

```

TSTLOOP - CHECK ITERATION COUNT

```

3177          .SBTTL TSTLOOP - CHECK ITERATION COUNT
3178          ;*
3179          ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
3180          ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
3181          ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
3182          ;
3183          ; CALL: LOOPTO ARG
3184          ;
3185          TSTLOOP:
3186          TST      NOITS          ; ITERATIONS INHIBITED?
3187          BNE     1$             ; YES.
3188          TST      QVP           ; NO.
3189          BMI     1$             ; LOOPS DISALLOWED IN QUICK PASS.
3190          DEC     LOOPCNT        ; BUMP LOOP COUNTER.
3191          BNE     2$
3192          1$:     CLC             ; LOOP DISALLOWED, OR DONE.
3193          BR      3$
3194          2$:     SEC             ; LOOP ENABLED.
3195          3$:     RTS            PC
3196
3197          .SBTTL TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
3198          ;*
3199          ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
3200          ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
3201          ; IN THE CURRENT RUN SEQUENCE.
3202          ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
3203          ;
3204          ; INPUT:
3205          ;
3206          ;     R0     POINTER TO TEST ID ASCIZ STRING
3207          ;
3208          ; OUTPUT:
3209          ;
3210          ;     R5     ADDRESS OF FIRST DEVICE REGISTER
3211          ;
3212          ; IMPLICIT OUTPUTS:
3213          ;
3214          ;     TSTCNT UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
3215          ;
3216          ; SIDE EFFECTS:
3217          ;
3218          ;     INTERRUPT LEVEL IS RASIED TO LEVEL OF
3219          ;     THE DEVICE UNDER TEST
3220          ;
3221          ;-
3222
3223          TSTSETUP:
3224          MOV     R0, -(SP)      ; SAVE THE TEST ID MESSAGE
3225          CLR     SIFLAG        ; CLEAR "SOFT INIT" FLAG
3226          CLR     ERRK         ; CLEAR LOCAL ERROR COUNTER.
3227          CLR     EXTA         ; CLEAR ERROR EXTENSION FLAG.
3228          CLRB   INTMASK      ; CLEAR INTERRUPT MASK (CHECK ERROR)
3229          MOV     UNITN, R0     ; GET THE UNIT NUMBER.
3230          ASL    R0            ; ... AND MAKE IT A WORD OFFSET.
3231          TST    NODEV         ; DID STARTUP FIND THE DEVICE?
3232          BEQ    4$            ; BR IF YES
3233          BPL    3$            ; BR IF NOT IDLE

```

E7

TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS

```

3234 016610 052760 160000 003200 BIS #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
3235 016616 104455 ERRDF 1,NXR,NXRERR ; NO DEVICE HERE -- PRINT IT
      016620 000001 TRAP C$ERDF
      016622 003742 .WORD 1
      016624 005740 .WORD NXR
3236 016626 000407 .WORD NXRERR
      016630 052760 160001 003200 3$: BR 2$
3237 016630 052760 160001 003200 3$: BIS #160001,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
3238 016636 104455 ERRDF 2,NOINIT ; DEVICE NOT IDLE
      016640 000002 TRAP C$ERDF
      016642 004337 .WORD 2
      016644 000000 .WORD NOINIT
3239 016646 012737 177777 003114 2$: .WORD 0
3240 016654 013700 002202 MOV #-1,DUFLG ; DROP THE UNIT
      016654 013700 002202 DODU UNITN
      016660 104451 MOV UNITN,R0
      016662 104444 TRAP C$DODU
      016662 104444 DOCLN ; ABORT THE PASS
3242 016664 000423 TRAP C$DCLN
3243 BR 5$
3244 016666 104421 4$: RFLAGS RO ; GET THE OPERATOR FLAGS.
      016666 104421 TRAP C$RFLA
3245 016670 032700 001000 BIT #PNT,R0 ; PRINT THE TEST NUMBERS?
3246 016674 001412 BEQ 1$ ; BR IF NO
3247 016676 011600 MOV (SP),R0 ;GET THE ID MESSAGE
3248 016700 010046 PRINTF #TNAM,R0 ;DISPLAY THE TEST ID
      016702 012746 016744 MOV RO,-(SP)
      016706 012746 000002 MOV #TNAM,-(SP)
      016712 010600 MOV #2,-(SP)
      016714 104417 MOV SP,R0
      016716 062706 000006 TRAP C$PNTF
3249 016722 005237 002214 1$: ADD #6,SP
      016726 013700 002212 INC TSTCNT ; BUMP TEST COUNTER.
3250 016726 013700 002212 SETPRI IPRI ;PRIORITY THAT OF DEVICE
      016732 104441 MOV IPRI,R0
      016734 005726 TRAP C$SPRI
3251 016734 005726 5$: TST (SP) ;FIX UP THE STACK
3252 016736 013705 002206 MOV CSRADDR,R5 ; ADDRESS OF TSV REGISTERS ON UNIBUS
3253 016742 000207 RTS PC
3254 016744 045 123 045 TNAM: .ASCIZ '#S#T#A Test'
3255 .EVEN
3256 .SBTTL TSTEND - PRINT ERRORS RECEIVED
3257 ;
3258 ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
3259 ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
3260 ;
3261 016760 104421 TSTEND: RFLAGS RO
      016760 104421 TRAP C$RFLA
3262 016762 030027 020000 BIT RO,#IER
3263 016766 001412 BEQ 1$ ; BR IF "IER" NOT SET.
3264 016770 013746 017016 PRINTF #ESUM,ERRK ; PRINT ERROR COUNT.
      016770 013746 017016 MOV ERRK,-(SP)
      016774 012746 017020 MOV #ESUM,-(SP)
      017000 012746 000002 MOV #2,-(SP)
      017004 010600 MOV SP,R0
      017006 104417 TRAP C$PNTF

```

F7

TSTEND - PRINT ERRORS RECEIVED

3265	017010	062706	000006			ADD	#6.SP	
3266	017014	000207		14:		RTS	PC	
3267	017016	000000				ERRK:	0	; LOCAL ERROR COUNT.
3268	017020	045	101	040		ESUM:	.ASCIZ	/#A #D#A ERRORS/
3269	017037	105	122	122		EMAXDU:	.ASCIZ	/ERROR LIMIT REACHED -- DROPPING UNIT/
3270							.EVEN	

INCERK - INCREMENT LOCAL ERROR COUNT

```

3272                                     .SBTTL INCERK - INCREMENT LOCAL ERROR COUNT
3273                                     ;*
3274                                     ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
3275                                     ;-
3276 017104 005237 017016 INCERK: INC ERRK ; INCREMENT LOCAL ERROR COUNT
3277 017110 010046 MOV RO,-(SP) ; SAVE RO
3278 017112 013700 002202 MOV UNITN,RO ; GET UNIT NUMBER,
3279 017116 006300 ASL RO ; ... AND MAKE IT A WORD OFFSET.
3280 017120 062700 003200 ADD #ERTABL,RO ; RO GETS ADDRESS OF ERROR TABLE ENTRY.
3281 017124 005210 INC (RO) ; INCREMENT THE DEVICE ERROR COUNT
3282 017126 032710 007777 BIT #7777,(RO) ; DID WE OVERFLOW THE FIELD?
3283 017132 001001 BNE 1$ ; BR IF NO.
3284 017134 005310 DEC (RO) ; YES -- BACK IT UP TO 7777.
3285 017136 012600 1$: MOV (SP)+,RO ; RESTORE RO
3286 017140 000207 RTS PC ; RETURN TO CALLER.
3287
3288 017142 010046 CKEMAX: MOV RO,-(SP) ; SAVE RO
3289 017144 013700 002202 MOV UNITN,RO ; GET UNIT NUMBER
3290 017150 006300 ASL RO ; ... AND MAKE IT A WORD OFFSET
3291 017152 016000 003200 MOV ERTABL(RO),RO ; GET ERROR TABLE ENTRY
3292 017156 042700 170000 BIC #170000,RO ; EXTRACT ERROR COUNT FIELD
3293 017162 020037 002174 CMP RO,GERRMAX ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
3294 017166 103004 BHIS 1$ ; BR IF YES
3295 017170 023737 017016 002172 CMP ERRK,LERRMAX ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
3296 017176 103417 BLO 2$ ; BR IF NO
3297 017200 1$: RFLAGS RO ; GET OPERATOR FLAGS
3298 017202 104421 TRAP C#RFLA
3299 017206 032700 000040 BIT #IDU,RO ; IS DROPPING INHIBITED?
3300 017210 001013 BNE 2$ ; BR IF YES.
3301 017216 012737 177777 003114 MOV #-1,DUFLG ; NO -- DROP THE UNIT
3302 017216 104455 ERRDF 4,EMAXDU
3303 017220 000004 TRAP C#ERDF
3304 017222 017037 .WORD 4
3305 017224 000000 .WORD EMAXDU
3306 017226 .WORD 0
3307 017226 013700 002202 DODU UNITN
3308 017232 104451 MOV UNITN,RO
3309 017234 DOCLN TRAP C#DODU
3310 017236 104444 TRAP C#DCLN
3311 017236 012600 2$: MOV (SP)+,RO ; RESTORE RO
3312 017240 000207 RTS PC ; RETURN TO CALLER

```

H7

CKDROP - CHECK IF UNIT SHOULD BE DROPPED

```

3307          .SBTTL CKDROP - CHECK IF UNIT SHOULD BE DROPPED
3308          ;
3309          ; CHECK IF UNIT SHOULD BE DROPPED
3310          ;
3311 017242 010046          CKDROP: MOV    RO,-(SP)
3312 017244          FORCERROR 1$,NOTSSR
3313 017254          RFLAGS RO
3314 017256 104421          TRAP   C$RFLA
3315 017262 032700 000040  BIT    #IDU,RO
3316 017264 001010          BNE   1$
3317 017266 012737 177777 003114  MOV   (SP),RO
3318 017274          MOV   #-1,DUFLG
3319 017274 013700 002202  DODU  UNITN
3320 017300 104451          MOV   UNITN,RO
3321 017302          TRAP   C$DODU
3322 017304          DOCLN          ;ABORT THE PASS
3323 017306 104444          TRAP   C$DCLN
3324 017308 012600          1$: MOV   (SP)-,RO
3325 017310 000207          RTS    PC
3326          .SBTTL CONFIG - DETERMINE CONFIGURATION OF SYSTEM
3327          ;
3328          ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
3329          ;
3330          ; CONFIG:
3331          JSR    PC,SOFINIT
3332          RTS    PC
3333          .SBTTL KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT
3334          ;
3335          ; SUBROUTINE - ENABLE MEM MGT.
3336          ;
3337          ; KTON: TST    KTFLG          ; GOT KT?
3338          BEQ    1$                  ; NO.
3339          MOV   #1,SRO                ; YES. ENABLE KT11.
3340          1$: RTS    PC
3341          ;
3342          ; SUBROUTINE - DISABLE MEM MGT.
3343          ;
3344          ; KTOFF: TST   KTFLG          ; GOT KT11?
3345          BEQ   1$                   ; NO.
3346          NOP
3347          MOV   #0,SRO                ; DISABLE KT.
3348          1$: RTS    PC

```

SETMAP - SETUP PAR6 MAPPING

```

3350          .SBTTL  SETMAP - SETUP PAR6 MAPPING
3351
3352          ;*
3353          ;
3354          ; THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
3355          ; AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
3356          ; IS RETURNED BIASED TO PAR6.
3357          ;
3358          ; INPUTS:
3359          ;
3360          ;       R0      HIGH ORDER ADDRESS BITS
3361          ;       R1      LOW ORDER ADDRESS BITS
3362          ;
3363          ; OUTPUTS:
3364          ;
3365          ;       R0      OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
3366          ;       CARRY   SET IF SUCCESS
3367          ;               CLR IF ERROR
3368          ;
3369          ; SETMAP:
3370          ; SAVREG          ; SAVE R1-R4 UNTIL NEXT RETURN
3371          ; TST           KTFLG          ; SYSTEM HAVE ABOVE 28K?
3372          ; BEQ           10$           ; BR IF NO
3373          ; MOV           R1,R2         ; SAVE LOW ORDER BITS
3374          ; .REPT        6
3375          ; ASR           R0           ; CONVERT WORD ADDRESS TO 32W BLOCKS
3376          ; ROR           R1           ; MAKE IT DOUBLE PRECISION
3377          ; .ENDR
3378          ; BIC           #177,R1      ; ALIGN FOR LOWER 4K BOUNDARY
3379          ; CMP           R1,KTFLG     ; HIGHER THAN EXISTING MEMORY?
3380          ; BHIS        10$           ; BR IF YES
3381          ; MOV           R1,@#KIPAR5 ; SETUP MAPPING REGISTER PAR5
3382          ; BIC           #160000,R2  ; SETUP DISPLACEMENT IN PAGE
3383          ; ADD           #120000,R2  ; ADD IN PAR5 BIAS
3384          ; MOV           R2,R0       ; RETURN IN R0
3385          ; SEC           ; SET SUCCESS
3386          ; BR           15$          ;
3387          ; CLC           ; SET FAILURE
3388          ; RTS          PC           ; RETURN
3389          ; .SBTTL  REGSAV - SAVE R1-R5 ON STACK
3390          ;*
3391          ;
3392          ; ROUTINE TO
3393          ; SAVE R1 THROUGH R5 ON THE STACK
3394          ;
3395          ; CALLING SEQUENCE:
3396          ;
3397          ;       JSR      R5,REGSAV
3398          ;
3399          ; THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
3400          ; THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
3401          ; THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
3402          ; REGISTERS.
3403          ;
3404          ; THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
3405          ; CALLED VIA A JSR PC INSTRUCTION
3406          ;

```

3369	017356		
3370	017356		
3371	017362	005737	003134
3372	017366	001433	
3373	017370	010102	
3374		000006	
3375			
3376			
3377			
3378	017422	042701	000177
3379	017426	020137	003134
3380	017432	103011	
3381	017434	010137	172352
3382	017440	042702	160000
3383	017444	062702	120000
3384	017450	010200	
3385	017452	000261	
3386	017454	000401	
3387	017456	000241	
3388	017460	000207	

J7

REGSAV - SAVE R1-R5 ON STACK

```
3407  
3408           ;-  
3409 017462           REGSAV:  
3410 017462   010446           MOV       R4,-(SP)  
3411 017464   010346           MOV       R3,-(SP)  
3412 017466   010246           MOV       R2,-(SP)  
3413 017470   010146           MOV       R1,-(SP)  
3414 017472   010546           MOV       R5,-(SP)  
3415 017474   016605   000012   MOV       10.(SP),R5  
3416 017500   004736           JSR       PC,@(SP).  
3417 017502   012601           MOV       (SP)+,R1  
3418 017504   012602           MOV       (SP)+,R2  
3419 017506   012603           MOV       (SP)+,R3  
3420 017510   012604           MOV       (SP)+,R4  
3421 017512   012605           MOV       (SP)+,R5  
3422 017514   000207           RTS       PC
```

K7

GETPAT - GET 8 BIT PATTERN FROM OPERATOR

```

3424 .SBTTL GETPAT - GET 8 BIT PATTERN FROM OPERATOR
3425
3426 ;*
3427 ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
3428 ;
3429 ;INPUTS: NONE.
3430 ;
3431 ;OUTPUTS:
3432 ; RO OCTAL NUMBER FROM THE OPERATOR
3433 ;
3434 ;CALLING SEQUENCE:
3435 ; JSR PC,GETPAT
3436 ;-
3436 017516 GETPAT::
3437 017516 SAVREG ;SAVE THE GENERAL REGISTERS
3438 017522 104443 1$: GMANID DATASC,PATDAT,0,377,0,377,NO
017522 104443 TRAP C$GMAN
017524 000406 BR 10000$
017526 017552 .WORD PATDAT
017530 000022 .WORD T$CODE
017532 017554 .WORD DATASC
017534 000377 .WORD 377
017536 000000 .WORD T$LOLIM
017540 000377 .WORD T$HILIM
017542
3439 017542 10000$: BNCOMPLETE 1$ ;RETRY IF ERROR
017542 103367 BCC 1$
3440 017544 013700 017552 MOV PATDAT,RO ;DATA PATTERN FROM OPERATOR
3441 017550 000207 RTS PC ;RETURN TO CALLER
3442
3443 ;*
3444 ;LOCAL DATA AREA
3445 ;-
3446
3447 017552 000000 PATDAT: .WORD 0 ;TEMPORARY STORAGE FOR DATA
3448 017554 105 116 124 DATASC: .ASCIZ 'ENTER DATA PATTERN'
3449 .EVEN

```

GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE

```

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

```

CHKMAN - CHECK MANUAL INTERVENTION LEGALITY

```

3484          .SBTTL  CHKMAN  - CHECK MANUAL INTERVENTION LEGALITY
3485          ;*
3486          ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
3487          ;
3488          ;INPUT:
3489          ;
3490          ;       NONE.
3491          ;
3492          ;OUTPUT:
3493          ;
3494          ;       CARRY  0      MANUAL INTERVENTION NOT ALLOWED
3495          ;               1      MANUAL INTERVENTION IS OK
3496          ;
3497          ;SIDE EFFECTS:
3498          ;
3499          ;       A MESSAGE IS DISPLAYED WARNING THAT TEST IS
3500          ;       NOT EXECUTED IF MANUAL INTERVENTION IS NOT
3501          ;       ALLOWED.
3502          ;
3503          ;
3504          ;-
3505
3506 020022      CHKMAN::
3507 020022      SAVREG          ;SAVE THE REGISTERS
3508 020026      MANUAL          ;SEE IF MANUAL INTERVENTION OK
3509 020026      104450         TRAP   C$MANI
3510 020030      103411         BCOMPLETE 1$      ;BRANCH IF ALLOWED
3511 020032      012746 020056  PRINTF #NOMAN      ;PRINT THE WARNING MESSAGE
3512 020036      012746 000001  MOV   #NOMAN,-(SP)
3513 020042      010600         MOV   #1,-(SP)
3514 020044      104417         MOV   SP,RO
3515 020046      062706 000004  TRAP  C$PNTF
3516 020052      000241         ADD   #4,SP
3517 020054      000207         CLC          ;CLEAR CARRY FOR ERROR
3518 020056      045      116      045  NOMAN:  RTS   PC      ;RETURN
3519 020056      045      116      045  NOMAN:  .ASCIZ  '%N%A *** Manual Intervention not Allowed - Test Aborted ***'
3520 020056      045      116      045  NOMAN:  .even

```

N7

ENVIRN - SETUP FREE DIAGNOSTIC SPACE

```

3517          .SBTTL  ENVIRN  - SETUP FREE DIAGNOSTIC SPACE
3518
3519          ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
3520
3521          ; ENVIRN: MEMORY  R0
3522          TRAP          C$MEM
3523          MOV          R0,FREE          ; GET 1ST FREE ADDRESS...
3524          ADD          #2,FREE
3525          MOV          (R0),FRESIZ     ; ...AND WORD COUNT.
3526          SUB          #4,FRESIZ
3527          MOV          L$UNIT,R2      ; GET NUMBER OF UNITS
3528          SUB          #7,FRESIZ     ; TAKE AWAY 7 WORDS PER UNIT
3529          DEC          R2
3530          BNE          10$
3531          MOV          FREE,R0        ; GET FIRST FREE ADDRESS
3532          ADD          FRESIZ,R0      ; POINT TO LAST FREE ADDRESS
3533          SUB          #2,R0         ; BACKUP 1 WORD
3534          MOV          R0,FREEHI     ; STORE LAST FREE ADDRESS
3535          NOP
3536          MOV          #BDVPCR,R1    ; *****
3537          MOV          R1,R2        ; GET BDV11 PCR ADDRESS
3538          ADD          #2,R2        ; COPY TO R2
3539          JSR          PC,XNXM      ; SET THE RANGE
3540          BCC          15$          ; SEE IF WE HAVE ONE
3541          BR           40$          ; OK TO SET FLAGS
3542          MOV          BDVPCR,R1     ; RETURN WITH FLAGS CLEAR
3543          ADD          #1,R1         ; SAVE PCR CONTENTS
3544          MOV          #BDVPCR,R2   ; ADD ONE TO IT
3545          INC          (R2)         ; GET BDV11 PCR ADDRESS
3546          MOV          BDVPCR,R3   ; TRY TO WRITE TO IT
3547          CMP          R1,R3       ; GET RESULTS
3548          BNE          20$         ; DID IT CHANGE?
3549          INC          T23A        ; NO, MUST BE 11/23B
3550          BIC          #170000,L$HIME ; SET THE FLAG
3551          NOP                    ; SUPERVISOR COULD BE WRONG
3552          PRINTF      #M8186      ; BR 40$ FOR RELEASE
3553          BR           40$        ; TELL THE SYSTEM TYPE
3554          INC          T23B        ; RETURN
3555          NOP                    ; SET THE FLAG
3556          PRINTF      #M8189      ; BR 40$ FOR RELEASE
3557          RTS           PC        ; TELL THE SYSTEM TYPE

```

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

3559                                     .SBTTL KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS
3560                                     ;*
3561                                     ;
3562                                     ;ROUTINE TO INIT KT-11
3563                                     ;
3564                                     ;-
3565
3566 020332                               KTINIT:
3567 020332 005037 003134                 CLR     KTFLG           ; INIT >28K MEMORY FLAG
3568 020336 005037 003136                 CLR     KTENABLE      ; INIT TEST >28K FLAG
3569 020342 023727 002120 001577        CMP     L$HIME,#1577   ; GOT ENOUGH MEMORY (>28K)?
3570 020350 101454                       BLOS   9$             ; NO.
3571 020352 013700 000004                 MOV     @#ERRVEC,R0   ; SAVE OLD ERR VEC PTR.
3572 020356 012737 020470 000004        MOV     #2$,@#ERRVEC  ; SET ERR VEC PTR.
3573 020364 005737 177572                 TST     @#SRO         ; GOT KT11?
3574 020370 000240                       NOP                    ; (TRAP IF NO).
3575 020372 013737 002120 003134        MOV     L$HIME,KTFLG ; YES. SET KT FLAG.
3576 020400 022737 010000 003134        CMP     #10000,KTFLG ; MORE THAN 18 BITS?
3577 020406 100404                       BMI    4$             ; NO
3578 020410 042737 003777 003134        BIC     #3777,KTFLG  ;
3579 020416 000403                       BR     5$             ;
3580 020420 042737 000177 003134 4$:   BIC     #177,KTFLG   ;
3581 020426 010037 000004 5$:         MOV     R0,@#ERRVEC  ; RESTORE OLD ERR VEC PTR.
3582 020432 005000                       CLR     R0            ; R0 = AR DATA.
3583 020434 012701 172340                 MOV     #KIPAR0,R1   ; R1 = KI REGS PTR.
3584 020440 012761 077406 177740 1$:   MOV     #77406,-40(R1) ; SET DESCRIPTOR REG.
3585 020446 010021                       MOV     R0,(R1)+     ; SET KIPAR REG.
3586 020450 062700 000200                 ADD     #200,R0      ; BUMP AR DATA BY "4K".
3587 020454 020027 002000                 CMP     R0,#2000     ; AT "I/O"?
3588 020460 001367                       BNE    1$            ; NO.
3589 020462 012741 177600                 MOV     #177600,-(R1) ; YES. SET KTPAR7 FOR I/O.
3590 020466 000405                       BR     9$
3591
3592 020470 012716 020476                 2$:   MOV     #6$, (SP)  ; SET UP RETURN
3593 020474 000002                       RTI                    ; RTI TO NEXT LOCATION
3594
3595 020476 010037 000004                 6$:   MOV     R0,@#ERRVEC ; RESTORE OLD ERR VEC PTR.
3596
3597 020502 000207                 9$:   RTS     PC

```

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

3599
3600      ;*      SUBROUTINE TO SET EXTENDED FEATURES SWITCH
3601      ;
3602      ;      Requires that SOFINIT and WRTCHR have been done previous to call.
3603      ;
3604      ;
3605      ;INPUTS:
3606      ;      R5      CURRENT UNIT NUMBER
3607      ;OUTPUTS:
3608      ;      The Extended Features Switch is set.
3609      ;
3610      ;-
3611
3612      020504      INVERT::
3613      020504      013705      002206      mov      csraddr,r5
3614      020510      004737      016034      jsr      pc,sofinit
3615      020514      103406      bcs      25$
3616      020516      010001      mov      r0,r1
3617      020520      errrdf      errno,ofierr,sfimsg
3618      020520      104455      TRAP      C$ERDF
3619      020522      002261      .WORD      1201
3620      020524      003654      .WORD      fierr
3621      020526      012074      .WORD      sfimsg
3622      020530      104455      trap c$erdf
3623      020532      25$:      ckloop
3624      020532      104406      TRAP      C$CLP1
3625      020534      013737      002202      076340      mov      unitn,t39dsw
3626      020542      012704      076320      mov      #t39pk2,r4
3627      020546      004737      010662      jsr      pc,wrtchr
3628      020552      103406      bcs      50$
3629      020554      010001      mov      r0,r1
3630      020556      errhrd      errno,wrtmsg,sfimsg
3631      020556      104456      TRAP      C$ERHRD
3632      020560      002261      .WORD      1201
3633      020562      005060      .WORD      wrtmsg
3634      020564      012074      .WORD      sfimsg
3635      020566      104456      trap c$erhrd
3636      020570      50$:      ckloop
3637      020570      104406      TRAP      C$CLP1
3638      020572      013701      075650      mov      t39bfr+12,r1
3639      020576      032701      000200      bit      #bit7,r1
3640      020602      001020      bne      1$
3641      020604      012737      100206      020650      10$:      MOV      #100206,CMDPKT      ; WRT SUB-SYS MEM CMD
3642      020612      012737      020660      020652      MOV      #WSMBK,CMDPKT+2      ; MSG BUF ADDR
3643      020620      012737      000006      020656      MOV      #6,CMDPKT+6      ; BYTE COUNT
3644      020626      012737      100010      020660      MOV      #100010,WSMBK      ; INVERT THE SWITCH
3645      020634      012704      020650      MOV      #CMDPKT,R4      ; SET CMDPKT INTO R4
3646      020640      004737      010662      JSR      PC,WRTCHR      ; DO IT
3647      020644      000207      1$:      RTS      PC      ; RETURN
3648
3649      ;      COMMAND PACKET.
3650
3651      .      =      <..+3>&177774      ;MUST BE ON MOD 4 BOUNDRY.
3652
3653      CMDPKT:: 0      ;1ST WORD IS TS05 COMMAND.
3654      0      ;2ND WORD IS THE BUFFER LOW ADDRESS.
3655      0      ;3RD WORD IS THE BUFFER HIGH ADDRESS.

```

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

3646 020656 000000          0          ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
3647
3648          ;          WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
3649
3650 020660 000000          WSMBK:: 0          ;1ST WORD:: SEL 0
3651 020662 000000          0          ;2ND WORD:: SEL 2
3652 020664 000000          0          ;3RD WORD:: SEL 4
3653          .EVEN
3654
3655          ;*          SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
3656          ;
3657          ;
3658          ;INPUTS:
3659          ;OUTPUTS:
3660          ;          The NXMFLG is set if we can test.
3661          ;          The NXMLO and NXMHI addresses are setup.
3662          ;
3663          ;-
3664 020666          MEMCK::
3665
3666 020666          SAVREG          ;SAVE THE REGISTERS
3667 020672 005037 003140          CLR          NXMFLG          ;CLEAR THE FLAG
3668 020676 005037 003142          CLR          NXMLO          ;CLEAR THE TEST ADDRESS LO
3669 020702 005037 003144          CLR          NXMHI          ;CLEAR THE TEST ADDRESS HI
3670 020706 005737 003150          TST          T23B          ;IS IT A 11/23B?
3671 020712 001407          BEQ          1$          ;NO
3672 020714 023727 002120 007777          CMP          L$HIME,#7777          ; GREATER THAN 128K
3673 020722 103406          BLO          2$          ; NO
3674 020724 004737 021042          JSR          PC,NXMTST          ;SETUP THE ADDRESS
3675 020730 000427          BR          13$          ;SET THE FLAG AND EXIT
3676 020732 005737 003146          1$: TST          T23A          ;IS IT A 11/23A?
3677 020736 001413          BEQ          4$          ;NO
3678 020740 023727 002120 005777          2$: CMP          L$HIME,#5777          ;GREATER THAN 96K
3679 020746 101023          BHI          14$          ;YES,23A/23B WITH 128K MEMORY
3680 020750 023727 002120 003777          CMP          L$HIME,#3777          ;GREATER THAN 64K BUT LESS THAN 92K?
3681 020756 103403          BLO          4$          ;NO, CHECK 24K
3682 020760 004737 021042          JSR          PC,NXMTST          ;SETUP THE ADDRESS
3683 020764 000411          BR          13$          ;SET THE FLAG AND EXIT
3684 020766 023727 002120 001577          4$: CMP          L$HIME,#1577          ;GREATER THAN 24K BUT LESS THAN 64K?
3685 020774 103410          BLO          14$          ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
3686 020776 004737 021042          JSR          PC,NXMTST          ;SETUP THE ADDRESS
3687 021002 062737 000077 003144          ADD          #77,NXMHI          ;FOOL THE 11/02 & 11/03
3688 021010 005237 003140          13$: INC          NXMFLG          ;SET THE FLAG
3689 021014 000411          BR          15$          ;EXIT
3690 021016 000410          14$: BR          15$          ;NOP FOR PRINTOUT
3691 021020          PRINTF          #NOMEM          ;TELL THEM & EXIT ***NO PRINT*****
          021020 012746 005462          MOV          #NOMEM,-(SP)
          021024 012746 000001          MOV          #1,-(SP)
          021030 010600          MOV          SP,R0
          021032 104417          TRAP          C$PNTF
          021034 062706 000004          ADD          #4,SP
3692 021040 000207          15$: RTS          PC          ;RETURN
3693
3694          ;*
3695          ;          SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
3696          ;
3697          ;OUTPUTS:NXMLO,NXMHI          ;SETUP WITH NXM ADDRESS

```


KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

3698
3699
3700
3701 021042 013701 002120      NXMTST: MOV      L$HIME,R1      ;GET TOP OF MEMORY
3702 021046 062701 000200      ADD      #200,R1      ;MAKE IT I/O BLOCK OR OTHER NXM
3703 021052 042701 000177      BIC      #177,R1
3704 021056 010102              MOV      R1,R2      ;RESAVE RESULTS
3705              000006      .REPT    6
3706              .ASL     R1      ;PUT IN PLACE FOR XFER
3707              .ENDR
3708 021074 010137 003142      MOV      R1,NXML0    ;SAVE TEST ADDRESS LOW
3709              000012      .REPT    10.
3710              .ASR     R2      ;PUT IN PLACE FOR XFER
3711              .ENDR
3712 021124 042702 177700      BIC      #177700,R2  ;DON'T WANT ILA!
3713 021130 010237 003144      MOV      R2,NXMHI    ;SAVE TEST ADDRESS HIGH
3714 021134 000207      RTS      PC          ;RETURN
3715
3716 021136              ENDMOD
3725              .TITLE  TSV4 - MISCELLANEOUS SECTIONS
3726
3727 021136              BGNMOD  TSV4
021136
3728
3734
3735
3736
3737
3738 021136              .SBTTL  PROTECTION TABLE
021136              BGNPROT
3739 021136 177777 177777 177777  L$PROT:: .WORD  -1. -1. -1. -1      ;NO DEVICE PROTECTION REQUIRED.
3740 021146              ENDPROT

```

INITIALIZE SECTION

```

3742                                     .SBTTL INITIALIZE SECTION
3743
3744                                     ;**
3745                                     ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
3746                                     ;AT THE BEGINNING OF EACH PASS.
3747
3748                                     ;
3749                                     ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
3750                                     ;IF "CONTINUE", NOTHING IS REQUIRED.
3751                                     ;
3752                                     ;--
3753                                     ;
3754                                     ;INSERT TEMPORARY JUMP TO ODT
3755                                     ;-
3756                                     BGNINIT
3757                                     L$INIT::
3758                                     40$: CLR      EXTFEA
3759                                     CLR      NXMFLG
3760                                     MOV      #EPRT1,EPRTSW           ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
3761                                     CLR      SIFLAG             ;CLEAR "SOFT INIT" FLAG
3762                                     CLR      KTENABLE          ;CLEAR TEST ABOVE 28K FLAG
3763                                     CLR      RAMSIZ             ;CLEAR RAM SIZE FOR RAMERR ROUTINE
3764                                     READEF   #EF.CONTINUE
3765                                     MOV      #EF.CONTINUE,RO
3766                                     TRAP    C$REFG
3767                                     BNCOMPLETE 1$
3768                                     BCC     1$
3769                                     CMP     UNITN,L$UNIT           ;UNIT IN RANGE?
3770                                     BHIS   4$                   ;BR IF NO.
3771                                     TST    DUFLG                ;DROPPED UNIT?
3772                                     BMI    NXTU                 ;BR IF YES
3773                                     MOV    UNITN,R1
3774                                     ASL    R1
3775                                     TST    ERTABL(R1)
3776                                     BEQ    SETU
3777                                     BIT    #BIT14,ERTABL(R1)     ;DROPPED?
3778                                     BNE    NXTU
3779                                     EXIT   INIT                 ;DO NOTHING IF "CONTINUE".
3780                                     TRAP   C$EXIT
3781                                     .WORD  L10030-.
3782                                     1$: READEF   #EF.NEW
3783                                     MOV    #EF.NEW,RO
3784                                     TRAP   C$REFG
3785                                     BNCOMPLETE NXTU           ;TAKE NEXT UNIT IF NOT NEW PASS.
3786                                     BCC    NXTU
3787                                     READEF   #EF.START
3788                                     MOV    #EF.START,RO
3789                                     TRAP   C$REFG
3790                                     BCOMPLETE 2$
3791                                     BCS    2$
3792                                     READEF   #EF.RESTART
3793                                     MOV    #EF.RESTART,RO
3794                                     TRAP   C$REFG
3795                                     BNCOMPLETE 31$
3796                                     BCC    31$
3797                                     2$: BRESET
3798                                     TRAP   C$RESET           ;1ST PASS, BUS-INIT...
3799                                     ;BUS RESET.

```

INITIALIZE SECTION

```

3783 021310 005037 002214      CLR      TSTCNT      ;NUMBER OF TESTS RUN IN PASS
3784 021314 005037 002222      CLR      FATFLG     ;CLEAR FATAL ERROR COUNT
3785 021320 005037 003146      CLR      T23A      ;CLEAR 11/23A FLAG
3786 021324 005037 003150      CLR      T23B      ;CLEAR 11/23B FLAG
3787      ;
3788      ;
3789      ;
3790 021330 005037 003402      CLR      SKIPT      ;CLEAR THE SUBTEST "SKIPPER"
3791 021334      ;
3792 021334 012737 177777 002204 20$:  MOV      #-1,QVP    ;...QUICK VERIFY...
3793 021342 004737 020152      JSR      PC,ENVIRN  ;SET ENVIRONMENT.
3794 021346 004737 020332      JSR      PC,KTINIT  ;INITIALIZE KT MEMORY MANAGEMENT
3795 021352 012700 003200      MOV      #ERTABL,RO
3796 021356 005020 003400 30$:  CLR      (RO)+      ;CLEAR THE ERROR TABLE
3797 021360 020027 003400      CMP      RO,#ERTABE
3798 021364 103774      BLO     30$
3799 021366 000404      BR      4$
3800 021370 005037 002204 31$:  CLR      QVP
3801 021374 000137 021444      JMP      PASRPT    ;GO REPORT THE STATUS
3802
3803 021400      ;
3804 021400 012737 177777 002202 4$:  NEWPAS: MOV      #-1,UNITN  ;INIT UNIT NUMBER...
3805 021406 005037 002220      CLR      DEVCNT    ;CLEAR COUNT OF DEVICES RUNNING
3806 021412      ;
3807 021412 104422      ;
3808 021414 005237 002202 002012  NXTU:  BREAK   C$BRK
3809 021426 103423      TRAP    UNITN
3810 021430 012737 177777 003114  INC     UNITN,L$UNIT  ;...AND SET NEXT UNIT NUMBER.
3811 021436 000401      CMP     UNITN,L$UNIT
3812 021440      BLO     SETU
3813 021440 104444      MOV     #-1,DUFLG
3814 021442 000240      BR     11$
3815 021444 023727 002012 000001 11$:  DOCLN  C$DCLN
3816 021452 101752      TRAP   C$DCLN
3817 021454 005737 002220      NOP
3818 021460 001747      ;
3819 021462 104421      ;
3820 021464 032700 000100 11$:  CMP     L$UNIT,#1  ;HOW MANY UNITS SELECTED?
3821 021470 001343      BLOS   NEWPAS     ;BR IF ONLY 1
3822      TST     DEVCNT  ;ARE ANY STILL RUNNING?
3823 021472      BEQ     NEWPAS   ;BR IF NO
3824 021472 104424      RFLAGS RO
3825 021474 000741      TRAP   C$RFLA
3826 021476      BIT     #ISR,RO  ;SHOULD WE PRINT STATISTICS
3827 021476      BNE     NEWPAS  ;BR IF NO
3828 021504      ;
3829 021504 103342      ;
3830 021506 005037 003114      DORPT
3831 021512 005237 002220      TRAP   C$DRPT
3832 021516 012001 002206      BR     NEWPAS
3833 021520 010137 002206      ;
3834 021520      ;
3835 021520      ;
3836 021520      ;
3837 021520      ;
3838 021520      ;
3839 021520      ;
3840 021520      ;
3841 021520      ;
3842 021520      ;
3843 021520      ;
3844 021520      ;
3845 021520      ;
3846 021520      ;
3847 021520      ;
3848 021520      ;
3849 021520      ;
3850 021520      ;
3851 021520      ;
3852 021520      ;
3853 021520      ;
3854 021520      ;
3855 021520      ;
3856 021520      ;
3857 021520      ;
3858 021520      ;
3859 021520      ;
3860 021520      ;
3861 021520      ;
3862 021520      ;
3863 021520      ;
3864 021520      ;
3865 021520      ;
3866 021520      ;
3867 021520      ;
3868 021520      ;
3869 021520      ;
3870 021520      ;
3871 021520      ;
3872 021520      ;
3873 021520      ;
3874 021520      ;
3875 021520      ;
3876 021520      ;
3877 021520      ;
3878 021520      ;
3879 021520      ;
3880 021520      ;
3881 021520      ;
3882 021520      ;
3883 021520      ;
3884 021520      ;
3885 021520      ;
3886 021520      ;
3887 021520      ;
3888 021520      ;
3889 021520      ;
3890 021520      ;
3891 021520      ;
3892 021520      ;
3893 021520      ;
3894 021520      ;
3895 021520      ;
3896 021520      ;
3897 021520      ;
3898 021520      ;
3899 021520      ;
3900 021520      ;
3901 021520      ;
3902 021520      ;
3903 021520      ;
3904 021520      ;
3905 021520      ;
3906 021520      ;
3907 021520      ;
3908 021520      ;
3909 021520      ;
3910 021520      ;
3911 021520      ;
3912 021520      ;
3913 021520      ;
3914 021520      ;
3915 021520      ;
3916 021520      ;
3917 021520      ;
3918 021520      ;
3919 021520      ;
3920 021520      ;
3921 021520      ;
3922 021520      ;
3923 021520      ;
3924 021520      ;
3925 021520      ;
3926 021520      ;
3927 021520      ;
3928 021520      ;
3929 021520      ;
3930 021520      ;
3931 021520      ;
3932 021520      ;
3933 021520      ;
3934 021520      ;
3935 021520      ;
3936 021520      ;
3937 021520      ;
3938 021520      ;
3939 021520      ;
3940 021520      ;
3941 021520      ;
3942 021520      ;
3943 021520      ;
3944 021520      ;
3945 021520      ;
3946 021520      ;
3947 021520      ;
3948 021520      ;
3949 021520      ;
3950 021520      ;
3951 021520      ;
3952 021520      ;
3953 021520      ;
3954 021520      ;
3955 021520      ;
3956 021520      ;
3957 021520      ;
3958 021520      ;
3959 021520      ;
3960 021520      ;
3961 021520      ;
3962 021520      ;
3963 021520      ;
3964 021520      ;
3965 021520      ;
3966 021520      ;
3967 021520      ;
3968 021520      ;
3969 021520      ;
3970 021520      ;
3971 021520      ;
3972 021520      ;
3973 021520      ;
3974 021520      ;
3975 021520      ;
3976 021520      ;
3977 021520      ;
3978 021520      ;
3979 021520      ;
3980 021520      ;
3981 021520      ;
3982 021520      ;
3983 021520      ;
3984 021520      ;
3985 021520      ;
3986 021520      ;
3987 021520      ;
3988 021520      ;
3989 021520      ;
3990 021520      ;
3991 021520      ;
3992 021520      ;
3993 021520      ;
3994 021520      ;
3995 021520      ;
3996 021520      ;
3997 021520      ;
3998 021520      ;
3999 021520      ;
4000 021520      ;

```

INITIALIZE SECTION

```

3833
3834 021524 012001      MOV      (R0),R1      ;GET VECTOR ADDRESS.
3835                   ;MOV      (R0),R2      ;GET INTERRUPT PRIORITY
3836                   ;MOV      R2,IPRI    ;SET INTERRUPT PRIORITY.
3837 021526 010137 002210 MOV      R1,IVEC      ;SET INTERRUPT VECTOR POINTER...
3838 021532 012721 016256 MOV      @INTR,(R1)+  ;...VECTOR...
3839 021536 013721 002212 MOV      IPRI,(R1)+  ;...AND PRIORITY.
3840
3841 021542             1$:
3842                   ;      TST      QVP      ;1ST PASS ??
3843                   ;      BEQ      5$      ;NO, SKIP THE PASS 1 STUFF.
3844
3845                   ;
3846                   ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
3847                   ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
3848                   ;
3849 021542 013701 002202      MOV      UNITN,R1
3850 021546 006301          ASL      R1
3851 021550 052761 100000 003200 BIS      @BIT15,ERTABL(R1) ;SAY DEVICE RUNNING
3852 021556 005037 005774      CLR      EXTA      ;CLEAR ERROR EXTENSION FLAG.
3853 021562 023727 002012 000001 CMP      L$UNIT,#1      ;ARE WE TESTING MULTIPLE UNITS?
3854 021570 101416          BLOS    10$      ;BR IF NO.
3855 021572          RFLAGS  RO      ;YES -- GET OPERATOR FLAGS.
3856 021572 104421          TRAP   C$RFLA
3857 021574 032700 001000      BIT      @PNT,RO      ;SHOULD WE PRINT UNIT #?
3858 021600 001412          BEQ      10$      ;BR IF NOT.
3859 021602          PRINTF  @PUNIT,UNITN ;PRINT THE UNIT #
3860 021602 013746 002202      MOV      UNITN,-(SP)
3861 021606 012746 021674      MOV      @PUNIT,-(SP)
3862 021612 012746 000002      MOV      @2,-(SP)
3863 021616 010600          MOV      SP,RO
3864 021620 104417          TRAP   C$PNTF
3865 021622 062706 000006      ADD      @6,SP
3866 021626          10$:
3867 021626 005037 003116      CLR      NODEV
3868 021632 013701 002206      MOV      CSRADDR,R1 ;ADDRESS OF FIRST REGISTER
3869 021636 010102          MOV      R1,R2      ;START OF REGISTERS
3870 021640 062702 000002      ADD      @TSSR,R2   ;ADDRESS OF TSSR REGISTER
3871 021644 004737 016436      JSR      PC,XNXM    ;TEST BOTH CONTROLLER REGISTERS...
3872 021650 103005          BCC     2$      ;...AND BR IF ALL OK.
3873 021652 010137 003116      MOV      R1,NODEV   ;FLAG DEVICE AS NON-EXISTENT
3874 021656 012737 177777 003114 MOV      @-1,DUFLG  ;DROP THIS UNIT.
3875 021664          2$:
3876                   ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
3877                   ;
3878                   5$:
3879 021664          SETPRI  @PRI00      ;ENABLE INTERRUPTS.
3880 021664 012700 000000      MOV      @PRI00,RO
3881 021670 104441          TRAP   C$SPRI
3882 021672          ENDINIT
3883 021672          L10030:
3884 021672 104411          TRAP   C$INIT
3885 021674          045    116    045 PUNIT: .ASCIZ /#N#A***** TESTING UNIT #D2#A *****/
3886 021676          .EVEN

```

ADD AND DROP UNITS SECTIONS

```

3878                                     .SBTTL  ADD AND DROP UNITS SECTIONS
3879
3880
3881      ;**
3882      ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
3883      ; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
3884      ; OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
3885      ;--
3885 021742      BGNAU
3886 021742      010001      L$AU::
3887 021744      006301      MOV      RO,R1      ; GET UNIT TO BE ADDED (RO)
3888 021746      052761      100000 003200      ASL      R1      ; MAKE IT A WORD INDEX
3889 021754      042761      040000 003200      BIS      #100000,ERTABL(R1) ; SET THE "ACTIVE" BIT
3890 021762      PRINTF      #1$,RO      ; CLEAR THE "DROPPED" BIT
3890 021762      010046      MOV      RO,-(SP)
3890 021764      012746      022010      MOV      #1$,-(SP)
3890 021770      012746      000002      MOV      #2$,-(SP)
3890 021774      010600      MOV      SP,RO
3890 021776      104417      TRAP     C$PNTF
3890 022000      062706      000006      ADD      #6,SP
3891 022004      EXIT      AU
3891 022004      000167      .WORD   J$JMP
3891 022006      000026      .WORD   L10031-2-
3892 022010      045      116      045 1$: .ASCIZ  /#N$A UNIT #D$A ADDED/
3893
3894
3895 022036      ENDAU      ; UNUSED.
3895 022036
3895 022036      104452      L10031: TRAP     C$AU
3896
3897      ;**
3898      ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
3899      ; TO BE REMOVED FROM THE TEST LIST.
3900
3901      ; SUPVSR DOES THE "DROPPING". THIS IS JUST TO TELL THE MAN.
3902      ; "DROPPED" UNITS ARE RE-SELECTED ON OPERATOR "STA" OR "ADD"
3903      ; COMMAND, OTHERWISE REMAIN INACTIVE. THE "DISPLAY" COMMAND
3904      ; WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
3905      ; WHICH ARE STILL ACTIVE.
3906      ; UPON ENTRY, RO CONTAINS THE UNIT TO BE DROPPED.
3907 022040      BGNDU
3907 022040      L$DU::
3908 022040      012737      177777 003114      MOV      #-1,DUFLG
3909 022046      010001      MOV      RO,R1
3910 022050      006301      ASL      R1
3911 022052      052761      140000 003200      BIS      #140000,ERTABL(R1) ; SAY DROPPED
3912 022060      000240      000240 000240      240,240,240 ; ??????????
3913 022066      PRINTF      #1$,RO
3913 022066      010046      MOV      RO,-(SP)
3913 022070      012746      022114      MOV      #1$,-(SP)
3913 022074      012746      000002      MOV      #2$,-(SP)
3913 022100      010600      MOV      SP,RO
3913 022102      104417      TRAP     C$PNTF
3913 022104      062706      000006      ADD      #6,SP
3914 022110      EXIT      DU
3914 022110      000167      .WORD   J$JMP
3914 022112      000030      .WORD   L10032-2-

```

ADD AND DROP UNITS SECTIONS

```

3915 022114      045      116      045 1$: .ASCIZ /#N#A UNIT #D#A DROPPED/
3916          .EVEN
3917 022144          ENDDU
      022144          L10032: TRAP C#DU
      022144 104453
3918
3919          ;**
3920          ; AUTO-DROP CODE SECTION.
3921 022146          ;--
      022146          BGNAUTO
3922 022146 013705 002206          L#AUTO::
3923 022152 012703 000550          MOV CSRADDR,R5          ;POINT TO DEVICE REGISTER
3924 022156 004737 016310          MOV #360.,R3          ;ENOUGH TIME FOR 2400' REEL TO REWIND
3925 022162 103420          10$: JSR PC,WAITF          ;WAIT FOR SSR TO SET
3926 022164          BCS 20$          ;LEAVE WHEN SSR IS SET
      022164 012727 000372          DELAY 250.          ;WAIT FOR .25 SECONDS
      022170 000000          MOV #250.,(PC)+
      022172 013727 002116          .WORD 0
      022176 000000          MOV L#DLY,(PC)+
      022200 005367 177772          .WORD 0
      022204 001375          DEC -6(PC)
      022206 005367 177756          BNE -.4
      022212 001367          DEC -22(PC)
      022214 005303          BNE -.20
3927 022214 005303          DEC R3          ;BUMP COUNTER DOWN
3928 022216 001357          BNE 10$          ;KEEP GOING
3929 022220 004737 017242          JSR PC,CKDROP          ;TRY AND DROP UNIT
3930 022224          20$: ENDAUTO
3931 022224          L10033: TRAP C#AUTO          ; UNUSED.
      022224 104461

```

CLEAN-UP AND REPORT CODING SECTIONS

```

3933          .SBTTL CLEAN-UP AND REPORT CODING SECTIONS
3934
3935
3936          ;**
3937          ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
3938          ; EXECUTED AT THE END OF EACH PASS (OR SUB-PASS).
3939          ; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
3940          ;--
3940 022226          BGNCLN
3941 022226          L$CLEAN::
3942 022226 013705 002206          MOV     CSRADDR,R5          ;POINT TO DEVICE REGISTER
3943 022232 005737 003114          TST     DUFLG          ;"DROPPED" FLAG IS SET ON...
3944 022236 100405          BMI     1$          ;...AND GROSS CONTROLLER FAULT...
3945          ;...DON'T TRY TO XCT CLEANUP CODE.
3946 022240 012765 000000 000002          MOV     #0,TSSR(R5)          ;DO SOFT INIT
3947 022246 004737 016310          JSR     PC,WAITF
3948 022252          1$:
3949 022252          2$:          ENDCLN
3949 022252          L10034:          TRAP    C$CLEAN
3950          ;**
3951          ; THE REPORT CODING SECTION CONTAINS THE
3952          ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
3953          ;--
3954 022254          BGNRPT
3955 022254          L$RPT::
3955 022254 012746 022516          PRINTS #DEVSUM
3956 022260 012746 000001          MOV     #DEVSUM,-(SP)
3957 022264 010600          MOV     #1,-(SP)
3958 022266 104416          MOV     SP,R0
3959 022270 062706 000004          TRAP   C$PNTS
3960 022274 010246          ADD     #4,SP
3961 022276 010346          MOV     R2,-(SP)
3962 022300 010446          MOV     R3,-(SP)
3963 022302 012704 003200          MOV     R4,-(SP)
3964 022306 005003          MOV     #ERTABL,R4          ; GET START OF ERROR TABLE.
3965 022310 011402          CLR     R3          ; CLEAR UNIT NUMBER
3966 022312 001467          1$:          MOV     (R4),R2          ; GET ERROR TABLE ENTRY & TEST IT.
3967 022314 100066          BEQ     4$          ; ZERO IF UNIT NOT RUN
3968 022316 032702 040000          BPL     4$
3969 022322 001015          BIT     #BIT14,R2          ; WAS UNIT DROPPED?
3970 022324 042702 170000          BNE     2$          ; BR IF YES
3971 022330          BIC     #C7777,R2          ; GET ERROR COUNT FIELD
3972 022330 010246          PRINTS #DEVONL,R3,R2          ; PRINT
3973 022332 010346          MOV     R2,-(SP)
3974 022334 012746 022553          MOV     R3,-(SP)
3975 022340 012746 000003          MOV     #DEVONL,-(SP)
3976 022344 010600          MOV     #3,-(SP)
3977 022346 104416          MOV     SP,R0
3978 022350 062706 000010          TRAP   C$PNTS
3979 022354 000446          ADD     #10,SP
3980 022356 020227 160000          BR     4$
3981 022362 001012          2$:          CMP     R2,#160000          ; WAS UNIT NON-EXISTENT?
3982 022364 010346          BNE     3$          ; BR IF NO
3983 022366 012746 022623          PRINTS #DEVNXR,R3
3984          MOV     R3,-(SP)
3985          MOV     #DEVNXR,-(SP)

```

CLEAN-UP AND REPORT CODING SECTIONS

```

022372 012746 000002      MOV    #2,-(SP)
022376 010600      MOV    SP,R0
022400 104416      TRAP  C#PNTS
022402 062706 000006      ADD    #6,SP
3972 022406 000431      BR     4$
3973 022410 020227 160001      3$:   CMP    R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
3974 022414 001012      BNE   30$              ; BR IF NO.
3975 022416      PRINTS #DEVNRD,R3
022416 010346      MOV    R3,-(SP)
022420 012746 022705      MOV    #DEVNRD,-(SP)
022424 012746 000002      MOV    #2,-(SP)
022430 010600      MOV    SP,R0
022432 104416      TRAP  C#PNTS
022434 062706 000006      ADD    #6,SP
3976 022440 000414      BR     4$
3977 022442 042702 170000      30$:  BIC   #1C7777,R2
3978 022446      PRINTS #DEVDR0,R3,R2
022446 010246      MOV    R2,-(SP)
022450 010346      MOV    R3,-(SP)
022452 012746 022766      MOV    #DEVDR0,-(SP)
022456 012746 000003      MOV    #3,-(SP)
022462 010600      MOV    SP,R0
022464 104416      TRAP  C#PNTS
022466 062706 000010      ADD    #10,SP
3979 022472 062704 000002      4$:   ADD    #2,R4
3980 022476 005203      INC    R3
3981 022500 020427 003400      CMP    R4,#ERTABE
3982 022504 103701      BLO   1$
3983 022506 012604      MOV    (SP),R4
3984 022510 012603      MOV    (SP),R3
3985 022512 012602      MOV    (SP),R2
3986 022514      ENDRPT ; UNUSED.
022514      L10035:
022514 104425      TRAP  C#RPT
3987
3988 022516      045 116 045 DEVSUM: .ASCIZ /#N#ADEVICE STATUS SUMMARY:#N/
3989 022553      045 101 040 DEVONL: .ASCIZ /#A UNIT #D3#A ONLINE, ERRORS = #D#N/
3990 022623      045 101 040 DEVNXR: .ASCIZ /#A UNIT #D3#A DROPPED, NON-EXISTENT REGISTER#N/
3991 022705      045 101 040 DEVNRD: .ASCIZ /#A UNIT #D3#A DROPPED, NOT READY AT STARTUP#N/
3992 022766      045 101 040 DEVDR0: .ASCIZ /#A UNIT #D3#A DROPPED, ERRORS = #D#N/
3993      .EVEN
3994
3995 023036      ENDMOD
3996

```


M8

CLEAN-UP AND REPORT CODING SECTIONS

4000
4001
4008
4009 023036
023036
4015
4023

.TITLE TSV5 - HARDWARE TESTS

TSV5:: BGNMOD TSV5

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

```

4025          .SBTTL TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS
4026
4027          ;* TEST DESCRIPTION:
4028          ;
4029          ; This test verifies that a Hardware Initialize command
4030          ; invoked after a Write Characteristics command sets up
4031          ; the Command, Message and Characteristic image blocks
4032          ; in the controller ram correctly.
4033
4034          ; TEST STEPS:
4035
4036          REPEAT FOR LOOPCNT
4037          BEGIN
4038          ; Do WRITE CHARACTERISTICS command.
4039          ; If the NBA bit in the TSSR register is NOT=0 then Print Error.
4040          ; Write to TSSR register to soft initialize the controller
4041          ; If controller RAM 310-377 NOT=0 then Print Error
4042          END
4043          ;--
4044
4045          BGNTST
4046          023036          T1::
4047          023036          MOV      #TST13ID,R0          ;ASCII MESSAGE TO IDENTIFY TEST
4048          023036          JSR      PC,TSTSETUP          ;DO INITIAL TEST SETUP
4049          023042          004737          016550          MOV      #10.,LOOPCNT          ;PERFORM 10 ITERATIONS
4050          023046          012737          000012          002216          T13LOOP:
4051          023054          JSR      PC,T13REST          ;SET PACKET TO START-UP VALUES
4052          023054          004737          023746
4053          023060          MOV      #TSTBLK*10.,R3          ;START OF TEST DATA
4054          023064          012704          023430          MOV      #T13PACKET,R4          ;GET THE ADDRESS OF COMMAND PACKET
4055          023070          012764          000010          000006          MOV      #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
4056          023076          004737          016034          54:
4057          023076          JSR      PC,SOFINIT          ;WRITE TO TSSR TO SOFT INITIALIZE
4058          023102          103405          BCS     104          ;BR IF SOFT INIT OKAY
4059          023104          010001          MOV     R0,R1          ;SAVE CONTENTS OF TSSR
4060          023106          ERRDF   ERRNO,SFIERR,SFIMSG          ;DEVICE FATAL DURING INIT
4061          023106          104455          TRAP   C4ERDF
4062          023110          000144          .WORD  100
4063          023112          003654          .WORD  SFIERR
4064          023114          012074          .WORD  SFIMSG
4065
4066          ;Do WRITE CHARACTERISTICS command.
4067          104:
4068          023116          005037          002222          CLR     FATFLG          ;CLEAR FATAL ERROR FLAG
4069          023122          010465          000000          MOV     R4,TSD8(R5)          ;SET THE PACKET ADDRESS TO EXECUTE
4070          023126          004737          016376          JSR     PC,CHKTSSR          ;WAIT FOR SSR TO SET
4071          023132          FORCERROR 124          ;@@DFORCE ERROR IF FORCER=1
4072          023146          103407          BCS     154          ;BR IF CARRY SET (GOOD RETURN)
4073          023150          010001          MOV     R0,R1          ;SAVE CONTENTS OF TSSR
4074          023152          NEXT.ERRNO
4075          023152          124:          ERRDF   ERRNO,T13SSR,PKTSSR          ;DEVICE FATAL SSR FAILED TO SET
4076          023152          104455          TRAP   C4ERDF
4077          023154          000145          .WORD  101
4078          023156          023657          .WORD  T13SSR
4079          023160          012106          .WORD  PKTSSR
4080          023162          005237          002222          154:          INC     FATFLG          ;SET FATAL ERROR FLAG
4081          023166          CKLOOP          ;LOOP ON ERROR, IF FLAG SET

```

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

```

023166 104406
4078 023170 016501 000002      MOV    TSSR(R5),R1      ;GET THE CONTENTS OF TSSR      TRAP    C#CLP1
4079 023174 012702 000200      MOV    #SSR,R2         ;EXPECTED CONTENTS OF TSSR
4080 023200 032701 000100      BIT    #OFL,R1         ;IS OFF-LINE BIT SET ?
4081 023204 001402              BEQ    25$              ;BRANCH IF NOT OFF-LINE
4082 023206 052702 000100      BIS    #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
4083
4084      ;If the NBA bit in the TSSR register is NOT=0 then Print Error.
4085 023212 25$:
4086 023212      FORCERROR    27$      ;@@D
4087 023226 020201      CMP    R2,R1          ;DOES EXPECTED MATCH RECEIVED ?
4088 023230 001404      BEQ    30$            ;OKAY IF MATCH
4089 023232      NEXT.ERRNO
4090 023232 27$:      ERRHRD  ERRNO,T13NBA,PKTSSR ;NBA NOT ZERO
      023232 104456      TRAP    C#ERHRD
      023234 000146      .WORD  102
      023236 023604      .WORD  T13NBA
      023240 012106      .WORD  PKTSSR
4091 023242 30$:      CKLOOP              ;LOOP ON ERROR ?      TRAP    C#CLP1
      023242 104406
4092
4093      ;Write to TSSR register to soft initialize the controller
4094 023244 40$:
4095 023244 004737 016034      JSR    PC,SOFINIT     ;WRITE TO TSSR TO SOFT INITIALIZE
4096 023250      FORCERROR    42$      ;@@D
4097 023264 103405      BCS    50$            ;BR IF SOFT INIT OKAY
4098 023266 010001      MOV    R0,R1         ;SAVE CONTENTS OF TSSR
4099 023270      NEXT.ERRNO
4100 023270 42$:      ERRDF  ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
      023270 104455      TRAP    C#ERDF
      023272 000147      .WORD  103
      023274 003654      .WORD  SFIERR
      023276 012074      .WORD  SFIMSG
4101
4102      ;If controller RAM 310-377 NOT=0 then Print Error
4103 023300 012704 000310 50$:      MOV    #310,R4        ;START WITH LOC 310
4104 023304 005002      CLR    R2             ;MEMORY EXPECTED SHOULD BE 000000
4105 023306 105065 000000      CLRB  TSDB(R5)        ;SET MAINTENANCE MODE
4106 023312 004737 016376      JSR    PC,CHKTSSR     ;WAIT FOR SSR READY
4107 023316 010465 000000 60$:      MOV    R4,TSDB(R5)    ;SELECT RAM ADDRESS
4108 023322 004737 016376      JSR    PC,CHKTSSR     ;WAIT FOR SSR READY
4109 023326 116501 000000      MOVB  TSBA(R5),R1     ;READ LOC CONTENTS
4110 023332      FORCERROR    62$,NOTSSR ;@@D
4111 023342 120102      CMPB  R1,R2          ;CHECK MEMORY FOR 000000
4112 023344 001406      BEQ    70$            ;BRANCH IF DATA OKAY
4113 023346      NEXT.ERRNO
4114 023346 62$:      ERRDF  ERRNO,T13MEM,RAMEXP ;MEMORY NOT ZERO AFTER INIT
      023346 104455      TRAP    C#ERDF
      023350 000150      .WORD  104
      023352 023545      .WORD  T13MEM
      023354 015570      .WORD  RAMEXP
4115 023356 005237 002222 70$:      INC    FATFLG         ;SET THE FATAL ERROR FLAG
4116 023362      CKLOOP
4117 023362 104406      TRAP    C#CLP1
      023364      ESCAPE  TST          ;EXIT ON FATAL ERROR      TRAP    C#ESCAPE
      023364 104410      .WORD  L10036-.
      023366 000426

```

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

```

4118
4119 023370 005204      82$:  INC      R4          ;LOOK AT NEXT RAM LOC.
4120 023372 020427 000400  CMP      R4,#400      ;AT TOP OF RAM ADDRESS SPACE
4121 023376 001347      BNE      60$          ;BRANCH TILL ALL MEMORY TESTED
4122
4123
4124 023400 005737 002222      TST      FATFLG       ;ANY FATAL ERRORS ?
4125 023404 001402      BEQ      160$        ;BRANCH IF NOT
4126 023406 004737 017242      JSR      PC,CKDROP   ;TRY TO DROP THE UNIT
4127 023412 004737 016516 160$:  JSR      PC,TSTLOOP ;DONE ALL ITERATIONS?
4128 023416 103002      BCC      165$        ;BR IF YES
4129 023420 000137 023054      JMP      T13LOOP     ;LOOP UNTIL ITERATION COUNT DONE
4130 023424
4131 023424      EXIT      TST
      023424 104432      TRAP      C$EXIT
      023426 000366      .WORD    L10036-.
4132
4133
4134
4135      ;*
4136      ;LOCAL STORAGE FOR THIS TEST
4137      ;-
4141 023430
4142 023430 100004      T13PACKET:          ;COMMAND PACKET FOR TEST
      .WORD    100004      ;WRITE CHARACTERISTICS COMMAND, WITH ACK
4143 023432 023440      .WORD    T13DATA     ;ADDRESS OF CHARACTERISTICS BLOCK
4144 023434 000000      .WORD    0
4145 023436 000010      .WORD    8.          ;STARTING VALUE OF BLOCK SIZE
4146
4147 023440
4148 023440 023452      T13DATA:           ;CHARACTERISTICS DATA BLOCK
      .WORD    T13BFR     ;ADDRESS OF MESSAGE BUFFER
4149 023442 000000      .WORD    0
4150 023444 000016      .WORD    14.         ;LENGTH OF MESSAGE BUFFER
4151 023446 000000 000000      .WORD    0,0
4152
4153 023452      T13BFR: .BLKW 8.    ;MESSAGE BUFFER
4154      ;LOCAL TEXT MESSAGES FOR TEST
4155      ;-
4156
4157 023472      111      156      151 TST13ID: .ASCIZ 'Initialization After WRITE CHARACTERISTICS'
4158 023545      111      156      143 T13MEM: .ASCIZ 'Incorrect RAM Data After Init'
4159
4160 023604      127      122      111 T13NBA: .ASCIZ 'WRITE CHARACTERISTICS Command Not Accepted'
4161 023657      103      157      156 T13SSR: .ASCIZ 'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
4162
4163
4164
4165      ;*
4166      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
4167      ;
4168      ;-
4169
4170      .EVEN
4171
4172 023746      T13REST:
4173 023746      SAVREG          ;SAVE THE REGISTERS
4174 023752 012701 023430      MOV      #T13PACKET,R1 ;START OF THE PACKET
4175 023756 012721 100004      MOV      #100004,(R1)  ;WRITE CHARACTERISTICS WITH ACK

```

TEST 1: INITIALIZE AFTER WRITE CHARACTERISTICS

```

4176 023762 012721 023440 MOV #T13DATA,(R1)+ ;ADDRESS OF CHAR DATA BLOCK
4177 023766 005021 CLR (R1)+ ;EXTENDED ADDRESS
4178 023770 012721 000010 MOV #8,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
4179 023774 012721 023452 MOV #T13BFR,(R1)+ ;ADDRESS OF MESSAGE BUFFER
4180 024000 005021 CLR (R1)+
4181 024002 012721 000016 MOV #14,(R1)+ ;LENGTH OF MESSAGE BUFFER
4182 024006 005021 CLR (R1)+
4183 024010 005011 CLR (R1)
4184 024012 000207 RTS PC ;RETURN
4185 024014 ENDTST

```

L10036: TRAP C#ETST

```

4186 024014 104401
4187
4188

```

.SBTTL TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```

4189 ;*
4190 ; THIS TEST VERIFIES THAT THE WRITE SUBSYSTEM MEMORY COMMAND WITH A
4191 ; BSELO SELECT CODE OF 0 (NO-OP) EXECUTES CORRECTLY. IT ALSO
4192 ; VERIFIES THAT A WRITE SUBSYSTEM MEMORY COMMAND WITH A NON-ZERO
4193 ; MODE FIELD IS REJECTED. THE TEST FURTHER VERIFIES MICROPROGRAM
4194 ; COMMAND DECODING AND HANDLING SEQUENCES.
4195 ;
4196 ;-
4197 024016 BGNTST
      024016
4202 024016 012700 025631 MOV #TST14ID,RO ;ASCII MESSAGE TO IDENTIFY TEST
4203 024022 004737 016550 JSR PC,TSTSETUP ;DO INITIAL TEST SETUP
4204 024026 012737 000024 002216 MOV #20,,LOOPCNT ;PERFORM 20 ITERATIONS
4205 024034 T14LOOP: BGNSUB ;////////// BEGIN SUBTEST ////////////
      024034
      024034 104402 T2.1: TRAP C#BSUB
4207 024036 004737 025676 JSR PC,T14REST ;SET PACKET TO INITIAL VALUES
4208 024042 004737 025734 JSR PC,T14RST ;SET PACKET TO INITIAL VALUES
4209 024046 SETPRI #PRI00 ;LOWER PRIORITY TO ALLOW INTERRUPTS
      024046 012700 000000 MOV #PRI00,RO
      024052 104441 TRAP C#SPRI
4210 024054 5#:
4211 024054 BGNSEG ;>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>>>
      024054 104404 TRAP C#BSEG
4212 024056 004737 016034 JSR PC,SOFINIT ;DO SOFT INIT OF CONTROLLER
4213 024062 103405 BCS 10# ;BR IF SOFT INIT = OK
4217 024064 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
4218 024066 ERRDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
      024066 104455 TRAP C#ERDF
      024070 000311 .WORD 201
      024072 003654 .WORD SFIERR
      024074 012074 .WORD SFIMSG
4219 024076 10#:
4220 024076 012704 025170 MOV #T14PK2,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
4221 024102 004737 010662 JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
4222 024106 103405 BCS 11# ;BR, IF COMMAND ISSUED OK
4226 024110 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
4227 024112 ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      024112 104456 TRAP C#ERHRD
      024114 000312 .WORD 202
      024116 005060 .WORD WRTMSG

```


F9

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```

4273 024276 001402          BEQ      60$      ;BRANCH IF NOT
4274 024300 004737 017242  JSR      PC,CKDROP   ;TRY TO DROP THE UNIT
4275 024304          60$:
4276          ;*
4277          ;
4278          ;TEST 2, SUBTEST 2
4279          ;
4280          ;CHECK THAT NON-ZERO MODE BITS BEING SET CAUSES
4281          ;WRITE SUBSYSTEM MEMORY COMMAND TO BE REJECTED
4282          ;
4283          ;-
4284
4285 024304          BGNSUB          ;////////// BEGIN SUBTEST ///////////
      024304          T2.2:
      024304 104402          TRAP      C#BSUB
4286
4287 024306          SETPRI #PRI00       ;LOWER PRIORITY TO ALLOW INTERRUPTS
      024306 012700 000000          MOV      #PRI00,R0
      024312 104441          TRAP      C#SPRI
4288 024314 012704 025170          5$:   MOV      #T14PK2,R4       ;GET THE ADDRESS OF COMMAND PACKET
4289 024320 004737 025676          JSR      PC,T14REST    ;RESTORE PACKET TO STARTING VALUES
4290 024324 004737 025734          JSR      PC,T14RST    ;RESTORE PACKET TO STARTING VALUES
4291 024330          BGNSEG          ;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
      024330 104404          TRAP      C#BSEG
4292
4293 024332 004737 016034          JSR      PC,SOFINIT   ;DO SOFT INIT OF CONTROLLER
4294 024336 103405          BCS     10$          ;BR IF SOFT INIT = OK
4298 024340 010001          MOV      R0,R1       ;SAVE CONTENTS OF TSSR
4299 024342          ERRDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
      024342 104455          TRAP      C#ERDF
      024344 000317          .WORD   207
      024346 003654          .WORD   SFIERR
      024350 012074          .WORD   SFIMSG
4300 024352          10$:
4301 024352 004737 010662          JSR      PC,WRTCHR    ;ISSUE WRITE CHARACTERISTICS
4302 024356 103405          BCS     11$          ;BR, IF COMMAND ISSUED OK
4306 024360 010001          MOV      R0,R1       ;SAVE CONTENTS OF TSSR
4307 024362          ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE SUBSYSTEM MEMORY FAILED
      024362 104456          TRAP      C#ERHRD
      024364 000320          .WORD   208
      024366 005060          .WORD   WRTMSG
      024370 012074          .WORD   SFIMSG
4308 024372          11$:
4309 024372 005037 002224          CLR      INTRECV     ;CLEAR INTERRUPT RECEIVED FLAG
4310 024376 012704 024550          MOV      #T14PACKET,R4 ;SET-UP WITH WRT SUBSYS MEM PACKET
4311 024402 052714 007000          BIS      #007000,(R4) ;NON-ZERO COMMAND MODE BITS
4312 024406 010465 000000          MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS
4313 024412 004737 016310          JSR      PC,WAITF    ;WAIT FOR SSR TO SET
4314 024416 103405          BCS     15$          ;BR IF CARRY SET (GOOD RETURN)
4315 024420 010001          MOV      R0,R1       ;SAVE CONTENTS OF TSSR
4319 024422          ERRDF ERRNO,T14SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      024422 104455          TRAP      C#ERDF
      024424 000321          .WORD   209
      024426 025363          .WORD   T14SSR
      024430 012106          .WORD   PKTSSR
4320 024432          15$:   CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      024432 104406          TRAP      C#CLP1

```


TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```

4369
4370
4372          025170
4374 025170 025170      T14PK2:  .=<.*10>&177770      ;COMMAND PACKET FOR TEST
4375 025170 100204      .WORD    100204      ;WRITE CHARA. MEM. CMND., WITH IE, ACK
4376 025172 025200      .WORD    T14DTA      ;ADDRESS OF SELECT DATA BLOCK
4377 025174 000000      .WORD    0
4378 025176 000010      .WORD    8.          ;STARTING VALUE OF BLOCK SIZE
4379
4380
4381 025200          T14DTA:          ;SELECT DATA BLOCK
4382 025200 024566      .WORD    T14BFR      ;ADDRESS OF MESSAGE BUFFER
4383 025202 000000      .WORD    0
4384 025204 000400      .WORD    256.        ;LENGTH OF MESSAGE BUFFER
4385 025206 000000 000000 .WORD    0,0
4386
4387
4388          ;*
4389          ;LOCAL TEXT MESSAGES FOR TEST
4390          ;-
4391
4392 025212          127      122      111  T14NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
4393 025266          127      122      111  T142REJ: .ASCIZ  'WRITE SUBSYSTEM MEMORY Not Rejected With Non-Zero Mode Field'
4394 025363          103      157      156  T14SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
4395 025453          105      170      160  T14NINT: .ASCIZ  'Expected Interrupt Not Received On WRITE SUBSYSTEM MEMORY'
4396 025545          111      156      143  T14TSBA: .ASCIZ  'Incorrect TSBA Address After WRITE SUBSYSTEM MEMORY'
4397 025631          102      141      163  TST14ID: .ASCIZ  'Basic WRITE SUBSYSTEM MEMORY Command'
4398
4399
4400
4401
4402          ;*
4403          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
4404          ;WRITE SUBSYSTEM MEMORY COMMAND
4405          ;
4406          ;-
4407
4408 025676          T14REST:          ;SAVE THE REGISTERS
4409 025676          SAVREG          ;START OF THE PACKET
4410 025702 012701 024550      MOV      #T14PACKET,R1      ;WRITE SUBSYSTEM MEM. WITH ACK, IE
4411 025706 012721 100206      MOV      #100206,(R1)+      ;ADDRESS OF DATA BLOCK
4412 025712 012721 024560      MOV      #T14DATA,(R1)+    ;EXTENDED ADDRESS
4413 025716 005021          CLR      (R1)+              ;SIZE OF DATA BLOCK IN BYTES
4414 025720 012721 000006      MOV      #6.,(R1)+         ;CLEAR BSELO AND BSEL1
4415 025724 005021          CLR      (R1)+              ;CLEAR SEL2
4416 025726 005021          CLR      (R1)+              ;CLEAR DATA AREA
4417 025730 005011          CLR      (R1)
4418 025732 000207          RTS      PC                ;RETURN
4419
4420
4421 025734          T14RST:          ;SAVE THE REGISTERS
4422 025734          SAVREG          ;START OF THE PACKET
4423 025740 012701 025170      MOV      #T14PK2,R1        ;WRITE CHARA. WITH ACK, IE
4424 025744 012721 100204      MOV      #100204,(R1)+     ;ADDRESS OF CHARAISTICS DATA BLOCK
4425 025750 012721 025200      MOV      #T14DTA,(R1)+    ;EXTENDED ADDRESS
4426 025754 005021          CLR      (R1)+              ;SIZE OF DATA BLOCK IN BYTES
4427 025756 012721 000010      MOV      #8.,(R1)+

```

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

4428	025762	012721	024566
4429	025766	005021	
4430	025770	012721	000400
4431	025774	005021	
4432	025776	005011	
4433	026000	005037	024566
4434	026004	000207	
4435	026006		
	026006		
	026006	104401	

```

MOV    #T14BFR,(R1)+
CLR    (R1)+
MOV    #256.,(R1)+
CLR    (R1)+
CLR    (R1)
CLR    T14BFR
RTS    PC
ENDTST

```

```

;MESSAGE BUFFER ADDRESS
;LENGTH OF MESSAGE BUFFER
;CLEAR 1ST LOC IN MESSAGE BUFFER
;RETURN

```

```

L10037: TRAP    C$ETST

```

TEST 2: BASIC WRITE SUBSYSTEM MEMORY COMMAND

```

4437
4438           .SBTTL TEST 3: DMA MEMORY ADDRESSING
4439
4440           ;**
4441           ; TEST 3
4442           ;
4443           ; TEST DESCRIPTION
4444           ;
4445           ; This test verifies that the controller can properly address and
4446           ; access all available CPU memory (other than that occupied by the
4447           ; diagnostic and diagnostic supervisor code) for both reading (DATI)
4448           ; and writing (DATO). Verified are the LSI-11 Bus drivers for all
4449           ; available address lines. Up to this point only 16 bits have been
4450           ; used for DMA transfers.
4451           ;
4452           ; TEST STEPS
4453           ;
4454           ; REPEAT FROM 1 TO LOOPCNT
4455           ; BEGIN
4456           ; Do Subtest 1 - Verify GET STATUS selected locations
4457           ; Do Subtest 2 - Verify message packets selected locations
4458           ; Do Subtest 3 - Verify Characteristic data selected locations
4459           ; Do Subtest 4 - Verify NXM to selected invalid addresses
4460           ; END
4461           ;
4462           ;--
4463
4464 026010           BGNTST
4465 026010
4469 026010 012700 030060           MOV #TST12ID,R0           ;ASCII MESSAGE TO IDENTIFY TEST
4470 026014 004737 016550           JSR PC,TSTSETUP           ;DO INITIAL TEST SETUP
4471 026020 012737 000012 002216           MOV #10.,LOOPCNT           ;PERFORM 10 ITERATIONS
4472 026026 005237 003152           INC T3BFLG           ;SET TEST FLAG
4473 026032 004737 020666           JSR PC,MEMCK           ;CHECK MEMORY
4474
4475 026036           T12LOOP:           ;LOOP ON TEST LABEL
4476
4477           .SBTTL TEST 3: SUBTEST 1: GET STATUS SELECTED LOCATIONS
4478           ;**
4479           ; TEST 3: SUBTEST 1:
4480           ;
4481           ; SUBTEST DESCRIPTION:
4482           ;
4483           ; This subtest verifies the controller can fetch a get status
4484           ; command from all available memory locations.
4485           ; Two word blocks are tested one at a time by first setting
4486           ; all available memory to a background pattern of 125252.
4487           ; A Get Status command is then executed to various addresses in
4488           ; each available memory 4k word block. The various addresses
4489           ; are determined by floating a 1 then a 0 through the address bits.
4490           ;
4491           ; TEST STEPS:
4492           ;
4493           ; BEGIN
4494           ; Write to TSSR to soft initialize
4495           ; Do a WRITE CHARACTERISTICS to setup a message buffer
4496           ;

```

TEST 3: SUBTEST 1: GET STATUS SELECTED LOCATIONS

```

4497      : REPEAT FOR SELECTED VALID ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
4498      : BEGIN
4499      : Get a valid modulo-4 test address
4500      : Do a GET STATUS command from the test address
4501      : END
4502      : END
4503      :--
4504
4505 026036      BGNSUB                               ;////////// BEGIN SUBTEST //////////
      026036      T3.1: TRAP C4BSUB
      026036 104402
4506
4507
4508      ;Write to TSSR to soft initialize
4509 026040 004737 016034      JSR PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
4510 026044 103405      BCS 15$      ;BR IF SOFT INIT = OK
4511 026046      NEXT.ERRNO
4512 026046 010001      MOV RO,R1      ;SAVE CONTENTS OF TSSR
4513 026050      ERRDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
      026050 104455      TRAP C4ERDF
      026052 000455      .WORD 301
      026054 003654      .WORD SFIERR
      026056 012074      .WORD SFIMSG
4514
4515      ;Do a WRITE CHARACTERISTICS to setup a message buffer
4516 026060 15$:
4517 026060 012704 027650      MOV #T12PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
4518 026064 004737 031230      JSR PC,T12SWRT      ;RESTORE PACKET TO STARTING VALUES
4519 026070 005037 003136      CLR KTENABLE      ;TURN OFF KT-11
4520 026074 010465 000000      MOV R4,TSDB(R5)      ;SET THE PACKET ADDRESS
4521 026100 004737 016376      JSR PC,CHKTSSR      ;WAIT FOR SSR TO SET
4522 026104      FORCERROR 17$
4523 026120 103405      BCS 20$      ;BR IF SSR SET IN CHKTSSR
4524 026122 010001      MOV RO,R1      ;SAVE CONTENTS OF TSSR
4525 026124      NEXT.ERRNO
4526 026124 17$:
      026124 104455      ERRDF ERRNO,T12WRTSSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      026126 000456      TRAP C4ERDF
      026130 030162      .WORD 302
      026132 012106      .WORD T12WRTSSR
      .WORD PKTSSR
4527
4528      ;Verify a Get Status can be fetched from each address
4529      ;Get a valid modulo-4 test address
4530      ;Do a GET STATUS command from the test address
4531 026134 005037 002222      20$: CLR FATFLG      ;CLEAR FATAL ERROR FLAG
4532 026140 005037 027720      CLR T12KT      ;TEST ABOVE 28K SWITCH
4533 026144 012702 027724      MOV #T12BLK,R2      ;POINT TO TEST PATTERN TABLE
4534 026150      T121LOOP:
4535 026150 005037 003136      CLR KTENABLE      ;TURN OFF ABOVE 28K TEST FLAG
4536 026154 012201      MOV (R2)+,R1      ;GET TEST PATTERN ADDRESS
4537 026156 005000      CLR RO      ;ASSUME NO TEST ABOVE 28K
4538 026160 005737 027720      TST T12KT      ;TEST ABOVE 28K THIS TIME?
4539 026164 001407      BEQ 25$      ;BR IF NO
4540 026166 016200 177776      MOV -2(R2),RO      ;GET TEST PATTERN AGAIN
4541 026172 042700 177774      BIC #1<A1716>,RO      ;SAVE 18 BIT ADDRESS ONLY
4542 026176 012737 000001 003136      MOV #1,KTENABLE      ;TURN ON ABOVE 28K TEST FLAG
4543 026204 004737 030726      25$: JSR PC,T12CONVERT ;CONVERT TEST PATTERN TO TEST ADDRESS

```

TEST 3: SUBTEST 1: GET STATUS SELECTED LOCATIONS

```

4544 026210 103034          BCC      65$          ;BR IF INVALID PACKET ADDRESS
4545 026212 013704 027714  MOV      T12LOADD,R4  ;COPY CURRENT PACKET LOW ADDRESS
4546 026216 013703 027712  MOV      T12HIADD,R3  ;COPY CURRENT PACKET HIGH ADDRESS
4547 026222 004737 031276  JSR      PC,T12SETGET  ;SETUP CURRENT PACKET TO GET STATUS
4548 026226 042703 177774  BIC      #7C<A1716>,R3 ;SAVE ADDRESS BITS 17-16
4549 026232 050304          BIS      R3,R4        ;SETUP 18 BIT PACKET ADDRESS
4550 026234 004737 017334  JSR      PC,KTOFF     ;TURN OFF KT-11
4551 026240 010465 000000  MOV      R4,TSD8(R5)  ;SET THE PACKET ADDRESS TO EXECUTE
4552 026244 004737 016376  JSR      PC,CHKTSSR   ;WAIT FOR SSR TO SET
4553 026250          FORCERROR 32$
4554 026264 103405          BCS      40$          ;BR IF SSR SET IN CHKTSSR
4555 026266 010001          MOV      R0,R1        ;SAVE CONTENTS OF TSSR
4556 026270          NEXT_ERRNO
4557 026270          ERRDF  ERRNO,T12GETSSR,PKTGETS ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C#ERDF
                                .WORD    303
                                .WORD    T12GETSSR
                                .WORD    PKTGETS
                                SET
                                TRAP      C#CLP1
4558 026300          40$: CKLOOP          ;LOOP ON ERROR, IF FLAG
4559 026302          65$:
4560 026302          FORCEEXIT 80$
4561 026312 020227 030056  CMP      R2,#T12TBE   ;DONE ALL TSTBLK TEST PATTERNS?
4562 026316 103002          BHIS     70$          ;BR IF YES
4563 026320 000137 026150  JMP      T121LOOP     ;DO ANOTHER MODULO- 4 ADDRESS
4564 026324 005737 027720  TST     T12KT         ;DONE ABOVE 28K TESTING TOO?
4565 026330 003012          BGT     80$          ;BR IF YES
4566 026332 005737 003134  TST     KTFLG         ;ANY MEMORY ABOVE 28K ON SYSTEM?
4567 026336 001407          BEQ     80$          ;BR IF NO
4568 026340 012737 000001 027720  MOV      #1,T12KT     ;SET SWITCH
4569 026346 012702 027724  MOV      #T12BLK,R2  ;RESET TEST PATTERN TABLE
4570 026352 000137 026150  JMP      T121LOOP     ;DO ABOVE 28K TESTING
4571 026356 004737 017334  JSR      PC,KTOFF     ;TURN OFF KT11
4572 026362          80$: ENDSUB          ;////////// END SUBTEST ////////////
                                L10043:
                                TRAP      C#ESUB
4573 026364 005737 002222  TST     FATFLG        ;ANY FATAL ERRORS ?
4574 026370 001402          BEQ     100$         ;BRANCH IF NOT
4575 026372 004737 017242  JSR      PC,CKDROP    ;TRY TO DROP THE UNIT
4576 026376          100$:

```

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

;; TEST 3: SUBTEST 2:

; SUBTEST DESCRIPTION:

```

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

```

; TEST STEPS:

4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593

TEST 3: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS

```

4594      ; BEGIN
4595      ; Write to TSSR to soft initialize
4596      ; Do a WRITE CHARACTERISTICS to setup a message buffer to compare
4597      ;
4598      ; REPEAT FOR SELECTED ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
4599      ; BEGIN
4600      ; Get a valid modulo-4 test address
4601      ; Set the packet message buffer to the TEST ADDRESS
4602      ; Do a WRITE CHARACTERISTICS
4603      ; Restore the test message buffer to background pattern
4604      ; END
4605      ; END
4606      ;--
4607
4608 026376      BGNSUB      ;//////////////// BEGIN SUBTEST //////////////////
4609 026376      104402      T3.2:      TRAP      C4BSUB
4610 026376
4611      ;Write to TSSR to soft initialize
4612 026400 004737 016034      JSR      PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
4613 026404 103405      BCS      15$      ;BR IF SOFT INIT = OK
4614 026406      NEXT.ERRNO
4615 026406 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4616 026410      ERRDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL ERROR DURING INIT
4617 026410 104455      TRAP      C4ERDF
4618 026412 000460      .WORD      304
4619 026414 003654      .WORD      SFIERR
4620 026416 012074      .WORD      SFIMSG
4621
4622      ;Do a WRITE CHARACTERISTICS to setup a message buffer to compare
4623 15$:
4624 026420 012704 027650      MOV      #T12PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
4625 026424 004737 031230      JSR      PC,T12SWRT      ;SET PACKET TO WRITE CHARACTERISTICS
4626 026430 004737 017334      JSR      PC,KTOFF      ;TURN OFF KT-11
4627 026434 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS
4628 026440 004737 016376      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
4629 026444      FORCERROR      17$
4630 026460 103405      BCS      20$      ;BR IF SSR SET IN CHKTSSR
4631 026462 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4632 026464      NEXT.ERRNO
4633 17$:      ERRDF      ERRNO,T12WRTSSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
4634 026464 104455      TRAP      C4ERDF
4635 026466 000461      .WORD      305
4636 026470 030162      .WORD      T12WRTSSR
4637 026472 012106      .WORD      PKTSSR
4638
4639      ;Get a valid modulo-4 test address
4640      ;Set the packet message buffer to the test address
4641      ;Do a WRITE CHARACTERISTICS
4642 20$:      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
4643      MOV      #T12BLK,R3      ;POINT TO TEST PATTERN TABLE
4644 T122LOOP:
4645      MOV      (R3),R1      ;GET TEST PATTERN ADDRESS
4646      MOV      R1,R0      ;GET ADDRESS ALL "18 BITS"
4647      BIC      #177774,R0      ;LEAVE ONLY A17 AND A16
4648      BIC      #3,R1      ;GET RID OF A17 AND A16

```

TEST 3: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS

```

4641 026520 004737 030726      JSR    PC,T12CONVERT      ;CONVERT TEST PATTERN TO TEST ADDRESS
4642 026524 103402              BCS    25$                ;BR IF VALID MESSAGE BUFFER ADDRESS
4643 026526 000137 026624      JMP    150$              ;GET ANOTHER TEST PATTERN TO TRY
4644 026532 012704 027650      25$:  MOV    #T12PACKET,R4  ;SET THE COMMAND PACKET ADDRESS
4645 026536 004737 031230      JSR    PC,T12SWRT        ;SETUP T12PACKET TO WRITE CHAR.
4646 026542 013737 027714 027660  MOV    T12LOADD,T12DATA  ;SETUP LOW ORDER MESSAGE BUFFER ADD.
4647 026550 013737 027712 027662  MOV    T12HIADD,T12DATA+2 ;SETUP HIGH ORDER MESSAGE BUFFER ADD.
4648 026556 004737 017334      JSR    PC,KTOFF          ;TURN OFF KT-11
4649 026562 010465 000000      MOV    R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
4650 026566 004737 016376      JSR    PC,CHKTSSR       ;WAIT FOR SSR TO SET
4651 026572              FORCERROR 32$
4652 026606 103405              BCS    50$                ;BR IF SSR SET IN CHKSSR
4653 026610 010001              MOV    RO,R1            ;SAVE CONTENTS OF TSSR
4654 026612              NEXT.ERRNO
4655 026612 32$:  ERRDF  ERRNO,T12WRTSSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP    C$ERDF
                                .WORD   306
                                .WORD   T12WRTSSR
                                .WORD   PKTSSR
4656 026622 50$:  CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                TRAP    C$CLP1
4657 026624 150$:
4658 026624              FORCEXIT 160$
4659 026634 020327 030056      CMP    R3,#T12TBE       ;DONE ALL TST12BLK TEST PATTERNS?
4660 026640 103002              BHIS   160$              ;BR IF YES
4661 026642 000137 026504      JMP    T122LOOP         ;DO ANOTHER MODULO- 4 ADDRESS
4662 026646 004737 017334      160$: JSR    PC,KTOFF          ;TURN OFF KT11
4663 026652              ENDSUB                  ;////////////////////// END SUBTEST ////////////////////////
                                L10044:
                                TRAP    C$ESUB
4664 026654 005737 002222      TST    FATFLG           ;ANY FATAL ERRORS ?
4665 026660 001402              BEQ    180$              ;BRANCH IF NOT
4666 026662 004737 017242      JSR    PC,CKDROP        ;TRY TO DROP THE UNIT
4667 026666      180$:

```

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

```

; **
; TEST 3: SUBTEST 3:
; SUBTEST DESCRIPTION:
;
; This subtest verifies the controller can fetch a
; Write Characteristics data block from all available
; memory locations.
; Write Characteristics commands are executed with
; characteristic data blocks at various memory addresses.
; The various memory addresses are determined by floating
; a 1 then a 0 through the address bits.
;
; TEST STEPS:
;
; BEGIN
; Write to TSSR to soft initialize
;
; REPEAT FOR SELECTED VALID ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K

```

4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690

TEST 3: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS

```

4691          :          BEGIN
4692          :          Get a valid test address
4693          :          Set the test packet characteristics data pointer to the
4694          :          test address.
4695          :          Store expected characteristic data in test address block
4696          :          Do a WRITE CHARACTERISTIC command
4697          :          END
4698          :          END
4699          :          END
4700          :          END
4701 026666      BGNSUB          ;//////////////// BEGIN SUBTEST //////////////////
          026666          T3.3:          TRAP          C#BSUB
          026666 104402
4702
4703
4704          ;Write to TSSR to soft initialize
4705 026670 004737 016034      JSR          PC,SOFINIT          ;DO SOFT INIT OF CONTROLLER
4706 026674 103405          BCS          20$          ;BR IF SOFT INIT = OK
4707 026676          NEXT.ERRNO
4708 026676 010001          MOV          R0,R1          ;SAVE CONTENTS OF TSSR
4709 026700          ERRDF          ERRNO,SFIERR,SFIMSG          ;DEVICE FATAL ERROR DURING INIT
          026700 104455          TRAP          C#ERDF
          026702 000463          .WORD          307
          026704 003654          .WORD          SFIERR
          026706 012074          .WORD          SFIMSG
4710
4711          ;Get a valid test address
4712 026710 005037 002222      20$:          CLR          FATFLG          ;CLEAR FATAL ERROR FLAG
4713 026714 005037 027720          CLR          T12KT          ;TEST ABOVE 28K SWITCH
4714 026720 012703 027724          MOV          #T12BLK,R3          ;POINT TO TEST PATTERN TABLE
4715 026724          T123LOOP:
4716 026724 005037 003136          CLR          KTENABLE          ;TURN OFF ABOVE 28K TEST FLAG
4717 026730 012301          MOV          (R3)+,R1          ;GET TEST PATTERN ADDRESS
4718 026732 010100          MOV          R1,R0          ;GET ADDRESS ALL "18 BITS"
4719 026734 042700 177774          BIC          #177774,R0          ;LEAVE ONLY A17 AND A16
4720 026740 042701 000003          BIC          #3,R1          ;GET RID OF A17 AND A16
4721 026744 005737 027720          TST          T12KT          ;TEST ABOVE 28K THIS TIME?
4722 026750 001407          BEQ          25$          ;BR IF NO
4723 026752 016300 177776          MOV          -2(R3),R0          ;GET TEST PATTERN AGAIN
4724 026756 042700 177774          BIC          #1<A1716>,R0          ;SAVE 18 BIT ADDRESS ONLY
4725 026762 012737 000001          MOV          #1,KTENABLE          ;TURN ON ABOVE 28K TEST FLAG
4726 026770 004737 030726          25$:          JSR          PC,T12CONVERT          ;CONVERT TEST PATTERN TO TEST ADDRESS
4727 026774 103402          BCS          30$          ;BR IF VALID TEST ADDRESS
4728 026776 000137 027100          JMP          60$          ;GET NEXT TEST PATTERN
4729          ;Set the test packet characteristics data pointer to the test address
4730 027002 012704 027650          30$:          MOV          #T12PACKET,R4          ;GET THE ADDRESS OF COMMAND PACKET
4731 027006 004737 031230          JSR          PC,T12SWRT          ;RESTORE PACKET TO STARTING VALUES
4732 027012 013764 027714          MOV          T12LOADD,PKLOW(R4)          ;STORE CHAR. DATA PTR LOW ADDRESS
4733 027020 013764 027712          MOV          T12HIADD,PKHI(R4)          ;STORE CHAR. DATA PTR HIGH ADDRESS
4734 027026 004737 031340          JSR          PC,T12CHAR          ;STORE EXPECTED DATA IN DATA BLOCK
4735          ;Do a WRITE CHARACTERISTIC command
4736 027032 004737 017334          JSR          PC,KTOFF          ;TURN OFF KT-11
4737 027036 010465 000000          MOV          R4,TSDB(R5)          ;SET THE PACKET ADDRESS TO EXECUTE
4738 027042 004737 016376          JSR          PC,CHKTSSR          ;WAIT FOR SSR TO SET
4739 027046          FORCERROR          32$
4740 027062 103405          BCS          40$          ;BR IF SSR SET IN CHKTSSR
4741 027064 010001          MOV          R0,R1          ;SAVE CONTENTS OF TSSR

```


TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

4792          ;          BEGIN
4793          ;          Get an invalid test address
4794          ;          Set the test packet characteristics data pointer to the
4795          ;          test address.
4796          ;          Do a WRITE CHARACTERISTIC command
4797          ;          If TSSR register NXM bit not set then print error message
4798          ;          END
4799          ;          END
4800          ;          END
4801          ;          END
4802 027164      BGNSUB          ;////////////////// BEGIN SUBTEST ////////////////////
          027164          T3.4:          TRAP      C#BSUB
          027164 104402
4803
4804
4805 027166 005737 003146      TST      T23A          ;26-APR-83 REV B - CHK FOR 23A CPU
4806 027172 001406          BEQ      5$          ;26-APR-83 REV B - BR, IF NOT 23A
4807 027174 023727 002120 007777  CMP      L#HIME,#7777 ;26-APR-83 REV B - CHK FOR > 256KB
4808 027202 103402          BLO      5$          ;26-APR-83 REV B - BR, IF < 256KB
4809 027204 000137 027576      JMP      NOEXTF      ;26-APR-83 REV B - JMP OVER 256KB
4810 027210          5$:
4811 027210 005737 003140      TST      NXMFLG      ;GOT ENOUGH MEMORY?
4812 027214 001002          BNE      10$         ;IF SET STAY
4813 027216 000137 027576      JMP      NOEXTF      ;LEAVE IF NOT SET
4814
4815          ;Write to TSSR to soft initialize
4816
4817 027222 004737 016034      10$:      JSR      PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
4818 027226 103405          BCS      11$         ;BR IF SOFT INIT = OK
4819 027230          NEXT.ERRNO
4820 027230 010001          MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4821 027232          ERRDF  ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
          027232 104455          TRAP      C#ERDF
          027234 000465          .WORD    309
          027236 003654          .WORD    SFIERR
          027240 012074          .WORD    SFIMSG
4822
4823          ;Do a WRITE CHARACTERISTIC command so to invert switch
4824
4825 027242      11$:      CKLOOP          ;LOOP IF SELECTED
          027242 104406          TRAP      C#CLP1
4826 027244 012704 027650      MOV      #T12PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
4827 027250 004737 031230      JSR      PC,T12SWRT ;RESTORE PACKET TO STARTING VALUES
4828 027254 005037 003136      CLR      KTENABLE ;TURN OFF KT-11
4829 027260 010465 000000      MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS
4830 027264 004737 016376      JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
4831 027270          FORCERROR 15$
4832 027304 103405          BCS      17$         ;BR IF SSR SET IN CHKTSSR
4833 027306 010001          MOV      R0,R1      ;SAVE CONTENTS OF TSSR
4834 027310          NEXT.ERRNO
4835 027310      15$:      ERRDF  ERRNO,T12WRTSSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
          027310 104455          TRAP      C#ERDF
          027312 000466          .WORD    310
          027314 030162          .WORD    T12WRTSSR
          027316 012106          .WORD    PKTSSR
4836 027320      17$:      CKLOOP          ;LOOP IF SELECTED
          027320 104406          TRAP      C#CLP1

```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

4837 027322 004737 020504          JSR    PC,INVERT          ;INVERT THE SWITCH
4838
4839                               ;Get an invalid test address
4840
4841 027326 005037 002222          20$:  CLR    FATFLG          ;CLEAR FATAL ERROR FLAG
4842 027332                               25$:
4843 027332 013737 003144 027712          MOV    NXMHI,T12HIADD      ;SAVE TEST ADDRESS HIGH
4844 027340 013737 003142 027714          MOV    NXMLO,T12LOADD      ;SAVE TEST ADDRESS LOW
4845 027346                               T124LOOP:
4846
4847                               ;Set the test packet characteristics data pointer to the
4848                               ; test address.
4849
4850 027346 012704 027650          30$:  MOV    @T12PACKET,R4    ;GET THE ADDRESS OF COMMAND PACKET
4851 027352 004737 031230          JSR    PC,T12SWRT          ;RESTORE PACKET TO STARTING VALUES
4852 027356 013764 027714 000002          MOV    T12LOADD,PKLOW(R4) ;STORE CHAR. DATA PTR LOW ADDRESS
4853 027364 013764 027712 000004          MOV    T12HIADD,PKHI(R4)  ;STORE CHAR. DATA PTR HIGH ADDRESS
4854
4855                               ;Do a WRITE CHARACTERISTIC command
4856 027372 004737 017334          JSR    PC,KTOFF           ;TURN OFF KT-11
4857 027376 010465 000000          MOV    R4,TSDB(R5)       ;SET THE PACKET ADDRESS TO EXECUTE
4858 027402 004737 016310          JSR    PC,WAITF           ;WAIT FOR SSR TO SET
4859 027406                               FORCERROR 32$
4860 027422 103407          BCS    40$                ;BR IF SSR SET IN CHKTSSR
4861 027424 010001          MOV    R0,R1              ;SAVE CONTENTS OF TSSR
4862 027426                               NEXT.ERRNO
4863 027426 104455          32$:  ERDF   ERRNO,T12WRTSSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
4864 027430 000467                               TRAP   C$ERDF
4865 027432 030162                               .WORD 311
4866 027434 012106                               .WORD T12WRTSSR
4867 027436 005237 002222          INC    FATFLG             ;SET FATAL ERROR FLAG
4868 027442 104406          40$:  CKLOOP              ;LOOP ON ERROR, IF FLAG SET
4869 027444                               TRAP   C$CLP1
4870 027444          FORCERROR 45$,NOTSSR      ;BY-PASS SUBTEST IF FATAL ERROR
4871 027454          ESCAPE  SUB              ;BY-PASS SUBTEST IF FATAL ERROR
4872 027454 104410                               TRAP   C$ESCAPE
4873 027456 000124                               .WORD L10046-.
4874
4875                               ;If TSSR register NXM bit not set then print error message
4876 027460          45$:
4877 027460 016501 000002          MOV    TSSR(R5),R1        ;GET TSSR CONTENTS
4878 027464          FORCERROR 52$
4879 027500 032701 004000          BIT    #NXM,R1           ;NXM SET?
4880 027504 001012          BNE    60$                ;BR IF YES
4881 027506          NEXT.ERRNO
4882 027506 013737 027714 002242          52$:  MOV    T12LOADD,ERRLO      ;MEMORY TEST ADDRESS LOW
4883 027514 013737 027712 002240          MOV    T12HIADD,ERRHI     ;MEMORY TEST ADDRESS HIGH
4884 027522          ERRHRD  ERRNO,T12NXM,ADDSSR ;REPORT ADDRESS AND TSSR ERROR
4885 027522 104456                               TRAP   C$ERHRD
4886 027524 000470                               .WORD 312
4887 027526 030617                               .WORD T12NXM
4888 027530 012166                               .WORD ADDSSR
4889
4890 027532          60$:  CKLOOP              ;LOOP ON ERROR, IF FLAG SET
4891 027532 104406                               TRAP   C$CLP1
4892 027534          FORCEXIT 90$
4893 027544 005737 003146          TST    T23A              ;IS IT A 11/23A?

```

F10

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

4882 027550 001012
4883 027552 013700 027712
4884 027556 005200
4885 027560 020027 000077
4886 027564 101004
4887 027566 010037 027712
4888 027572 000137 027346
4889 027576
4890 027576
4891 027576 004737 017334
4892 027602
027602
027602 104403
4893 027604 005737 002222
4894 027610 001402
4895 027612 004737 017242
4896 027616 004737 016516
4897 027622 103002
4898 027624 000137 026036
4899 027630
4900 027630 004737 017334
4901 027634 005037 003152
4902 027640
027640 104432
027642 001540
4903
4904
4905
4906
4907
4908
4910 027650
4912 027650
4913 027650 100004
4914 027652 027660
4915 027654 000000
4916 027656 000010
4917
4918 027660
4919 027660 027672
4920 027662 000000
4921 027664 000016
4922 027666 000000 000000
4923
4924 027672
4925
4926 027712 000000
4927 027714 000000
4928 027716 000000
4929 027720 000000
4930 027722 000000
4931
4932
4933
4934
4935
4936 027724 000001

```

```

BNE 90$
MOV T12HIADD,RO
65$: INC RO
CMP RO,#77
BHI 90$
75$: MOV RO,T12HIADD
JMP T124LOOP
90$:
NOEXTF:
JSR PC,KTOFF
ENDSUB
TST FATFLG
BEQ 100$
JSR PC,CKDROP
100$: JSR PC,TSTLOOP
BCC 105$
JMP T12LOOP
105$: JSR PC,KTOFF
CLR T38FLG
EXIT TST

```

```

;*
;LOCAL STORAGE FOR THIS TEST
;-
T12PACKET: = <..10>E177770
.WORD 100004
.WORD T12DATA
.WORD 0
.WORD 8.
T12DATA:
.WORD T12BFR
.WORD 0
.WORD 14.
.WORD 0,0
T12BFR: .BLKW 8.
T12HIADD: .WORD 0
T12LOADD: .WORD 0
T12PAR6: .WORD 0
T12KT: .WORD 0
T124TST: .WORD 0
;*
;TABLE OF ADDRESSES
;-
T12BLK: .WORD 000001

```

```

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

;TURN OFF KT11
;//////////////////// END SUBTEST //////////////////////
L10046: TRAP C$ESUB

;ANY FATAL ERRORS ?
;BRANCH IF NOT
;TRY TO DROP THE UNIT
;SHOULD WE DO ITERATIONS?
;BR IF NO
;LOOP UNTIL ITERATION COUNT DONE

;TURN OFF MEMORY MANAGEMENT
;CLEAR TEST FLAG
;ALL DONE THIS TEST
TRAP C$EXIT
.WORD L10042-.

;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND, WITH ACK
;ADDRESS OF CHARACTERISTICS BLOCK

;STARTING VALUE OF BLOCK SIZE

;CHARACTERISTICS DATA BLOCK
;LOW ADDRESS OF MESSAGE BUFFER
;HIGH ORDER OF MESSAGE BUFFER
;LENGTH OF MESSAGE BUFFER

;MESSAGE BUFFER

;HIGH ADDRESS
;LOW ADDRESS
;ADDRESS IN PAR FORMAT
;TEST ABOVE 28K SWITCH
;ADDRESS TEST BIT

```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

4937	027726	000002	.WORD	000002
4938	027730	000003	.WORD	000003
4939	027732	000005	.WORD	000005
4940	027734	000006	.WORD	000006
4941	027736	000007	.WORD	000007
4942	027740	000011	.WORD	000011
4943	027742	000012	.WORD	000012
4944	027744	000013	.WORD	000013
4945	027746	000021	.WORD	000021
4946	027750	000022	.WORD	000022
4947	027752	000023	.WORD	000023
4948	027754	000041	.WORD	000041
4949	027756	000042	.WORD	000042
4950	027760	000043	.WORD	000043
4951	027762	000101	.WORD	000101
4952	027764	000102	.WORD	000102
4953	027766	000103	.WORD	000103
4954	027770	000201	.WORD	000201
4955	027772	000202	.WORD	000202
4956	027774	000203	.WORD	000203
4957	027776	000401	.WORD	000401
4958	030000	000402	.WORD	000402
4959	030002	000403	.WORD	000403
4960	030004	001001	.WORD	001001
4961	030006	001002	.WORD	001002
4962	030010	001003	.WORD	001003
4963	030012	002001	.WORD	002001
4964	030014	002002	.WORD	002002
4965	030016	002003	.WORD	002003
4965	030020	004001	.WORD	004001
4967	030022	004002	.WORD	004002
4968	030024	004003	.WORD	004003
4969	030026	010001	.WORD	010001
4970	030030	010002	.WORD	010002
4971	030032	010003	.WORD	010003
4972	030034	020001	.WORD	020001
4973	030036	020002	.WORD	020002
4974	030040	020003	.WORD	020003
4975	030042	040001	.WORD	040001
4976	030044	040002	.WORD	040002
4977	030046	040003	.WORD	040003
4978	030050	100001	.WORD	100001
4979	030052	100002	.WORD	100002
4980	030054	100003	.WORD	100003
4981	030056	177777	T12TBE: .WORD	177777

```

;+
;LOCAL TEXT MESSAGES FOR TEST
;-

```

4982									
4983									
4984									
4985									
4986	030060	104	115	101	TST12ID:	.ASCIZ	'DMA Memory Addressing'		
4987	030106	103	157	156	T12GETSSR:	.ASCIZ	'Contents of TSSR Incorrect After GET STATUS'		
4988	030162	103	157	156	T12WRTSSR:	.ASCIZ	'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'		
4989	030251	115	145	163	T12MSGBUF:	.ASCIZ	'Message Buffer Contents Incorrect After WRITE CHARACTERISTICS'		
4990	030347	102	141	143	T12BKGND:	.ASCIZ	'Background Pattern Disturbed By WRITE CHARACTERISTICS'		
4991	030435	105	170	160	T12NINT:	.ASCIZ	'Expected Interrupt Not Received On WRITE CHARACTERISTICS'		
4992	030526	127	162	151	T12DPR:	.ASCIZ	'Write Characteristic data in ram does not match expected'		
4993	030617	124	123	123	T12NXM:	.ASCIZ	'TSSR NXM bit failed to set when non-existent memory address specifi		

ed'

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

4994 .EVEN
4995
4996
4997
4998
4999
5000
5001
5002
5003
5004
5005
5006
5007
5008
5009
5010
5011
5012
5013
5014
5015
5016 030726
5017 030726
5018 030732 005037 027714
5019 030736 005037 027712
5020 030742 005037 027716
5021 030746 042701 170000
5022 030752 010005
5023 030754 004737 017334
5024 030760 013702 003126
5025 030764 062702 000020
5026 030770 060102
5027 030772 042702 000003
5028 030776 013703 003132
5029 031002 162703 000020
5030 031006 010237 027714
5031 031012 010237 027716
5032 031016 020203
5033 031020 101007
5034 031022 020237 003126
5035 031026 103007
5036 031030 005737 003136
5037 031034 001004
5038 031036 000424
5039 031040 162702 000020
5040 031044 000754
5041 031046
5042 031046 005737 003136
5043 031052 001420
5044 031054 005737 003134
5045 031060 001413
5046 031062 004737 017316
5047 031066 010500
5048 031070 010037 027712
5049 031074 010201
5050 031076 004737 017356

```

```

;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:
;SAVE R1-R5 UNTIL NEXT RETURN
;CLEAR LOW ADDRESS
;CLEAR HIGH ADDRESS
;CLEAR PAR6 BIASED ADDRESS
;FORCE TO LOWER 12 BITS OF ADDRESS
;SAVE HIGH ORDER ADDRESS BITS
;SHUTOFF MEMORY MANAGEMENT
;GET FIRST FREE ADDRESS
;IN CASE TEST PATTERN=0
;ADD IN TEST PATTERN
;MAKE IT MODULO-4
;GET LAST FREE ADDRESS
;SAVE AT LEAST 8 WORDS (IN CASE MESSAGE BUFFER)
;SAVE POSSIBLE LOW ADDRESS
;SAVE IT IN PAR6 BIASED TOO
;IS THIS ADDRESS ABOVE FREE SPACE?
;BR IF YES
;IS IT IN FREE SPACE?
;BR IF YES- ITS GOOD
;TESTING ABOVE 28K?
;BR IF YES
;BR IF NOT IN FREE SPACE
;FORCE FIT THE TEST PATTERN
;TRY THIS TEST PATTERN ADDRESS
;TESTING ABOVE 28K?
;BR IF NO
;ANY MEMORY ABOVE 28K?
;BR IF NO
;TURN ON MEMORY MANAGEMENT
;GET HIGH ORDER ADDRESS
;SAVE POSSIBLE HIGH ADDRESS
;GET COMPUTED LOW ORDER ADDRESS
;RETURN PAR6 BIASED ADDRESS IN R0

```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

5051 031102 010037 027716      MOV      R0,T12PAR6      ;COPY PAR6 BIASED ADDRESS
5052 031106 103403              BCS      105$           ;BR IF VALID ADDRESS
5053 031110 000241      90$:     CLC              ;CLR C BIT FOR FAILURE
5054 031112 000401              BR        105$           ;
5055 031114 000261      100$:   SEC              ;SET SUCCESS
5056 031116 000207      105$:   RTS      PC      ;RETURN
5057
5058
5059
5060      ;*
5061      ;ROUTINE TO READ THE FIRST 2 BYTES FROM RAM
5062      ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
5063
5064      ;INPUT:
5065
5066      ;           R4      ADDRESS OF THE COMMAND PACKET
5067      ;           R5      FIRST DEVICE UNIBUS ADDRESS
5068
5069      ;OUTPUT:
5070
5071      ;           CARRY   SET - RAM MATCHES PACKET
5072      ;                  CLR - RAM DOES NOT MATCH PACKET
5073
5074      ;IMPLICIT OUTPUT:
5075
5076      ;           THE TABLE RAMDATA IS FILLED WITH THE
5077      ;           DATA HELD IN RAM.
5078      ;           RAMSIZ  SET TO 2 FOR PRAMPKT ROUTINE
5079
5080      ;SIDE EFFECTS:
5081
5082      ;           THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
5083
5084      ;-
5085      T12CKRAM::
5086      SAVREG
5087      MOV      #RAMDATA,R1      ;SAVE THE GENERAL REGISTERS
5088      MOV      #RMPKTBEG,R2     ;ADDRESS TO SAVE THE RAM DATA
5089      CLR      R3                ;BYTE ADDRESS OF FIRST RAM DATA
5090      JSR      PC,CHKTSSR       ;CLEAR THE ERROR FLAG
5091      MOVB    #0,TSDB(R5)       ;WAIT FOR SSR
5092      JSR      PC,CHKTSSR       ;SET MAINTENANCE MODE
5093      MOV      R2,TSDB(R5)     ;WAIT FOR SSR TO SET
5094      JSR      PC,CHKTSSR       ;SELECT NEXT RAM ADDRESS
5095      MOVB    TSBA(R5),(R1)     ;WAIT FOR SSR TO SET
5096      CMPEQ  (R1),.(R4)        ;READ THE RAM DATA
5097      BEQ     20$              ;COMPARE TO EXPECTED
5098      INC     R3                ;BRANCH IF OK
5099      INC     R2                ;SET ERROR FLAG
5100      CMP     R2,#RMPKTBEG+2   ;ADDRESS OF NEXT RAM LOCATION
5101      BLT     10$              ;DONE 2 BYTES?
5102      TST     R3                ;BR IF NO
5103      BEQ     30$              ;WAS AN ERROR FOUND ?
5104      CLC     CLC              ;BRANCH IF NOT
5105      BR     50$                ;CLEAR CARRY TO SHOW ERROR
5106      SEC     SEC              ;AND EXIT
5107      MOV     #2,RAMSIZ        ;SHOW GOOD COMPARE
                                ;SETUP RAMSIZ

```

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

5108 031226 000207          RTS      PC          ;RETURN
5109
5110
5111
5112
5113
5114
5115 031230          T12SWRT:
5116 031230          SAVREG          ;SAVE THE REGISTERS
5117 031234 012701 027650      MOV      #T12PACKET,R1      ;START OF THE PACKET
5118 031240 012721 100004      MOV      #100004,(R1)+     ;WRITE CHARACTERISTICS WITH ACK
5119 031244 012721 027660      MOV      #T12DATA,(R1)+   ;ADDRESS OF CHAR DATA BLOCK
5120 031250 005021          CLR      (R1)+             ;EXTENDED ADDRESS
5121 031252 012721 000010      MOV      #8,(R1)+        ;SIZE OF DATA BLOCK IN BYTES
5122 031256 012721 027672      MOV      #T12BFR,(R1)+   ;ADDRESS OF MESSAGE BUFFER
5123 031262 005021          CLR      (R1)+             ;
5124 031264 012721 000016      MOV      #14,(R1)+      ;LENGTH OF MESSAGE BUFFER
5125 031270 005021          CLR      (R1)+             ;
5126 031272 005011          CLR      (R1)             ;
5127 031274 000207          RTS      PC          ;RETURN
5128
5129
5130
5131
5132
5133
5134
5135
5136
5137
5138 031276          T12SETGET:
5139 031276          SAVREG          ;SAVE THE REGISTERS
5140 031302 010401          MOV      R4,R1            ;GET LOW ORDER ADDRESS
5141 031304 005737 003136      TST      KTENABLE        ;TESTING ABOVE 28K?
5142 031310 001404          BEQ      10$            ;BR IF NO
5143 031312 010300          MOV      R3,R0          ;GET HIGH ORDER ADDRESS
5144 031314 004737 017356      JSR      PC,SETMAP      ;RETURN ADDRESS BIASED TO PAR6 IN R0
5145 031320 010001          MOV      R0,R1          ;GET ADDRESS
5146 031322 012700 000017      10$:  MOV      #P.GETSTATUS,R0 ;GET STATUS COMMAND CODE NO IE
5147 031326 052700 100000      BIS      #P.ACK,R0      ;SET ACK
5148 031332 010021          MOV      R0,(R1)+      ;STORE GET STATUS IN PACKET
5149 031334 005021          CLR      (R1)+         ;CLEAR UNUSED WORD
5150 031336 000207          RTS      PC          ;RETURN
5151
5152
5153
5154
5155
5156
5157
5158 031340          T12CHAR:
5159 031340          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
5160 031344 012700 027660      MOV      #T12DATA,R0     ;GET T12PACKET DATA POINTER
5161 031350 013701 027714      MOV      T12LOAD,R1     ;ASSUME NOT ABOVE 28K
5162 031354 005737 003136      TST      KTENABLE        ;TESTING ABOVE 28K?
5163 031360 001402          BEQ      10$            ;BR IF NO
5164 031362 013701 027716      MOV      T12PAR6,R1     ;SET TEST ADDRESS ABOVE 28K

```


K10

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```
5165 031366 012021          104:  MOV (R0)·.(R1)· ;STORE DATA WORD 1
5166 031370 012021          MOV (R0)·.(R1)· ;STORE DATA WORD 2
5167 031372 012021          MOV (R0)·.(R1)· ;STORE DATA WORD 3
5168 031374 012021          MOV (R0)·.(R1)· ;STORE DATA WORD 4
5169 031376 012021          MOV (R0)·.(R1)· ;STORE DATA WORD 5
5170 031400 000207          RTS PC ;RETURN
5171
5172 031402          ENDTST
      031402
      031402 104401
```

L10042: TRAP C\$ETST

TEST 3: SUBTEST 4: NXM TO SELECTED INVALID ADDRESSES

```

5174
5175
5176
5177
5178
5179
5180
5181
5182
5183 031404
      031404
5184
5185
5190 031404 005737 002214
5191 031410 001402
5192 031412 005237 003402
5193 031416 012700 034043
5194 031422 004737 016550
5195 031426 012737 000005 002216
5196 031434
5197
5198
5199
5200
5201
5202
5203
5204
5205 031434
      031434
      031434 104402
5206 031436
      031436 012700 000000
      031442 104441
5207 031444 005737 003402
5208 031450 001402
5209 031452 000137 031734
5210 031456 004737 034062
5211 031462 004737 034134
5212 031466 004737 016034
5213 031472 103405
5217 031474 010001
5218 031476
      031476 104455
      031500 000621
      031502 003654
      031504 012074
5219 031506
5220 031506 012704 032760
5221 031512 004737 010662
5222 031516 103405
5226 031520 010001
5227 031522
      031522 104456
      031524 000622
      031526 005060
      031530 012074

      .SBTTL TEST 4: RAM EXERCISER TEST
      ;*
      ; THIS TEST USES THE READ AND WRITE RAM (BOTH SINGLE AND 256
      ; LOCATIONS) SELECT CODES OF THE WRITE SUBSYSTEM MEMORY COMMAND
      ; TO EXERCISE THE CONTROLLER'S RAM MEMORY AND DMA LOGIC
      ;
      ;-
      BGNTST
      T4::
      TST TSTCNT ;CHECK FOR RUN MODE
      BEQ 10$ ;BR, IF NOT ONLY PROGRAM RUN
      INC SKIPT ;SET SKIP SW
      10$: MOV #TST15ID,RO ;ASCII MESSAGE TO IDENTIFY TEST
      JSR PC,TSTSETUP ;DO INITIAL TEST SETUP
      MOV #5,LOOPCNT ;PERFORM 5 ITERATIONS
      T15LOOP:
      ;
      ; TEST 4, SUBTEST 1
      ;
      ; THIS SUBTEST WRITES THE ADDRESS (8 BITS) INTO THE
      ; RAM MEMORY SINGLE WORD (8 BITS) MODE
      ;
      ;-
      BGNSUB ;//////////////////// BEGIN SUBTEST //////////////////////
      T4.1:
      SETPRI #PRI00 ;LOWER PRIORITY TO ALLOW INTERRUPTS
      TRAP C#BSUB
      MOV #PRI00,RO
      TRAP C#SPRI
      TST SKIPT ;SHOULD WE SKIP THIS SUBTEST
      BEQ 10$ ;BR, IF NOW SKIP REQUIRED
      JMP 50$ ;SKIP SUBTEST
      10$: JSR PC,T15REST ;SET COMMAND PACKET
      JSR PC,T15RT2 ;SET UP OTHER COMMAND PACKET
      JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
      BCS 20$ ;BR IF INIT WAS OK
      MOV RO,R1 ;CONTENTS OF TSSR REGISTER
      ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      TRAP C#ERDF
      .WORD 401
      .WORD SFIERR
      .WORD SFIMSG
      20$: MOV #T15PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
      JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
      BCS 23$ ;BR, IF COMMAND ISSUED OK
      MOV RO,R1 ;SAVE CONTENTS OF TSSR
      ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      TRAP C#ERHRD
      .WORD 402
      .WORD WRTMSG
      .WORD SFIMSG

```

TEST 4: RAM EXERCISER TEST

5228	031532	012703	000400		23 \dagger :	MOV	#256.,R3		;STARTING ADDRESS FOR RAM WRITE
5229	031536	112737	000001	033471		MOVB	#1,T15BS1		;SIZE OF TRANSFER
5230	031544	112737	000002	033470		MOVB	#2,T15BS0		;WRITE RAM "COMMAND"
5231	031552				25 \dagger :				
5232	031552	010337	033472			MOV	R3,T15S2		;ADDRESS FOR RAM
5233	031556	012704	033460			MOV	#T15PK2,R4		;WRITE SUBSYS MEM PACKET
5234	031562	110337	033474			MOVB	R3,T15S3		;DATA FOR WRITE (ADDRESS)
5235	031566	010465	000000			MOV	R4,TSDB(R5)		;ISSUE COMMAND
5236	031572	004737	016376			JSR	PC,CHKTSSR		;WAIT FOR SSR
5237	031576	103407				BCS	30 \dagger		;BR, IF NO ERROR
5238	031600	010001				MOV	RO,R1		;ERROR, SAVE TSSR
5242	031602					ERRHRD	ERRNO,T15SSR,PKTSSR		;TSSR NOT CORRECT AFTER WRITE SUB MEM
	031602	104456							TRAP C \dagger ERHRD
	031604	000623							.WORD 403
	031606	033476							.WORD T15SSR
	031610	012106							.WORD PKTSSR
5243	031612					ESCAPE	SUB		;DON'T CONTINUE IF ERROR ON WRITE
	031612	104410							TRAP C \dagger ESCAPE
	031614	000122							.WORD L10050-
5244	031616				30 \dagger :	CKLOOP			;SCOPE LOOP
	031616	104406							TRAP C \dagger CLP1
5245									
5246									
5247	031620	005203				INC	R3		;NEXT ADDRESS
5248	031622	020327	010000			CMP	R3,#10000		;END OF RAM MEMORY CHECK
5249	031626	001351				BNE	25 \dagger		;LOOP TILL ALL RAM WRITTEN
5250	031630	005002				CLR	R2		;CLEAR OUT R2 HIGH BITS
5251	031632	005303				DEC	R3		;SET BACK TO 7777
5252	031634	110337	033474		40 \dagger :	MOVB	R3,T15S3		;GET DATA PATTERN BACK IN SHAPE
5253	031640	010337	033472			MOV	R3,T15S2		;ADDRESS FOR RAM READ
5254	031644	112737	000001	033470		MOVB	#1,T15BS0		;READ RAM COMMAND
5255	031652	010465	000000			MOV	R4,TSDB(R5)		;SEND OUT PACKET ADDRESS TO CONTR.
5256	031656	004737	016376			JSR	PC,CHKTSSR		;WAIT FOR READY, NON-AMBIGUOUS
5257	031662	103405				BCS	43 \dagger		;BR, IF NO PROBLEM
5258	031664	010001				MOV	RO,R1		;SAVE TSSR
5262	031666					ERRDF	ERRNO,T15SSR,PKTSSR		;TSSR NOT CORRECT
	031666	104455							TRAP C \dagger ERDF
	031670	000624							.WORD 404
	031672	033476							.WORD T15SSR
	031674	012106							.WORD PKTSSR
5263	031676				43 \dagger :	CKLOOP			;SCOPE LOOP
	031676	104406							TRAP C \dagger CLP1
5264	031700	013701	033022			MOV	T15BFR+20,R1		;GET RAM READ DATA
5265	031704	010302				MOV	R3,R2		;SET UP FOR COMPARE
5266	031706	120102				CMPB	R1,R2		;CHECK WITH DATA WRITTEN
5267	031710	001404				BEQ	45 \dagger		;BR IF OK, DATA IN = DATA OUT
5271	031712					ERRHRD	ERRNO,T15AM4,EXPBREC		;WRITTEN DATA NOT = TO READ
	031712	104456							TRAP C \dagger ERHRD
	031714	000625							.WORD 405
	031716	033755							.WORD T15AM4
	031720	015542							.WORD EXPBREC
5272	031722				45 \dagger :	CKLOOP			;SCOPE LOOP
	031722	104406							TRAP C \dagger CLP1
5273	031724	005303				DEC	R3		;DROP DATA COUNTER (PATTERN)
5274	031726	020327	000377			CMP	R3,#255.		;AT BOTTOM YET
5275	031732	001340				BNE	40 \dagger		;BR, IF MORE TO CHECK
5276	031734				50 \dagger :	CKLOOP			;SCOPE LOOP

TEST 4: RAM EXERCISER TEST

```

5277 031734 104406          ENDSUB          ;////////////////// END SUBTEST ////////////////////
      031736                                     L10050: TRAP C#CLP1
      031736 104403                                     TRAP C#ESUB
5278
5279
5280 031740          BGNSUB          ;////////////////// BEGIN SUBTEST ////////////////////
      031740                                     T4.2: TRAP C#BSUB
      031740 104402

5281
5282
5283
5284
5285
5286
5287
5288
5289
5290 031742 004737 034062
5291 031746 004737 034134
5292 031752 004737 016034
5293 031756 103405
5297 031760 010001
5298 031762
      031762 104455
      031764 000626
      031766 003654
      031770 012074
5299 031772
5300 031772 012704 032760
5301 031776 004737 010662
5302 032002 103405
5306 032004 010001
5307 032006
      032006 104456
      032010 000627
      032012 005060
      032014 012074
5308 032016
5309 032016 112737 000001 033471
5310 032024 012704 033460
5311 032030 012703 000400
5312 032034 112737 000002 033470
5313 032042 105037 033474
5314 032046 010337 033472
5315 032052 010465 000000
5316 032056 004737 016376
5317 032062 103405
5318 032064 010001
5322 032066
      032066 104456
      032070 000630
      032072 033476
      032074 012106
5323 032076
5324 032076 104406

```

```

;*
;
;TEST 4, SUBTEST 2
;
;
;
;

```

```

THIS SUBTEST WRITES RAM WITH ALL ZEROS
THEN WALKS AN ALL ONES WORD DOWN THROUGH MEMORY

```

```

JSR PC,T15REST ;RESTORE PACKET FOR WRITE CHARA
JSR PC,T15RT2 ;RESTORE PACKET FOR WRT SUB SYS MEM
JSR PC,SOFINIT ;DO INITIALIZE ON CONTROLLER
BCS 20# ;BR IF INIT WAS OK
MOV R0,R1 ;CONTENTS OF TSSR REGISTER
ERRDF ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
                                TRAP C#ERDF
                                .WORD 406
                                .WORD SFIERR
                                .WORD SFIMSG
20#: MOV #T15PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
     JSR PC,WRTCHR ;ISSUE WRITE CHARACTERISTICS
     BCS 25# ;BR, IF COMMAND ISSUED OK
     MOV R0,R1 ;SAVE CONTENTS OF TSSR
     ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP C#ERHRD
                                .WORD 407
                                .WORD WRTMSG
                                .WORD SFIMSG
25#: MOV #1,T15BS1 ;SET SIZE OF TRANSFER 1 BYTE
     MOV #T15PK2,R4 ;SET NEW PACKET ADDRESS
     MOV #256,R3 ;STARTING ADDRESS IN RAM
     MOV #2,T15BS0 ;WRITE RAM COMMAND
     CLRB T15S3 ;SET DATA TO 000
     MOV R3,T15S2 ;ADDRESS TO PACKET DATA AREA
     MOV R4,TSDB(R5) ;SEND OUT PACKET ADDRESS
     JSR PC,CHKTSSR ;WAIT FOR SSR
     BCS 33# ;BR, IF NO PROBLEM
     MOV R0,R1 ;SAVE TSSR
     ERRHRD ERRNO,T15SSR,PKTSSR ;TSSR NOT CORRECT
                                TRAP C#ERHRD
                                .WORD 408
                                .WORD T15SSR
                                .WORD PKTSSR
33#: CKLOOP ;SCOPE LOOP
                                TRAP C#CLP1

```

TEST 4: RAM EXERCISER TEST

```

5325
5326 032100 005203          INC      R3          ;NEXT ADDRESS
5327 032102 020327 010000  CMP      R3,#10000  ;END OF RAM MEMORY CHECK
5328 032106 001357          BNE     30$         ;BR, MORE RAM TO GO
5329 032110 005303          DEC     R3          ;SET BACK TO 7777
5330 032112 005002          CLR     R2          ;SET TO ALL ZEROS
5331 032114 112737 000001 033470  MOVB   #1,T15B50   ;READ RAM COMMAND
5332 032122 010337 033472  MOV     R3,T15S2   ;ADDRESS TO BE READ TO PACKET DATA
5333 032126 010465 000000  MOV     R4,TSDB(R5);SEND OUT PACKET ADDRESS
5334 032132 004737 016376  JSR    PC,CHKTSSR ;WAIT FOR SSR TO SET
5335 032136 103405          BCS    41$         ;BR, IF ALL IS WELL
5336 032140 010001          MOV     R0,R1      ;SAVE TSSR
5340 032142          ERRHRD ERRNO,T15SSR,PKTSSR ;TSSR NOT CORRECT
      032142 104456          TRAP   C$ERHRD
      032144 000631          .WORD 409
      032146 033476          .WORD T15SSR
      032150 012106          .WORD PKTSSR
5341 032152          41$: CKLOOP          ;SCOPE LOOP
      032152 104406          TRAP   C$CLP1
5342 032154 013701 033022  MOV     T15BFR+20,R1 ;PICK UP READ DATA
5343 032160 120102          CMPB   R1,R2      ;BOTH SHOULD BE 00000000 BINARY
5344 032162 001404          BEQ    42$         ;BR, IF DATA IS GOOD
5348 032164          ERRHRD ERRNO,T15AM3,EXPBREC ;CHARACTERISTICS DATA NOT CORRECT
      032164 104456          TRAP   C$ERHRD
      032166 000632          .WORD 410
      032170 033653          .WORD T15AM3
      032172 015542          .WORD EXPBREC
5349 032174          42$: CKLOOP          ;SCOPE LOOPER
      032174 104406          TRAP   C$CLP1
5350 032176 012702 000377          MOV     #000377,R2 ;SET ALL ONES WORD
5351 032202 112737 000002 033470  MOVB   #2,T15B50   ;WRITE RAM COMMAND
5352 032210 112737 000377 033474  MOVB   #000377,T15S3 ;ALL ONES PATTERN
5353 032216 010465 000000  MOV     R4,TSDB(R5);PASS PACKET ADDRESS TO CONTR.
5354 032222 004737 016376  JSR    PC,CHKTSSR ;WAIT FOR SSR
5355 032226 103405          BCS    43$         ;BR, IF OK (NO ERROR)
5356 032230 010001          MOV     R0,R1      ;SAVE TSSR
5360 032232          ERRHRD ERRNO,T15SSR,PKTSSR ;TSSR NOT CORRECT
      032232 104456          TRAP   C$ERHRD
      032234 000633          .WORD 411
      032236 033476          .WORD T15SSR
      032240 012106          .WORD PKTSSR
5361 032242          43$: CKLOOP          ;SCOPE LOOP
      032242 104406          TRAP   C$CLP1
5362 032244 112737 000001 033470  MOVB   #1,T15B50   ;SET UP FOR RAM READ
5363 032252 010465 000000  MOV     R4,TSDB(R5);ISSUE RAM READ
5364 032256 004737 016376  JSR    PC,CHKTSSR ;WAIT FOR SSR TO SET
5365 032262 103405          BCS    44$         ;BR, IF OK (NO ERROR)
5366 032264 010001          MOV     R0,R1      ;SAVE TSSR
5370 032266          ERRDF  ERRNO,T15SSR,PKTSSR ;TSSR NOT CORRECT
      032266 104455          TRAP   C$ERDF
      032270 000634          .WORD 412
      032272 033476          .WORD T15SSR
      032274 012106          .WORD PKTSSR
5371 032276 013701 033022  44$: MOV     T15BFR+20,R1 ;PICK UP REC'D DATA
5372 032302 120102          CMPB   R1,R2      ;CHECK WITH DATA WRITTEN
5373 032304 001404          BEQ    45$         ;BR IF OK, DATA IN = DATA OUT
5377 032306          ERRHRD ERRNO,T15AM2,EXPBREC ;WRITTEN DATA NOT = TO READ

```


TEST 4: RAM EXERCISER TEST

```

5424 032470 103405          BCS      33$          ;BR, IF NO PROBLEM
5425 032472 010001          MOV      R0,R1       ;SAVE TSSR
5429 032474          ERRHRD  ERRNO,T15SSR,PKTSSR ;TSSR NOT CORRECT
      032474 104456          TRAP    C$ERHRD
      032476 000640          .WORD  416
      032500 033476          .WORD  T15SSR
      032502 012106          .WORD  PKTSSR
5430 032504          33$:   CKLOOP          ;SCOPE LOOP
      032504 104406          TRAP    C$CLP1
5431
5432
5433 032506 005203          INC      R3          ;NEXT ADDRESS
5434 032510 020327 010000    CMP      R3,#10000   ;END OF RAM MEMORY CHECK
5435 032514 001357          BNE     30$          ;BR, MORE RAM TO GO
5436 032516 005303          DEC     R3          ;SET BACK TO 7777
5437 032520 112702 000377    MOV     #377,R2     ;SET TO ALL ONES
5438 032524 112737 000001 033470    MOV     #1,T15B50   ;READ RAM COMMAND
5439 032532 010337 033472    MOV     R3,T15S2    ;ADDRESS TO BE READ TO PACKET DATA
5440 032536 010465 000000    MOV     R4,TSDB(R5) ;SEND OUT PACKET ADDRESS
5441 032542 004737 016376    JSR     PC,CHKTSSR  ;WAIT FOR SSR TO SET
5442 032546 103405          BCS     41$          ;BR, IF ALL IS WELL
5443 032550 010001          MOV     R0,R1       ;SAVE TSSR
5447 032552          ERRHRD  ERRNO,T15SSR,PKTSSR ;TSSR NOT CORRECT
      032552 104456          TRAP    C$ERHRD
      032554 000641          .WORD  417
      032556 033476          .WORD  T15SSR
      032560 012106          .WORD  PKTSSR
5448 032562          41$:   CKLOOP          ;SCOPE LOOP
      032562 104406          TRAP    C$CLP1
5449 032564 013701 033022    MOV     T15BFR+20,R1 ;PICK UP READ DATA
5450 032570 120102          CMPB   R1,R2       ;BOTH SHOULD BE 11111111 BINARY
5451 032572 001404          BEQ    42$          ;BR, IF DATA IS GOOD
5455 032574          ERRHRD  ERRNO,T15AM3,EXPBREC ;CHARACTERISTICS DATA NOT CORRECT
      032574 104456          TRAP    C$ERHRD
      032576 000642          .WORD  418
      032600 033653          .WORD  T15AM3
      032602 015542          .WORD  EXPBREC
5456 032604 012702 000377    MOV     #000377,R2  ;SET ALL ONES WORD
5457 032610 012737 000002 033470    MOV     #2,T15B50   ;WRITE RAM COMMAND
5458 032616 112737 000377 033474    MOV     #000377,T15S3 ;ALL ONES PATTERN
5459 032624 010465 000000    MOV     R4,TSDB(R5) ;PASS PACKET ADDRESS TO CONTR.
5460 032630 004737 016376    JSR     PC,CHKTSSR  ;WAIT FOR SSR
5461 032634 103405          BCS     43$          ;BR, IF OK (NO ERROR)
5462 032636 010001          MOV     R0,R1       ;SAVE TSSR
5466 032640          ERRHRD  ERRNO,T15SSR,PKTSSR ;TSSR NOT CORRECT
      032640 104456          TRAP    C$ERHRD
      032642 000643          .WORD  419
      032644 033476          .WORD  T15SSR
      032646 012106          .WORD  PKTSSR
5467 032650          43$:   CKLOOP          ;SCOPE LOOP
      032650 104406          TRAP    C$CLP1
5468 032652 112737 000001 033470    MOV     #1,T15B50   ;SET UP FOR RAM READ
5469 032660 010465 000000    MOV     R4,TSDB(R5) ;ISSUE RAM READ
5470 032664 004737 016376    JSR     PC,CHKTSSR  ;WAIT FOR SSR TO SET
5471 032670 103405          BCS     44$          ;BR, IF OK (NO ERROR)
5472 032672 010001          MOV     R0,R1       ;SAVE TSSR
5476 032674          ERRHRD  ERRNO,T15SSR,PKTSSR ;TSSR NOT CORRECT

```

TEST 4: RAM EXERCISER TEST

```

032674 104456
032676 000644
032700 033476
032702 012106
5477 032704 013701 033022 44$: MOV T15BFR+20,R1 ;PICK UP REC'D DATA
5478 032710 120102 CMPB R1,R2 ;CHECK WITH DATA WRITTEN
5479 032712 001404 BEQ 45$ ;BR IF OK, DATA IN = DATA OUT
5483 032714 104456 ERRHRD ERRNO,T15AM2,EXPBREC ;WRITTEN DATA NOT = TO READ
032714 104456 TRAP C$ERHRD
032716 000645 .WORD 421
032720 033552 .WORD T15SSR
032722 015542 .WORD PKTSSR
5484 032724 104406 45$: CKLOOP ;SCOPE LOOP
032724 104406 TRAP C$ERHRD
5485 032726 005303 DEC R3 ;DROP RAM ADDRESS POINTER
5486 032730 020327 000377 CMP R3,#255. ;AT START YET
5487 032734 001271 BNE 40$ ;BR, IF MORE RAM TO CHECK
5488
5489 032736 50$: ENDSUB ;//////////////////// END SUBTEST //////////////////////
5490 032736 L10052: TRAP C$ESUB
032736 104403
5491 032740 004737 016516 JSR PC,TSTLOOP ;DO WE NEED TO ITERATE TEST ?
5492 032744 103002 BCC 63$ ;BRANCH IF NOT
5493 032746 000137 031434 JMP T15LOOP ;EXECUTE AGAIN
5494 032752 104432 63$: EXIT TST ;ALL DONE THIS TEST
5495 032752 001216 TRAP C$EXIT
032754 001216 .WORD L10047-.

;*
;LOCAL STORAGE FOR THIS TEST
;
5501 032760 T15PACKET: .=<..10>&177770 ;COMMAND PACKET FOR TEST
5503 032760 .WORD 100204 ;WRITE CHARACTERISTICS COMMAND, WITH IE, ACK
5504 032760 100204 .WORD T15DATA ;ADDRESS OF CHARACTERISTICS BLOCK
5505 032762 032770 .WORD 0
5506 032764 000000 .WORD 8. ;STARTING VALUE OF BLOCK SIZE
5507 032766 000010 T15DATA: ;CHARACTERISTICS DATA BLOCK
5508 032770 .WORD T15BFR ;ADDRESS OF MESSAGE BUFFER
5509 032770 033002 .WORD 0
5510 032772 000000 .WORD 256. ;LENGTH OF MESSAGE BUFFER
5511 032774 000400 .WORD 0,0
5512 032776 000000 000000 T15BFR: .BLKW 150. ;MESSAGE BUFFER
5513 033002
5514 ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
5515 ;
5516
5518 033460 T15PK2: .=<..10>&177770 ;WRITE SUB SYS MEM COMMAND, IE AND ACK
5520 C33460 100206 .WORD T15BF2 ;ADDRESS OF SELECT BLOCK DATA
5521 033460 033470 .WORD 0
5522 033462 000000 .WORD 6. ;SIZE OF DATA PACKET
5523 033464 000006
5524 033466
5525
5526 .EVEN
5527 033470 T15BF2:

```


TEST 4: RAM EXERCISER TEST

```

5528 033470      000          T15BS0: .BYTE 0          ;BSELO AREA
5529 033471      000          T15BS1: .BYTE 0          ;BSEL1 AREA
5530 033472    000000        T15S2:  .WORD 0          ;SEL 2 AREA
5531 033474    000000        T15S3:  .WORD 0          ;DATA AREA
5532
5533
5534
5535
5536
5537          ;*
5538          ;LOCAL TEXT MESSAGES FOR TEST
5539          ;-
5540 033476      127      122      111 T15SSR: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted'
5541 033552      127      122      111 T15AM2: .ASCIZ 'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Ones Word Read Back'
5542 033653      127      122      111 T15AM3: .ASCIZ 'WRITE SUBSYSTEM MEMORY COMMAND Failed On All Zeros Word Read Back'
5543 033755      127      122      111 T15AM4: .ASCIZ 'WRITE SUBSYSTEM MEMORY COMMAND Failed On Address Test'
5544 034043      122      101      115 TST15ID: .ASCIZ 'RAM Exerciser'
5545          .EVEN
5546
5547          ;*
5548          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
5549          ;WRITE SUBSYSTEM MEMORY COMMAND
5550
5551          ;-
5552
5553 034062          T15REST:
5554 034062          SAVREG          ;SAVE THE REGISTERS
5555 034066    012701    032760      MOV      #T15PACKET,R1      ;START OF THE PACKET
5556 034072    012721    100204      MOV      #100204,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK, IE
5557 034076    012721    032770      MOV      #T15DATA,(R1)+    ;ADDRESS OF CHARAISTICS DATA BLOCK
5558 034102    005021          CLR      (R1)+              ;EXTENDED ADDRESS
5559 034104    012721    000010      MOV      #8,(R1)+          ;SIZE OF DATA BLOCK IN BYTES
5560 034110    012721    033002      MOV      #T15BFR,(R1)+    ;ADDRESS OF MESSAGE BUFFER
5561 034114    005021          CLR      (R1)+
5562 034116    012721    000400      MOV      #256,(R1)+        ;LENGTH OF MESSAGE BUFFER
5563 034122    005021          CLR      (R1)+
5564 034124    005011          CLR      (R1)
5565 034126    005037    033002      CLR      T15BFR            ;CLEAR 1ST LOC IN MESSAGE BUFFER
5566 034132    000207          RTS      PC                ;RETURN
5567
5568
5569 034134          T15RT2:
5570 034134          SAVREG          ;SAVE THE REGISTERS
5571 034140    012701    033460      MOV      #T15PK2,R1       ;START OF THE PACKET
5572 034144    012721    100206      MOV      #100206,(R1)+    ;WRITE SUBSYSTEM MEM. WITH ACK, IE
5573 034150    012721    033470      MOV      #T15BF2,(R1)+   ;ADDRESS OF DATA BLOCK
5574 034154    005021          CLR      (R1)+              ;EXTENDED ADDRESS
5575 034156    012721    000006      MOV      #6,(R1)+         ;SIZE OF DATA BLOCK IN BYTES
5576 034162    005021          CLR      (R1)+
5577 034164    005021          CLR      (R1)+
5578 034166    005011          CLR      (R1)
5579 034170    000207          RTS      PC                ;RETURN
5580 034172          ENDTST
          034172
          034172    104401

```

```

L10047: TRAP C#ETST

```

TEST 5: EXTENDED FEATURES SWITCH AND TIMERS A,B

```

5582          .SBTTL TEST 5: EXTENDED FEATURES SWITCH AND TIMERS A,B
5583
5584          ;**
5585          ; TEST DESCRIPTION:
5586          ;
5587          ; This test verifies the Invert Extended Features function
5588          ; can logically invert the Extended features switch and
5589          ; that the internal timers A and B operate correctly.
5590
5591          ; TEST STEPS:
5592          ;
5593          ; REPEAT FOR LOOPCNT
5594          ; BEGIN
5595          ; Do Subtest 1 - Verify Extended Features Switch
5596          ; Do Subtest 2 - Verify Timers A,B
5597          ; END
5598          ;--
5599
5600 034174          BGNTST
5601 034174
5602 034174          MOV #TST16ID,R0          T5::
5603 012700 036252          JSR PC,TSTSETUP          ;ASCII MESSAGE TO IDENTIFY TEST
5604 034200 004737 016550          MOV #10.,LOOPCNT          ;DO INITIAL TEST SETUP
5605 034204 012737 000012 002216          ;PERFORM 10 ITERATIONS
5606 034212          T16LOOP:
5607
5608          .SBTTL TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST
5609
5610          ;**
5611          ; TEST 5: SUBTEST 1:
5612          ;
5613          ; SUBTEST DESCRIPTION:
5614          ;
5615          ; This subtest verifies that the Invert Sense of Extended features
5616          ; Switch function (Write Subsystem Memory,Write Misc command)
5617          ; operates properly.
5618          ; First the state of the Extended Features switch is read in the
5619          ; message packet supplied by the write characteristics command.
5620          ; Then, the sense of the switch is logically inverted.
5621          ; A Write characteristics command is executed and it is verified
5622          ; that the Extended status register (XST4) is returned when
5623          ; in Extended mode, and not returned if not in extended mode.
5624          ; The subtest also verifies that specifying a Message Buffer
5625          ; address with any of bits 21-19 ,set will cause the command to
5626          ; be rejected.
5627
5628          ; TEST STEPS:
5629          ;
5630          ; BEGIN
5631          ; Write to TSSR register to soft initialize the controller
5632          ; Do WRITE CHARACTERISTICS to check for Extended Features Switch
5633          ; IF Extended Features Hardware Switch CLEAR
5634          ; THEN
5635          ; (* Verify Extended Features switch can be Inverted to SET *)
5636          ; Do Write Subsystem Write Miscellaneous to SET Extended Features.
5637          ; DO a WRITE CHARACTERISTICS with an extended characteristic word
5638          ; Compare the controller ram to the extended characteristic word
5639
5640
5641

```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```

5642      ;           If Data word in controller ram NOT= to word sent Then Print Error
5643      ;           If Message Buffer Data Length NOT= 12. Then Print Error
5644      ;           ELSE
5645      ;           (* Verify Extended Features switch can be Inverted to CLEAR *)
5646      ;           Do Write Subsystem Write Miscellaneous to CLEAR Extended Features.
5647      ;           Do a WRITE CHARACTERISTICS without an extended characteristic word
5648      ;           If Message Buffer Data Length NOT= 10. Then Print Error
5649      ;           END-IF
5650      ;           (* Verify Function Reject when Message Buffer 21-19 are non-zero *)
5651      ;           Write to TSSR register to soft initialize the controller
5652      ;           REPEAT FOR MESSAGE BUFFER ADDRESS bits <21:19> FROM 0 TO 7
5653      ;           DO a WRITE CHARACTERISTICS with a message address bit<21:19> non-zero
5654      ;           If TSSR termination code NOT= Function Reject Then Print Error
5655      ;           END-REPEAT
5656      ;           END
5657      ;           ;--
5658 034212 BGNSUB                               ;////////// BEGIN SUBTEST ////////////
      034212                                     T5.1:      TRAP      C#BSUB
      034212 104402
5659
5660
5661 034214 5#:
5662      ;           Write to TSSR register to soft initialize the controller
5663 034214 004737 016034 JSR      PC,SOFINIT           ;WRITE TO TSSR TO SOFT INITIALIZE
5664 034220 103405 BCS      10#                 ;BR IF SOFT INIT OKAY
5665 034222 010001 MOV      R0,R1               ;SAVE CONTENTS OF TSSR
5666 034224 ERRDF  ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
      034224 104455 TRAP      C#ERDF
      034226 000764 .WORD    500
      034230 003654 .WORD    SFIERR
      034232 012074 .WORD    SFIMSG
5667      ;           Do WRITE CHARACTERISTICS to check for Extended Features Switch
5668 034234 004737 037420 10#: JSR      PC,T16REST           ;RESTORE PACKET DEFAULTS
5669 034240 005037 002222 CLR      FATFLG             ;CLEAR FATAL ERROR FLAG
5670 034244 012704 037600 MOV      #T16PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
5671 034250 004737 010662 JSR      PC,WRTCHR           ;DO WRITE CHARACTERISTICS COMMAND
5672 034254 FORCERROR 12# ;@@DFORCE ERROR IF FORCER=1
5673 034270 103407 BCS      15#                 ;BR IF CARRY SET (GOOD RETURN)
5674 034272 010001 MOV      R0,R1               ;SAVE CONTENTS OF TSSR
5675 034274 NEXT.ERRNO
5676 034274 12#: ERRDF  ERRNO,T16SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      034274 104455 TRAP      C#ERDF
      034276 000765 .WORD    501
      034300 036322 .WORD    T16SSR
      034302 012106 .WORD    PKTSSR
5677 034304 005237 002222 15#: INC      FATFLG             ;SET FATAL ERROR FLAG
5678 034310 CKLOOP ;LOOP ON ERROR, IF FLAG SET
      034310 104406 TRAP      C#CLP1
5679
5680      ;           If Extended Features Hardware Switch Clear then:
5681      ;           (* Verify Extended Features switch can be Inverted to SET *)
5682      ;           REPEAT FOR TEST PATTERNS IN TSTBLK TABLE
5683 034312 012701 037622 MOV      #T16BFR,R1         ;MESSAGE BUFFER ADDRESS
5684 034316 032761 000200 000012 BIT      #X2.EXTF,XST2(R1) ;EXTENDED FEATURES SWITCH CLEAR?
5685 034324 001402 BEQ      20#                 ;BR IF YES
5686 034326 000137 034676 JMP      200#
5687 034332 012703 002766 20#: MOV      #TSTBLK+10.,R3 ;START OF TEST DATA

```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```

5688      ; Do Write Subsystem Write Miscellaneous to SET Extended Features.
5689
5690 034336 004737 037560      JSR    PC,T16SEXT      ;SETUP PACKET FOR WRITE MISC INVERT
5691 034342 012704 037650      MOV    #T16PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
5692 034346 010465 000000      MOV    R4,TSDB(R5)     ;SET THE PACKET ADDRESS TO EXECUTE
5693 034352 004737 016376      JSR    PC,CHKTSSR      ;WAIT FOR SSR TO SET
5694 034356      FORCERROR 32$      ;@@DFORCE ERROR IF FORCER=1
5695 034372 103407      BCS   40$              ;BR IF CARRY SET (GOOD RETURN)
5696 034374 010001      MOV    R0,R1           ;SAVE CONTENTS OF TSSR
5697 034376      NEXT.ERRNO
5698 034376 32$:  ERRDF  ERRNO,T162SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      TRAP    C$ERDF
      .WORD  502
      .WORD  T162SSR
      .WORD  PKTSSR
5699 034406 005237 002222      INC    FATFLG          ;SET FATAL ERROR FLAG
5700 034412 40$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      TRAP    C$CLP1
5701
5702      ; DO a WRITE CHARACTERISTICS with an extended characteristic word
5703 034414 012737 125252 002314  MOV    #125252,DATA     ;SETUP TEST DATA FOR EXTENDED WORD
5704 034422 012704 037600      MOV    #T16PACKET,R4   ;GET THE ADDRESS OF COMMAND PACKET
5705 034426 012764 000020 000006  MOV    #16.,PKBCNT(R4) ;STORE MESSAGE PACKET SIZE
5706 034434 013737 002314 037620  MOV    DATA,T16DATA+10 ;STORE TEST DATA IN EXTENDED WORD
5707 034442 004737 010662      JSR    PC,WRTCHR        ;DO WRITE CHARACTERISTICS COMMAND
5708 034446      FORCERROR 42$      ;@@DFORCE ERROR IF FORCER=1
5709 034462 103407      BCS   50$              ;BR IF CARRY SET (GOOD RETURN)
5710 034464 010001      MOV    R0,R1           ;SAVE CONTENTS OF TSSR
5711 034466      NEXT.ERRNO
5712 034466 42$:  ERRDF  ERRNO,T16SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      TRAP    C$ERDF
      .WORD  503
      .WORD  T16SSR
      .WORD  PKTSSR
5713 034476 005237 002222      INC    FATFLG          ;SET FATAL ERROR FLAG
5714 034502 50$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      TRAP    C$CLP1
5715
5716      ; If the TSBA Address Register NOT= Expected Then Print Error
5716 034504 016501 000000      MOV    TSBA(R5),R1     ;GET TSBA REGISTER CONTENTS
5717 034510 012702 037622      MOV    #T16BFR,R2     ;START OF THE DATA BUFFER
5718 034514 062702 000020 62$:  ADD    #16.,R2         ;EXPECTED CONTENTS OF TSBA
5719 034520      FORCERROR 72$,NOTSSR ;@@DFORCE ERROR IF FORCER=1
5720 034530 020102      CMP    R1,R2           ;COMPARE EXPECTED TO RECEIVED
5721 034532 001404      BEQ   80$              ;ERROR IF NOT EQUAL
5722 034534      NEXT.ERRNO
5723 034534 72$:  ERRHRD  ERRNO,T16TSBA,EXPREC ;PRINT THE ERROR & EXPD/RCV
      TRAP    C$ERHRD
      .WORD  504
      .WORD  T16TSBA
      .WORD  EXPREC
5724 034544 80$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      TRAP    C$CLP1
5725
5726      ; Compare the controller ram to the extended characteristic word
5726      ; If Data word in controller ram NOT= to word sent Then Print Error
5727 034546 012704 037610      MOV    #T16DATA,R4    ;GET CHARACTERISTIC DATA ADDRESS
5728 034552 004737 011264      JSR    PC,CKRAM2       ;DOES RAM DATA EQUAL DATA SENT?
5729 034556      FORCERROR 92$      ;@@DFORCE ERROR IF FORCER=1

```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```

5730 034572 103404          BCS      100$          ;BR IF YES
5731 034574                NEXT.ERRNO
5732 034574          92$:  ERRHRD  ERRNO,PKTRAM,RAMERR  ;REPORT THE RAM ERROR(S)
                                TRAP      C$ERHRD
                                .WORD     505
                                .WORD     PKTRAM
                                .WORD     RAMERR
5733 034604          100$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
5734 034604          104406
;      If Message Buffer Data Length NOT= 12. Then Print Error
5735 034606 012702 037622  MOV      #T16BFR,R2  ;GET MESSAGE BUFFER ADDRESS
5736 034612 016201 000002  MOV      2(R2),R1    ;GET RECV DATA FIELD LENGTH
5737 034616 012702 000014  MOV      #12.,R2    ;GET EXPD DATA FIELD LENGTH
5738 034622          FORCERROR 112$,NOTSSR  ;@@DFORCE ERROR IF FORCER=1
5739 034632 020102          CMP      R1,R2        ;COMPARE EXPECTED TO RECEIVED
5740 034634 001404          BEQ      120$          ;ERROR IF NOT EQUAL
5741 034636          NEXT.ERRNO
5742 034636          112$:  ERRHRD  ERRNO,T16LEN,EXPREC  ;PRINT THE ERROR & EXPD/RECV
                                TRAP      C$ERHRD
                                .WORD     506
                                .WORD     T16LEN
                                .WORD     EXPREC
5743 034646          120$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
5744 034646          104406
5745 034650 004737 016034  JSR      PC,SOFINIT  ;WRITE TO TSSR TO SOFT INITIALIZE
5746 034654 103405          BCS      125$          ;BR IF SOFT INIT OKAY
5747 034656 010001          MOV      R0,R1        ;SAVE CONTENTS OF TSSR
5748 034660          ERRDF   ERRNO,SFIERR,SFIMSG  ;DEVICE FATAL DURING INIT
                                TRAP      C$ERDF
                                .WORD     506
                                .WORD     SFIERR
                                .WORD     SFIMSG
5749 034670          125$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
5750 034672 000137 035056  JMP      300$          ;
5751 034676          ;      (* Verify Extended Features switch can be Inverted to CLEAR *)
5752 034676          200$:  ;
5753 034676          ;      Do Write Subsystem Write Miscellaneous to CLEAR Extended Features.
5754 034676          ;      JSR      PC,T16SEXT  ;SETUP PACKET FOR WRITE MISC INVERT
5755 034676 004737 037560  MOV      #T16PK2,R4  ;GET WRITE SUBSYSTEM COMMAND PACKET
5756 034702 012704 037650  MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
5757 034706 010465 000000  JSR      PC,CHKTSSR  ;WAIT FOR SSR TO SET
5758 034712 004737 016376  FORCERROR 232$      ;@@DFORCE ERROR IF FORCER=1
5759 034716          BCS      240$          ;BR IF CARRY SET (GOOD RETURN)
5760 034732 103407          MOV      R0,R1        ;SAVE CONTENTS OF TSSR
5761 034734 010001          NEXT.ERRNO
5762 034736          232$:  ERRDF   ERRNO,T162SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     507
                                .WORD     T162SSR
                                .WORD     PKTSSR
5763 034736          104455
5764 034746 005237 002222  INC      FATFLG      ;SET FATAL ERROR FLAG
5765 034752          240$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
5766 034752          104406

```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```

5767          ; DO a WRITE CHARACTERISTICS without an extended characteristic word
5768 034754 012704 037600          ; MOV #T16PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
5769 034760 012764 000016 000006 ; MOV #14.,PKBCNT(R4) ;STORE MESSAGE PACKET SIZE
5770 034766 004737 010662          ; JSR PC,WRTCHR ;DO WRITE CHARACTERISTICS COMMAND
5771 034772          ; FORCERROR 242$ ;@DFORCE ERROR IF FORCER=1
5772 035006 103407          ; BCS 250$ ;BR IF CARRY SET (GOOD RETURN)
5773 035010 010001          ; MOV RO,R1 ;SAVE CONTENTS OF TSSR
5774 035012          ; NEXT.ERRNO
5775 035012 242$: ERRDF ERRNO,T16SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
          ; TRAP C$ERDF
          ; .WORD 508
          ; .WORD T16SSR
          ; .WORD PKTSSR
5776 035022 005237 002222          ; INC FATFLG ;SET FATAL ERROR FLAG
5777 035026 250$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
          ; TRAP C$CLP1
5778          ; If Message Buffer Data Length NOT= 10. Then Print Error
5779 035030 013701 037624          ; MOV T16BFR-2,R1 ;GET RECV DATA FIELD LENGTH
5780 035034 012702 000012          ; MOV #10.,R2 ;GET EXPD DATA FIELD LENGTH
5781 035040 020102          ; CMP R1,R2 ;COMPARE EXPECTED TO RECEIVED
5782 035042 001404          ; BEQ 270$ ;ERROR IF NOT EQUAL
5783 035044          ; NEXT.ERRNO
5784 035044 262$: ERRHRD ERRNO,T16LEN,EXPREC ;PRINT THE ERROR & EXPD/RECV
          ; TRAP C$ERHRD
          ; .WORD 509
          ; .WORD T16LEN
          ; .WORD EXPREC
5785 035054 270$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
          ; TRAP C$CLP1
5786
5787
5788          ; (* Verify Function Reject when Message Buffer 21-19 are non-zero *)
5789          ; Write to TSSR register to soft initialize the controller
5790 035056 300$:
5791          ; REPEAT FOR MESSAGE BUFFER ADDRESS bits <21:19> FROM 0 TO 7
5792 035056 012737 000001 002314 320$: MOV #1,DATA ;START AT BITS<21:19>=001
5793          ; DO a WRITE CHARACTERISTICS with a message address bit<21:19> non-zero
5794 035064 325$:
5795 035064 012704 037600          ; MOV #T16PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
5796 035070 012764 000016 000006 ; MOV #14.,PKBCNT(R4) ;STORE MESSAGE PACKET SIZE
5797 035076 013700 002314          ; MOV DATA,RO ;GET TEST DATA
5798          ; .REPT 3
5799          ; ASL RO ;SHIFT INTO BITS 21:19
5800          ; .ENDR
5801 035110 010037 037612          ; MOV RO,T16DATA-2 ;STORE BUFFER ADDRESS BITS 21:19
5802 035114 010465 000000          ; MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
5803 035120 004737 016310          ; JSR PC,WAITF ;WAIT FOR SSR
5804 035124          ; FORCERROR 342$ ;@DFORCE ERROR IF FORCER=1
5805 035140 103407          ; BCS 350$ ;BR IF CARRY SET (GOOD RETURN)
5806 035142 010001          ; MOV RO,R1 ;SAVE CONTENTS OF TSSR
5807 035144          ; NEXT.ERRNO
5808 035144 342$: ERRDF ERRNO,T16SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
          ; TRAP C$ERDF
          ; .WORD 510
          ; .WORD T16SSR
          ; .WORD PKTSSR
5809 035154 005237 002222          ; INC FATFLG ;SET FATAL ERROR FLAG

```

TEST 5: SUBTEST 1: VERIFY EXTENDED FEATURES TEST

```

5810 035160          350$: CKLOOP                ;LOOP ON ERROR, IF FLAG SET
      035160 104406                                TRAP      C#CLP1
5811
5812      ; If TSSR termination code NOT= Function Reject Then Print Error
5813 035162 016501 000002      MOV      TSSR(R5),R1      ;GET RECV TSSR
5814 035166 010102      MOV      R1,R2      ;COPY RECV TSSR
5815 035170 042702 000016      BIC      #TERCLS,R2      ;CLEAR TC<2:0> EXPD
5816 035174 052702 000006      BIS      #TSREJ,R2      ;SET EXPD TC<2:0>= FUNCTION REJECT
5817 035200      FORCERROR 352$,NOTSSR      ;@@DFORCE ERROR IF FORCER=1
5818 035210 020102      CMP      R1,R2      ;EXPD EQUAL RECV?
5819 035212 001404      BEQ     360$      ;BR IF YES
5820 035214      NEXT.ERRNO
5821 035214          352$: ERRHRD  ERRNO,T16REJ,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      035214 104456                                TRAP      C#ERHRD
      035216 000777                                .WORD     511
      035220 036704                                .WORD     T16REJ
      035222 012106                                .WORD     PKTSSR
5822 035224          360$: CKLOOP                ;LOOP ON ERROR, IF FLAG SET
      035224 104406                                TRAP      C#CLP1
5823 035226      FORCEXIT 370$
5824 035236 005237 002314      INC      DATA      ;GET NEXT TST PATTERN
5825 035242 023727 002314 000007      CMP      DATA,#7      ;DONE ALL DATA?
5826 035250 101002      BHI     370$      ;BR IF YES
5827 035252 000137 035064      JMP     325$      ;DO ANOTHER TEST PATTERN
5828      ;
5829 035256          370$:
5830 035256      ENDSUB                                ;//////////////// END SUBTEST //////////////////
      035256                                L10054:
      035256 104403                                TRAP      C#ESUB
5831
5832 035260 005737 002222      TST     FATFLG      ;ANY FATAL ERRORS ?
5833 035264 001402      BEQ     460$      ;BRANCH IF NOT
5834 035266 004737 017242      JSR     PC,CKDROP      ;TRY TO DROP THE UNIT
5835 035272          460$:
5836
5837
5838
5839
5840
5841      .SBTTL TEST 5: SUBTEST 2: VERIFY TIMERS A,B
5842
5843      ;**
5844      ; TEST 5: SUBTEST 2:
5845      ;
5846      ; SUBTEST DESCRIPTION:
5847      ;
5848      ; This subtest verifies that timers A,B can be reset
5849      ; and that Timer A is twice the frequency of Timer B.
5850      ; Timer A has a period of 25 microseconds and Timer B
5851      ; has a period of 50 microseconds. The timers are
5852      ; checked at 1, 28, 53, and 78 microseconds.
5853      ;
5854      ; TEST STEPS:
5855      ;
5856      ; Write to TSSR register to soft initialize the controller
5857      ; Do WRITE CHARACTERISTICS to setup a Message Buffer
5858

```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

5859      ;      (* Verify Timers A,B after RESET TIMER with 0 microsecond delay *)
5860      ;      Do a Write Control RESET TIMER with 1 microsecond delay
5861      ;      Do a Write Subsystem READ STATUS
5862      ;      If Timer A NOT= 0 Then Print Error
5863      ;      If Timer B NOT= 0 Then Print Error
5864      ;      (* Verify Timers A,B after RESET TIMER with 28 microsecond delay *)
5865      ;      Do a Write Control RESET TIMER with 28 microsecond delay
5866      ;      If Timer A NOT= 1 Then Print Error
5867      ;      If Timer B NOT= 1 Then Print Error
5868      ;      Do a Write Control RESET TIMER with 53 microsecond delay
5869      ;      If Timer A NOT= 0 Then Print Error
5870      ;      If Timer B NOT= 1 Then Print Error
5871      ;      Do a Write Control RESET TIMER with 78 microsecond delay
5872      ;      If Timer A NOT= 1 Then Print Error
5873      ;      If Timer B NOT= 0 Then Print Error
5874      ;--
5875      035272      BGNSUB                      ;//////////////// BEGIN SUBTEST //////////////////
          035272      TS.2:
          035272      104402      TRAP      C#BSUB
5876      ;      Write to TSSR register to soft initialize the controller
5877      5$:
5878      035274      004737      016034      JSR      PC,SOFINIT          ;WRITE TO TSSR TO SOFT INITIALIZE
5879      035300      103405      BCS      10$          ;BR IF SOFT INIT OKAY
5880      035302      010001      MOV      R0,R1          ;SAVE CONTENTS OF TSSR
5881      035304      ERRDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
          035304      104455      TRAP      C#ERDF
          035306      000777      .WORD      511
          035310      003654      .WORD      SFIERR
          035312      012074      .WORD      SFIMSG
5882      ;      Do WRITE CHARACTERISTICS to setup a Message Buffer
5883      035314      004737      037420      10$:      JSR      PC,T16REST          ;RESTORE PACKET DEFAULTS
5884      035320      005037      002222      CLR      FATFLG          ;CLEAR FATAL ERROR FLAG
5885      035324      012704      037600      MOV      #T16PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
5886      035330      012764      000010      000006      MOV      #8.,PKBCNT(R4)      ;MESSAGE PACKET SIZE NO EXTEND
5887      035336      004737      010662      JSR      PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
5888      035342      FORCERROR      12$          ;@DFORCE ERROR IF FORCER=1
5889      035356      103407      BCS      15$          ;BR IF CARRY SET (GOOD RETURN)
5890      035360      010001      MOV      R0,R1          ;SAVE CONTENTS OF TSSR
5891      035362      NEXT,ERRNO
5892      035362      12$:      ERRDF      ERRNO,T16SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
          035362      104455      TRAP      C#ERDF
          035364      001000      .WORD      512
          035366      036322      .WORD      T16SSR
          035370      012106      .WORD      PKTSSR
5893      035372      005237      002222      15$:      INC      FATFLG          ;SET FATAL ERROR FLAG
5894      035376      104406      CKLOOP          ;LOOP ON ERROR, IF FLAG SET
          035376      TRAP      C#CLP1
5895      ;
5896      ;      (* Verify Timers A,B after RESET TIMER with 1 microsecond delay *)
5897      ;      Do a Write Control RESET TIMER with 1 microsecond delay
5898      035400      012700      000001      MOV      #MS.FSD,R0          ;RESET TIMER COMMAND
5899      035404      013701      036242      MOV      T16D01,R1          ;1 MICROSECOND DELAY
5900      035410      004737      037532      JSR      PC,T16WMISC          ;SETUP T16PK2 COMMAND PACKET
5901      035414      012704      037650      MOV      #T16PK2,R4          ;GET WRITE SUBSYSTEM COMMAND PACKET
5902      035420      010465      000000      MOV      R4,TSDB(R5)          ;SET THE PACKET ADDRESS TO EXECUTE
5903      035424      004737      016376      JSR      PC,CHKTSSR          ;WAIT FOR SSR TO SET
5904      035430      FORCERROR      32$          ;@DFORCE ERROR IF FORCER=1

```


TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

5905 035444 103407          BCS      40$          ;BR IF CARRY SET (GOOD RETURN)
5906 035446 010001          MOV      R0,R1        ;SAVE CONTENTS OF TSSR
5907 035450                NEXT,ERRNO
5908 035450 32$:          ERRDF  ERRNO,T162SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD    513
                                .WORD    T162SSR
                                .WORD    PKTSSR
5909 035460 005237 002222          INC      FATFLG        ;SET FATAL ERROR FLAG
5910 035464 104406          CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
5911                ; If Timer A NOT= 0 Then Print Error
5912                ; If Timer B NOT= 0 Then Print Error
5913 035466 005002          CLR      R2           ;INIT EXPD
5914 035470 042702 000010          BIC      #S2.ATIM,R2   ;TIMER A EXPD=0
5915 035474 042702 000004          BIC      #S2.BTIM,R2   ;TIMER B EXPD=0
5916 035500 012700 037642          MOV      #T16BFSTA,R0  ;GET RECV READ STATUS
5917 035504 016001 000002          MOV      2(R0),R1     ;GET RECV BYTE 2
5918 035510 042701 177763          BIC      #T<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
5919 035514          FORCERROR 72$,NOTSSR ;@@D
5920 035524 020201          CMP      R2,R1        ;EXPD EQUAL RECV?
5921 035526 001404          BEQ     80$          ;BR IF YES
5922 035530          NEXT,ERRNO
5923 035530 72$:          ERRHRD  ERRNO,T16T01,TIMEXP ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD    514
                                .WORD    T16T01
                                .WORD    TIMEXP
5924 035540 104406          CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
5925                ; Do a Write Control RESET TIMER with 28 microsecond delay
5926                ;
5927 035542 012700 000001          MOV      #MS.RSD,R0    ;RESET TIMER COMMAND
5928 035546 013701 036244          MOV      T16D28,R1    ;28 MICROSECOND DELAY
5929 035552 004737 037532          JSR     PC,T16WMISC   ;SETUP T16PK2 COMMAND PACKET
5930 035556 012704 037650          MOV      #T16PK2,R4   ;GET WRITE SUBSYSTEM COMMAND PACKET
5931 035562 010465 000000          MOV      R4,TSDB(R5)  ;SET THE PACKET ADDRESS TO EXECUTE
5932 035566 004737 016376          JSR     PC,CHKTSSR   ;WAIT FOR SSR TO SET
5933 035572          FORCERROR 112$          ;@@DFORCE ERROR IF FORCER=1
5934 035606 103407          BCS     120$         ;BR IF CARRY SET (GOOD RETURN)
5935 035610 010001          MOV      R0,R1        ;SAVE CONTENTS OF TSSR
5936 035612          NEXT,ERRNO
5937 035612 112$:          ERRDF  ERRNO,T162SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD    515
                                .WORD    T162SSR
                                .WORD    PKTSSR
5938 035622 005237 002222          INC      FATFLG        ;SET FATAL ERROR FLAG
5939 035626 104406          CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
5940                ; If Timer A NOT= 1 Then Print Error
5941                ; If Timer B NOT= 1 Then Print Error
5942 035630 005002          CLR      R2           ;INIT EXPD
5943 035632 052702 000010          BIS     #S2.ATIM,R2   ;TIMER A EXPD=1
5944 035636 052702 000004          BIS     #S2.BTIM,R2   ;TIMER B EXPD=1
5945 035642 012700 037642          MOV      #T16BFSTA,R0  ;GET RECV READ STATUS
5946 035646 016001 000002          MOV      2(R0),R1     ;GET RECV BYTE 2

```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

5947 035652 042701 177763      BIC      #†C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
5948 035656                    FORCERROR 172‡,NOTSSR ;@@D
5949 035666 020201            CMP      R2,R1 ;EXPD EQUAL RECV?
5950 035670 001404            BEQ      180‡ ;BR IF YES
5951 035672                    NEXT.ERRNO
5952 035672 172‡:             ERRHRD  ERRNO,T16T28,TIMEXP ;REPORT ERROR
                                TRAP      C‡ERHRD
                                .WORD    516
035672 104456                    .WORD    T16T28
035674 001004                    .WORD    TIMEXP
035676 037120
035700 015612
5953 035702 180‡:             CKLOOP ;LOOP ON ERROR, IF FLAG SET
035702 104406                    TRAP      C‡CLP1
5954
5955 ; Do a Write Control RESET TIMER with 53 microsecond delay
5956 035704 012700 000001      MOV      #MS.RSD,R0 ;RESET TIMER COMMAND
5957 035710 013701 036246      MOV      T16D53,R1 ;53 MICROSECOND DELAY
5958 035714 004737 037532      JSR      PC,T16WMISC ;SETUP T16PK2 COMMAND PACKET
5959 035720 012704 037650      MOV      #T16PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
5960 035724 010465 000000      MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
5961 035730 004737 016376      JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
5962 035734                    FORCERROR 212‡ ;@@DFORCE ERROR IF FORCER=1
5963 035750 103407            BCS     220‡ ;BR IF CARRY SET (GOOD RETURN)
5964 035752 010001            MOV      R0,R1 ;SAVE CONTENTS OF TSSR
5965 035754                    NEXT.ERRNO
5966 035754 212‡:             ERRDF  ERRNO,T162SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C‡ERDF
                                .WORD    517
                                .WORD    T162SSR
                                .WORD    PKTSSR
035754 104455
035756 001005
035760 036357
035762 012106
5967 035764 005237 002222      INC      FATFLG ;SET FATAL ERROR FLAG
5968 035770 220‡:             CKLOOP ;LOOP ON ERROR, IF FLAG SET
035770 104406                    TRAP      C‡CLP1
5969 ; If Timer A NOT= 0 Then Print Error
5970 ; If Timer B NOT= 1 Then Print Error
5971 035772 005002            CLR      R2 ;INIT EXPD
5972 035774 042702 000010      BIC      #S2.ATIM,R2 ;TIMER A EXPD=0
5973 036000 052702 000004      BIS      #S2.BTIM,R2 ;TIMER B EXPD=1
5974 036004 012700 037642      MOV      #T16BFSTA,R0 ;GET RECV READ STATUS
5975 036010 016001 000002      MOV      2(R0),R1 ;GET RECV BYTE 2
5976 036014 042701 177763      BIC      #†C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
5977 036020                    FORCERROR 272‡,NOTSSR ;@@D
5978 036030 020201            CMP      R2,R1 ;EXPD EQUAL RECV?
5979 036032 001404            BEQ      280‡ ;BR IF YES
5980 036034                    NEXT.ERRNO
5981 036034 272‡:             ERRHRD  ERRNO,T16T53,TIMEXP ;REPORT ERROR
                                TRAP      C‡ERHRD
                                .WORD    518
036034 104456                    .WORD    T16T53
036036 001006                    .WORD    TIMEXP
036040 037220
036042 015612
5982 036044 280‡:             CKLOOP ;LOOP ON ERROR, IF FLAG SET
036044 104406                    TRAP      C‡CLP1
5983 ; Do a Write Control RESET TIMER with 78 microsecond delay
5984 036046 012700 000001      MOV      #MS.RSD,R0 ;RESET TIMER COMMAND
5985 036052 013701 036250      MOV      T16D78,R1 ;78 MICROSECOND DELAY
5986 036056 004737 037532      JSR      PC,T16WMISC ;SETUP T16PK2 COMMAND PACKET
5987 036062 012704 037650      MOV      #T16PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
5988 036066 010465 000000      MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE

```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

5989 036072 004737 016376      JSR      PC,CHKTSSR          ;WAIT FOR SSR TO SET
5990 036076                    FORCERROR 312$             ;@@DFORCE ERROR IF FORCER=1
5991 036112 103407                    BCS      320$             ;BR IF CARRY SET (GOOD RETURN)
5992 036114 010001                    MOV      R0,R1           ;SAVE CONTENTS OF TSSR
5993 036116                    NEXT,ERRNO
5994 036116 312$:  ERRDF  ERRNO,T162SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                    TRAP      C$ERDF
                    .WORD     519
                    .WORD     T162SSR
                    .WORD     PKTSSR
                    036116 104455
                    036120 001007
                    036122 036357
                    036124 012106
5995 036126 005237 002222      INC      FATFLG           ;SET FATAL ERROR FLAG
5996 036132 320$:  CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
                    TRAP      C$CLP1
                    036132 104406
5997 ; If Timer A NOT= 1 Then Print Error
5998 ; If Timer B NOT= 0 Then Print Error
5999 036134 005002      CLR      R2              ;INIT EXPD
6000 036136 052702 000010      BIS      #S2.ATIM,R2     ;TIMER A EXPD=1
6001 036142 042702 000004      BIC      #S2.BTIM,R2     ;TIMER B EXPD=0
6002 036146 012700 037642      MOV      #T16BFSTA,R0    ;GET RECV READ STATUS
6003 036152 01f001 000002      MOV      2(R0),R1        ;GET RECV BYTE 2
6004 036156 042701 177763      BIC      #+C<S2.ATIM!S2.BTIM>,R1 ;SAVE TIMER A:B RECV ONLY
6005 036162                    FORCERROR 372$,NOTSSR     ;@@@
6006 036172 020201      CMP      R2,R1           ;EXPD EQUAL RECV?
6007 036174 001404      BEQ     380$             ;BR IF YES
6008 036176                    NEXT,ERRNO
6009 036176 372$:  ERRHRD  ERRNO,T16T78,TIMEXP ;REPORT ERROR
                    TRAP      C$ERHRD
                    .WORD     520
                    .WORD     T16T78
                    .WORD     TIMEXP
                    036176 104456
                    036200 001010
                    036202 037320
                    036204 015612
6010 036206 380$:  CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
                    TRAP      C$CLP1
                    036206 104406
6011
6012 036210      ENDSUB                    ;////////// END SUBTEST ////////////
                    L10055: TRAP      C$ESUB
                    036210 104403
6013
6014 036212 005737 002222      TST     FATFLG           ;ANY FATAL ERRORS ?
6015 036216 001402      BEQ     460$             ;BRANCH IF NOT
6016 036220 004737 017242      JSR     PC,CKDROP        ;TRY TO DROP THE UNIT
6017 036224 004737 016516      JSR     PC,TSTLOOP       ;SHOULD WE DO ITERATIONS?
6018 036230 103002      BCC     465$             ;BR IF NO
6019 036232 000137 034212      JMP     T16LOOP          ;LOOP UNTIL ITERATIONS DONE
6020 036236 465$:
6021
6022
6023 036236      EXIT      TST                    ;////////// EXIT TEST ////////////
                    TRAP      C$EXIT
                    .WORD     L10053-.
                    036236 104432
                    036240 001524
6024
6025
6026
6027 ;*
        ;LOCAL STORAGE FOR THIS TEST
        ;-
6028
6029 036242 000001      T16D01: .WORD 1          ;1 MICROSECOND DELAY (ACTUALLY .8 MIC)
6030 036244 000040      T16D28: .WORD 40         ;28 MICROSECOND DELAY (.8 MICROS PER)
6031 036246 000076      T16D53: .WORD 76         ;53 MICROSECOND

```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

6032 036250 000142          T16D78:          .WORD 142          ;78 MICROSECOND
6033                          ;*
6034                          ;LOCAL TEXT MESSAGES FOR TEST
6035                          ;-
6036
6037 036252      105      170      164 TST16ID:          .ASCIZ 'Extended Features Switch and Timers A,B'
6038 036322      127      122      111 T16SSR: .ASCIZ 'WRITE CHARACTERISTICS Failed'
6039 036357      127      122      111 T162SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Misc) Failed'
6040 036423      127      122      111 T163SSR: .ASCIZ 'WRITE SUBSYSTEM (Read Status) Failed'
6041 036470      102      165      163 T16TSBA: .ASCIZ 'Bus Address Register (TSBA) Incorrect after Write Characteristics'
6042 036572      104      141      164 T16LEN: .ASCIZ 'Data Field Length in Message Buffer Incorrect after Write Characteristics'
6043 036704      124      123      123 T16REJ: .ASCIZ 'TSSR Function Reject Not Returned When Non-Existent Buffer Address Specifie
d'
6044 037021      124      151      155 T16T01: .ASCIZ 'Timer A,B Incorrect after Reset Timer with 1 microsecond Delay'
6045 037120      124      151      155 T16T28: .ASCIZ 'Timer A,B Incorrect after Reset Timer with 28 microsecond Delay'
6046 037220      124      151      155 T16T53: .ASCIZ 'Timer A,B Incorrect after Reset Timer with 53 microsecond Delay'
6047 037320      124      151      155 T16T78: .ASCIZ 'Timer A,B Incorrect after Reset Timer with 78 microsecond Delay'
6048                          .EVEN
6049
6050                          ;*
6051                          ; SET DEFAULT PACKET
6052                          ;-
6053
6054 037420      012700 037600          T16REST:
6055 037424      012720 100004          MOV          #T16PACKET,R0          ;PACKET ADDRESS
6056 037430      012720 037610          MOV          #100004,(R0)+         ;WRITE CHARACTERISTICS WITH ACK
6057 037434      005020                    MOV          #T16DATA,(R0)+         ;ADDRESS OF CHAR DATA BLOCK
6058 037436      012720 000012          CLR          (R0)+                 ;EXTENDED ADDRESS
6059 037442      012720 037622          MOV          #10,(R0)+             ;SIZE OF MESSAGE PACKET
6060 037446      005020                    MOV          #T16BFR,(R0)+         ;MESSAGE BUFFER ADDRESS
6061 037450      012720 000024          CLR          (R0)+                 ;CLEAR EXTENDED BUFFER ADDRESS
6062 037454      005020                    MOV          #20,(R0)+             ;LENGTH OF MESSAGE BUFFER
6063 037456      005010                    CLR          (R0)+                 ;CLEAR ESS,ENB,EAI,ERI
6064 037460      005037 037622          CLR          (R0)+                 ;CLEAR EXTENDED FEATURES WORD
6065 037464      000207                    CLR          T16BFR                 ;CLEAR 1ST LOCATION IN MESSAGE BUFFER
6066                          RTS          PC
6067
6068                          ;*
6069                          ; CLEAR MESSAGE BUFFER
6070
6071 037466          T16CLRBUF:
6072 037472      012701 037622          SAVREG
6073 037476      012702 000026          MOV          #T16BFR,R1           ;SAVE R1-R5 UNTIL NEXT RETURN
6074 037502      105021                    MOV          #T16BEND-T16BFR,R2   ;GET MESSAGE BUFFER ADDRESS
6075 037504      005302          10$: CLR          (R1)+             ;SIZE OF MESSAGE BUFFER IN BYTES
6076 037506      003375                    DEC          R2                    ;CLEAR A BYTE
6077 037510      000207                    BGT          10$                   ;DONE?
6078                          RTS          PC                    ;BR IF NO
6079                          ;RETURN
6080                          ;*
6081                          ; SETUP T16PK2 PACKET FOR READ STATUS
6082
6083 037512      004737 037466          T16SRD:
6084 037516      012700 037660          JSR          PC,T16CLRBUF          ;CLEAR MESSAGE BUFFER
6085 037522      112720 000005          MOV          #T16DT2,R0           ;WRITE SUBSYSTEM DATA BUFFER
6086 037526      105010                    MOV          #PW.RDSTATUS,(R0)+    ;STORE READ STATUS COMMAND IN BSEL0
6087 037530      000207                    CLR          (R0)+                 ;CLEAR BSEL1
6088                          RTS          PC                    ;RETURN

```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

6089
6090
6091      ;*
6092      ; SETUP T16PK2 PACKET FOR WRITE MISC.
6093      ;
6094      ; INPUT:
6095      ;     R0     CONTAINS WRITE MISC FUNCTION CODE (BSEL1)
6096      ;     R1     CONTAINS DELAY (TIMES 800 NS) FOR BSEL2
6097
6098      ;-
6099      T16WMISC:
6100      SAVREG                                ;SAVE R1-R5 UNITL NEXT RETURN
6101      JSR     PC,T16CLRBUF                    ;CLEAR MESSAGE BUFFER
6102      MOV     #T16DT2,R2                      ;WRITE SUBSYSTEM DATA BUFFER
6103      MOVB   #PW.WMISC,(R2)+                 ;STORE WRITE MISCELLANEOUS IN BSEL0
6104      MOVB   R0,(R2)+                         ;STORE WRITE MISC CODE IN BSEL1
6105      MOVB   R1,(R2)                          ;STOP DELAY (RESET TIMER) IN BSEL2
6106      RTS     PC                              ;RETURN
6107
6108      ;*
6109      ; SETUP T16PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
6110      ;-
6111      T16SEXT:
6112      MOV     #T16DT2,R0                      ;WRITE SUBSYSTEM DATA BUFFER
6113      MOVB   #PW.WMISC,(R0)+                 ;STORE WRITE MISCELLANEOUS IN BSEL0
6114      MOVB   #MS.EXT,(R0)                   ;STORE INVERT EXTENDED FEATURES IN BSEL1
6115      RTS     PC                              ;RETURN
6116
6117
6118      ;
6119      ;.=<..+10>E177770
6120
6121      ;
6122      ;WRITE CHARACTERISTICS COMMAND PACKET
6123      ;
6124      T16PACKET:
6125      .WORD  100004                          ;COMMAND PACKET FOR TEST
6126      .WORD  T16DATA                          ;WRITE CHARACTERISTICS COMMAND, WITH ACK
6127      .WORD  0                                ;ADDRESS OF CHARACTERISTICS BLOCK
6128      .WORD  10.                              ;MESSAGE PACKET SIZE
6129
6130      T16DATA:
6131      .WORD  T16BFR                            ;CHARACTERISTICS DATA BLOCK
6132      .WORD  0                                ;ADDRESS OF MESSAGE BUFFER
6133      .WORD  20.                              ;LENGTH OF MESSAGE BUFFER
6134      .WORD  0                                ;ESS,ENB,EAI,ERI
6135      .WORD  0                                ;EXTENDED FEATURES WORD
6136
6137      ;MESSAGE BUFFER
6138
6139      T16BFR:
6140      .WORD  0                                ;BEGIN MESSAGE BUFFER
6141      .WORD  0                                ;MESSAGE TYPE
6142      .WORD  0                                ;DATA FIELD LENGTH
6143      .WORD  0                                ;RBPCR
6144      .WORD  0                                ;XST0
6145      .WORD  0                                ;XST1
6146      .WORD  0                                ;XST2
6147      .WORD  0                                ;XST3
6148      .WORD  0                                ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM

```

TEST 5: SUBTEST 2: VERIFY TIMERS A,B

```

6148 037642
6149 037650
6150
6151
6152
6156 037650
6157 037650 100006
6158 037652 037660
6159 037654 000000
6160 037656 000012
6161
6162 037660
6163 037660 000
6164 037661 000
6165 037662 000000
6166 037664
6167
6168
6169 037764
037764
037764 104401

T16BFSTA: .BLKB 6.
T16BEND:
;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
;T16PK2:
.WORD P.WRTSUB:P.ACK
.WORD T16DT2
.WORD 0
.WORD 10.
;WRITE SUBSYSTEM WITH ACK
;LOW ADDRESS OF DATA BLOCK
;HIGH ADDRESS OF DATA BLOCK
;MINIMUM MESSAGE PACKET SIZE

T16DT2:
.BYTE 0
.BYTE 0
.WORD 0
.BLKB 64.
;DATA BLOCK
;BSELO
;BSEL1
;SEL2
;WRITE FIFO DATA OUTPUT BUFFER

ENDTST

L10053: TRAP C#ETST

```

TEST 6: FIFO EXERCISER

```

6171                                     .SBTTL TEST 6: FIFO EXERCISER
6172                                     ;**
6173                                     ; TEST DESCRIPTION:
6174                                     ;
6175                                     ;   This test uses the Write Subsystem Memory command to
6176                                     ;   verify the controller's FIFO and associated status and
6177                                     ;   control logic.
6178                                     ;
6179                                     ; TEST STEPS:
6180                                     ;
6181                                     ;   REPEAT FOR LOOPCNT
6182                                     ;   BEGIN
6183                                     ;     Do Subtest 1   - FIFO Initialize status test
6184                                     ;     Do Subtest 2   - FIFO Write Single Byte test
6185                                     ;     Do Subtest 3   - FIFO Write Multiple Bytes test
6186                                     ;     Do Subtest 4   - FIFO Verify ILW Status test
6187                                     ;     Do Subtest 5   - FIFO Input Ready test
6188                                     ;     Do Subtest 6   - FIFO Verify Reset FIFO test
6189                                     ;   END
6190                                     ;--
6191
6192
6193 037766                                BGNTST
6194 037766
6198 037766 012700 046216                MOV    #TST17ID,RO          ;ASCII MESSAGE TO IDENTIFY TEST
6199 037772 004737 016550                JSR    PC,TSTSETUP      ;DO INITIAL TEST SETUP
6200 037776 012737 000012 002216        MOV    #10,,LOOPCNT    ;PERFORM 10 ITERATIONS
6201 040004 004737 017334                JSR    PC,KTOFF        ;SHUT OFF MEMORY MANAGEMENT
6202 040010 005037 003136                CLR    KTENABLE       ;REALLY SHUT DOWN KT-11
6203 040014
6204
6205
6206
6207                                     .SBTTL TEST 6: SUBTEST 1: FIFO INITIALIZE STATUS TEST
6208                                     ;**
6209                                     ; TEST 6: SUBTEST 1:
6210                                     ;
6211                                     ; SUBTEST DESCRIPTION:
6212                                     ;
6213                                     ;   This test verifies, by using the Read Status select code,
6214                                     ;   that the FIFO status is in the correct initial state after
6215                                     ;   the controller is initialized (Input Ready TRUE,
6216                                     ;   Output Ready and Data In Miss FALSE). These status
6217                                     ;   signals are checked by the controller's self-test
6218                                     ;   sequence, so this subtest is actually more of a partial
6219                                     ;   check of the Read Status function than the FIFO status.
6220                                     ;
6221                                     ; TEST STEPS:
6222                                     ;
6223                                     ;   BEGIN
6224                                     ;   Write to TSSR to soft initialize
6225                                     ;   Do a WRITE CHARACTERISTICS to setup a message buffer
6226                                     ;   Do a WRITE SUBSYSTEM Read Status
6227                                     ;   If Input Ready NOT=1 Then Print Error
6228                                     ;   If Output Ready NOT=0 Then Print Error
6229                                     ;   If Data In Miss NOT=0 Then Print Error
6230                                     ;   END

```

TEST 6: SUBTEST 1: FIFO INITIALIZE STATUS TEST

```

6231          ;--
6232 040014   BGNSUB          ;////////// BEGIN SUBTEST //////////
        040014          T6.1: TRAP C#BSUB
        040014 104402
6233
6234          ;
6235 040016   5$: Write to TSSR register to soft initialize the controller
6236 040016 004737 016034   JSR PC,SOFINIT          ;WRITE TO TSSR TO SOFT INITIALIZE
6237 040022 103405         BCS 10$          ;BR IF SOFT INIT OKAY
6238 040024 010001         MOV R0,R1          ;SAVE CONTENTS OF TSSR
6239 040026   ERRDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
        040026 104455          TRAP C#ERDF
        040030 001130          .WORD 600
        040032 003654          .WORD SFIERR
        040034 012074          .WORD SFIMSG
6240          ;
6241 040036 005037 002222   10$: Do a WRITE CHARACTERISTICS to setup a message buffer
6242 040042 012704 047610   CLR FATFLG          ;CLEAR FATAL ERROR FLAG
6243 040046 004737 010662   MOV #T17PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
6244 040052   FORCERROR 42$ JSR PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
6245 040066 103407         BCS 50$          ;BR IF CARRY SET (GOOD RETURN)
6246 040070 010001         MOV R0,R1          ;SAVE CONTENTS OF TSSR
6247 040072   NEXT.ERRNO
6248 040072   42$: ERRDF ERRNO,T17SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
        040072 104455          TRAP C#ERDF
        040074 001131          .WORD 601
        040076 046235          .WORD T17SSR
        040100 012106          .WORD PKTSSR
6249 040102 005237 002222   50$: INC FATFLG          ;SET FATAL ERROR FLAG
6250 040106   CKLOOP          ;LOOP ON ERROR, IF FLAG SET
        040106 104406          TRAP C#CLP1
6251
6252          ;
6253 040110 004737 047374   Do a Write Subsystem READ STATUS
6254 040114 012704 047760   JSR PC,T17SRD          ;SETUP PACKET FOR READ STATUS
6255 040120 010465 000000   MOV #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
6256 040124 004737 016376   MOV R4,TSDB(R5)     ;SET THE PACKET ADDRESS TO EXECUTE
6257 040130   FORCERROR 62$ JSR PC,CHKTSSR          ;WAIT FOR SSR TO SET
6258 040144 103407         BCS 70$          ;BR IF CARRY SET (GOOD RETURN)
6259 040146 010001         MOV R0,R1          ;SAVE CONTENTS OF TSSR
6260 040150   NEXT.ERRNO
6261 040150   62$: ERRDF ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
        040150 104455          TRAP C#ERDF
        040152 001132          .WORD 602
        040154 046336          .WORD T173SSR
        040156 012106          .WORD PKTSSR
6262 040160 005237 002222   70$: INC FATFLG          ;SET FATAL ERROR FLAG
6263 040164   CKLOOP          ;LOOP ON ERROR, IF FLAG SET
        040164 104406          TRAP C#CLP1
6264          ;
6265 040166 004737 047556   Set WORDS 0-7 of expd message buffer = to recv since not testing
6266 040172 012701 046012   JSR PC,T17SETEXP    ;SET WORDS 0-7 EXPD=RECV
6267 040176 012702 047652   MOV #T17EXSTA,R1    ;GET EXPECTED READ STATUS
6268 040182 012221         MOV #T17BFSTA,R2      ;GET RECV READ STATUS
6269 040204 011211         MOV (R2)+,(R1)+     ;SET EXPD WORD #8 = RECV TEMP
6270 040206 052711 000020   MOV (R2),(R1)       ;SET EXPD WORD #9 = RECV TEMP
6271 040212 042711 000040   BIS #S2.INRDY,(R1) ;SET EXP INPUT READY= TRUE
        BIC #S2.OUTRDY,(R1) ;SET EXP OUTPUT READY= FALSE

```


TEST 6: SUBTEST 1: FIFO INITIALIZE STATUS TEST

```

6272 040216 042711 000200      BIC      #S2.DIM,(R1)          ;SET EXP DATA IN MISS = FALSE
6273                          ; If Input Ready NOT=1 then Print Error
6274                          ; If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
6275 040222 005000      CLR      R0                    ;HIGH RECV ADDRESS FOR CKMSG2
6276 040224 012701 047632      MOV      #T17BFR,R1          ;LOW RECV ADDRESS FOR CKMSG2
6277 040230 012702 045772      MOV      #T17EXP,R2         ;EXPD ADDRESS
6278 040234 012703 000024      MOV      #20.,R3           ;NUMBER OF BYTES TO COMPARE
6279 040240 004737 011540      JSR      PC,CKMSG2          ;EXPD EQUAL RECV?
6280 040244      FORCERROR      82$,NOTSSR      ;@@D
6281 040254 103404      BCS      90$                ;BR IF YES
6282 040256      NEXT.ERRNO
6283 040256 82$:      ERRHRD  ERRNO,T171CMP,MSGSTAT ;REPORT ERROR
                          TRAP      C$ERHRD
                          .WORD     603
                          .WORD     T171CMP
6284 040266 90$:      CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
                          TRAP      C$CLP1
6285 040270      ENDSUB
6286 040270      ;////////////////////// END SUBTEST ////////////////////////
                          L10057:
6287 040270 104403      TRAP      C$ESUB
6288 040272 005737 002222      TST      FATFLG             ;ANY FATAL ERRORS ?
6289 040276 001402      BEQ      160$              ;BRANCH IF NOT
6290 040300 004737 017242      JSR      PC,CKDROP         ;TRY TO DROP THE UNIT
6291 040304 160$:

```

```

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

```

```

6292
6293
6294
6295
6296
6297
6298
6299
6300
6301
6302
6303
6304
6305
6306
6307
6308
6309
6310
6311
6312
6313
6314
6315
6316
6317
6318
6319
6320
6321

```

```

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

```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```

6322      : Do a Write Subsystem WRITE NPR to set tape direction out
6323      : Do a Write Subsystem WRITE FIFO with byte count equal to 1
6324      : Do a Write Subsystem READ STATUS
6325      : If Input Ready NOT=1 Then Print Error
6326      : If Output Ready NOT=1 Then Print Error
6327      : If Data In Miss NOT=0 Then Print Error
6328      : Do Write Subsystem READ FIFO with byte count equal to 1
6329      : If Data read from FIFO NOT= to Data sent Then Print Error
6330      : Do a Write Subsystem READ STATUS
6331      : If Input Ready NOT=1 Then Print Error
6332      : If Output Ready NOT=0 Then Print Error
6333      : If Data In Miss NOT=0 Then Print Error
6334      :
6335      : END
6336      : END
6337      :--
040304    BGNSUB                               ;//////////////// BEGIN SUBTEST ///////////
040304    T6.2:                                T6.2:
040304    104402                               TRAP      C#BSUB

6338
6339      : Write to TSSR register to soft initialize the controller
6340      5$:
040306    JSR      PC,SOFINIT                   ;WRITE TO TSSR TO SOFT INITIALIZE
040306    004737   BCS      10$                  ;BR IF SOFT INIT OKAY
040312    103405   MOV      RO,R1                ;SAVE CONTENTS OF TSSR
040314    010001   ERDF    ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
040316    104455   TRAP      C#ERDF
040316    001133   .WORD    603
040320    003654   .WORD    SFIERR
040322    012074   .WORD    SFIMSG

6345      : Do a WRITE CHARACTERISTICS to setup a message buffer
6346      10$:
040326    CLR      FATFLG                      ;CLEAR FATAL ERROR FLAG
040332    012704   MOV      @T17PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
040336    004737   JSR      PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
040342    010662   FORCERROR 42$              ;@DFORCE ERROR IF FORCER=1
040356    103407   BCS      50$                ;BR IF CARRY SET (GOOD RETURN)
040360    010001   MOV      RO,R1                ;SAVE CONTENTS OF TSSR
040362    42$:
040362    ERDF    ERRNO,T17SSR,PKTSSR         ;DEVICE FATAL SSR FAILED TO SET
040362    104455   TRAP      C#ERDF
040364    001134   .WORD    604
040366    046235   .WORD    T17SSR
040370    012106   .WORD    PKTSSR

6354      : INC      FATFLG                      ;SET FATAL ERROR FLAG
6355      50$:
040372    005237   CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
040376    104406   TRAP      C#CLP1

6356      : Do a Write Subsystem READ STATUS
6357      : JSR      PC,T17SRD                   ;SETUP PACKET FOR READ STATUS
6358      : MOV      @T17PK2,R4                 ;GET WRITE SUBSYSTEM COMMAND PACKET
6359      : MOV      R4,TSDB(R5)                ;SET THE PACKET ADDRESS TO EXECUTE
6360      : JSR      PC,CHKTSSR                ;WAIT FOR SSR TO SET
6361      : FORCERROR 62$                      ;@DFORCE ERROR IF FORCER=1
6362      : BCS      70$                        ;BR IF CARRY SET (GOOD RETURN)
6363      : MOV      RO,R1                      ;SAVE CONTENTS OF TSSR
6364      : NEXT.ERRNO
6365      62$:
040440    ERDF    ERRNO,T173SSR,PKTSSR         ;DEVICE FATAL SSR FAILED TO SET
040440    104455   TRAP      C#ERDF
040442    001135   .WORD    605

```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```

040444 046336                                .WORD  T173SSR
040446 012106                                .WORD  PKTSSR
6366 040450 005237 002222                    70$:  INC    FATFLG                ;SET FATAL ERROR FLAG
6367 040454 104406                            CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                           TRAP    C$CLP1
6368 ; Set WORDS 0-7 of expd message buffer = to recv since not testing
6369 040456 004737 047556                    ; JSR    PC,T17SETEXP          ;SET WORDS 0-7 EXPD=RECV
6370 040462 012701 046012                    ; MOV    #T17EXSTA,R1         ;GET EXPECTED READ STATUS
6371 040466 012702 047652                    ; MOV    #T17BFSTA,R2        ;GET RECV READ STATUS
6372 040472 012221                            ; MOV    (R2)+,(R1)+         ;SET EXPD WORD #8 = RECV TEMP
6373 040474 011211                            ; MOV    (R2),(R1)           ;SET EXPD WORD #9 = RECV TEMP
6374 040476 052711 000020                    ; BIS    #S2.INRDY,(R1)      ;SET EXP INPUT READY= TRUE
6375 040502 042711 000040                    ; BIC    #S2.OUTRDY,(R1)     ;SET EXP OUTPUT READY= FALSE
6376 040506 042711 000200                    ; BIC    #S2.DIM,(R1)       ;SET EXP DATA IN MISS = FALSE
6377 ; If Input Ready NOT=1 then Print Error
6378 ; If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
6379 040512 005000                            ; CLR    R0                  ;HIGH RECV ADDRESS FOR CKMSG2
6380 040514 012701 047632                    ; MOV    #T17BFR,R1         ;LOW RECV ADDRESS FOR CKMSG2
6381 040520 012702 045772                    ; MOV    #T17EXP,R2         ;EXPD ADDRESS
6382 040524 012703 000024                    ; MOV    #20,R3             ;NUMBER OF BYTES TO COMPARE
6383 040530 004737 011540                    ; JSR    PC,CKMSG2          ;EXPD EQUAL RECV?
6384 040534                                ; FORCERROR 82$,NOTSSR      ;@AD
6385 040544 103404                            ; BCS    90$                ;BR IF YES
6386 040546                                ; NEXT.ERRNO
6387 040546 82$: ERRHRD ERRNO,T171CMP,MSGSTAT ;REPORT ERROR
                                           TRAP    C$ERHRD
                                           .WORD  606
                                           .WORD  T171CMP
                                           .WORD  MSGSTAT
6388 040556 104406                            90$:  CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                           TRAP    C$CLP1
6389 ; Repeat for DATA from 0 to 377
6390 ;
6391 040560 012737 000000 002314            100$: MOV    #0,DATA          ;GET FIRST DATA
6392 040566                                ; REPEAT LABEL
6393 ; Do a Write Subsystem WRITE NPR to set tape direction out
6394 040566 012700 000100                    ; MOV    #NP.OUT,R0        ;SET TAPE DIRECTION OUT
6395 040572 004737 047436                    ; JSR    PC,T17SNPR        ;SETUP T17PK2 FOR WRITE NPR
6396 040576 012704 047760                    ; MOV    #T17PK2,R4        ;GET WRITE SUBSYSTEM COMMAND PACKET
6397 040602 010465 000000                    ; MOV    R4,TSDB(R5)       ;SET THE PACKET ADDRESS TO EXECUTE
6398 040606 004737 016376                    ; JSR    PC,CHKTSSR        ;WAIT FOR SSR TO SET
6399 040612                                ; FORCERROR 102$           ;@ADFORCE ERROR IF FORCER=1
6400 040626 103407                            ; BCS    105$              ;BR IF CARRY SET (GOOD RETURN)
6401 040630 010001                            ; MOV    R0,R1             ;SAVE CONTENTS OF TSSR
6402 040632                                ; NEXT.ERRNO
6403 040632 102$: ERRDF ERRNO,T174SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                           TRAP    C$ERDF
                                           .WORD  607
                                           .WORD  T174SSR
                                           .WORD  PKTSSR
6404 040642 005237 002222                    105$: INC    FATFLG                ;SET FATAL ERROR FLAG
6405 040646 104406                            CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                           TRAP    C$CLP1
6406 ; Do a Write Subsystem WRITE FIFO with byte count equal to 1
6407 040650 012700 000001                    ; MOV    #1,R0             ;WRITE 1 BYTE
6408 040654 012701 002314                    ; MOV    #DATA,R1          ;FIFO WRITE DATA ADDRESS
6409 040660 004737 047462                    ; JSR    PC,T17WEIF        ;SETUP T17PK2 FOR WRITE FIFO

```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```

6410 040664 012704 047760      MOV      #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
6411 040670 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
6412 040674 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
6413 040700                FORCERROR      107#      ;@@DFORCE ERROR IF FORCER=1
6414 040714 103407                BCS      110#          ;BR IF CARRY SET (GOOD RETURN)
6415 040716 010001                MOV      R0,R1         ;SAVE CONTENTS OF TSSR
6416 040720                NEXT.ERRNO
6417 040720 107#:  ERRDF  ERRNO,T175SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C#ERDF
                                .WORD    608
                                .WORD    T175SSR
                                .WORD    PKTSSR
                                040720 104455
                                040722 001140
                                040724 046446
                                040726 012106
6418 040730 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
6419 040734 104406 110#:  CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
6420
6421      ; Do a Write Subsystem READ STATUS
6422 040736 004737 047374      JSR      PC,T17SRD     ;SETUP PACKET FOR READ STATUS
6423 040742 012704 047760      MOV      #T17PK2,R4    ;GET WRITE SUBSYSTEM COMMAND PACKET
6424 040746 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
6425 040752 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
6426 040756                FORCERROR      112#      ;@@DFORCE ERROR IF FORCER=1
6427 040772 103407                BCS      120#          ;BR IF CARRY SET (GOOD RETURN)
6428 040774 010001                MOV      R0,R1         ;SAVE CONTENTS OF TSSR
6429 040776                NEXT.ERRNO
6430 040776 112#:  ERRDF  ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C#ERDF
                                .WORD    609
                                .WORD    T173SSR
                                .WORD    PKTSSR
                                040776 104455
                                041000 001141
                                041002 046336
                                041004 012106
6431 041006 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
6432 041012 104406 120#:  CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
6433      ; Set WORDS 0-7 of expd message buffer = to recv since not testing
6434 041014 004737 047556      JSR      PC,T17SETEXP   ;SET WORDS 0-7 EXPD=RECV
6435 041020 012701 046012      MOV      #T17EXSTA,R1  ;GET EXPECTED READ STATUS
6436 041024 012702 047652      MOV      #T17BFSTA,R2  ;GET RECV READ STATUS
6437 041030 012221                MOV      (R2),(R1)      ;SET EXPD WORD #8 = RECV TEMP
6438 041032 011211                MOV      (R2),(R1)      ;SET EXPD WORD #9 = RECV TEMP
6439 041034 052711 000020      BIS      #S2.INRDY,(R1) ;SET EXP INPUT READY= 1
6440 041040 052711 000040      BIS      #S2.OTRDY,(R1) ;SET EXP OUTPUT READY= 1
6441 041044 042711 000200      BIC      #S2.DIM,(R1)  ;SET EXP DATA IN MISS = 0
6442
6443      ; If Input Ready NOT=1 then Print Error
6444      ; If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
6444 041050 005000                CLR      R0             ;HIGH RECV ADDRESS FOR CKMSG2
6445 041052 012701 047632      MOV      #T17BFR,R1    ;LOW RECV ADDRESS FOR CKMSG2
6446 041056 012702 045772      MOV      #T17EXP,R2    ;EXPD ADDRESS
6447 041062 012703 000024      MOV      #20,R3        ;NUMBER OF BYTES TO COMPARE
6448 041066 004737 011540      JSR      PC,CKMSG2     ;EXPD EQUAL RECV?
6449 041072                FORCERROR      132#,NOTSSR ;@@D
6450 041102 103404                BCS      140#          ;BR IF YES
6451 041104                NEXT.ERRNO
6452 041104 132#:  ERRHRD  ERRNO,T173CMP,MSGSTAT ;REPORT ERROR
                                TRAP      C#ERHRD
                                .WORD    610
                                .WORD    T173CMP
                                .WORD    MSGSTAT
                                041104 104456
                                041106 001142
                                041110 046733
                                041112 012410

```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```

6453 041114      140$: CKLOOP                ;LOOP ON ERROR, IF FLAG SET
      041114 104406                        TRAP      C#CLP1
6454
6455      ; Do Write Subsystem READ FIFO with byte count equal to 1
6456 041116 012700 000001      MOV      #1,R0                ;SET READ BYTE COUNT
6457 041122 004737 047516      JSR      PC,T17RFIF          ;SETUP T17PK2 FOR READ FIFO
6458 041126 012704 047760      MOV      #T17PK2,R4          ;GET WRITE SUBSYSTEM COMMAND PACKET
6459 041132 010465 000000      MOV      R4,TSDB(R5)         ;SET THE PACKET ADDRESS TO EXECUTE
6460 041136 004737 016376      JSR      PC,CHKTSSR          ;WAIT FOR SSR TO SET
6461 041142                        FORCERROR 142$                ;@@@FORCE ERROR IF FORCER=1
6462 041156 103407      BCS      150$                ;BR IF CARRY SET (GOOD RETURN)
6463 041160 010001      MOV      R0,R1                ;SAVE CONTENTS OF TSSR
6464 041162      NEXT.ERRNO
6465 041162      142$: ERRDF  ERRNO,T176SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
      041162 104455                        TRAP      C#ERDF
      041164 001143                        .WORD    611
      041166 046512                        .WORD    T176SSR
      041170 012106                        .WORD    PKTSSR
6466 041172 005237 002222      INC      FATFLG              ;SET FATAL ERROR FLAG
6467 041176      150$: CKLOOP                ;LOOP ON ERROR, IF FLAG SET
      041176 104406                        TRAP      C#CLP1
6468      ; Set WORDS 0-7 of expd message buffer = to recv since not testing
6469 041200 004737 047556      JSR      PC,T17SETEXP        ;SET WORDS 0-7 EXPD=RCV
6470 041204 012701 046012      MOV      #T17EXSTA,R1        ;GET EXPECTED READ STATUS
6471 041210 012702 047652      MOV      #T17BFSTA,R2        ;GET RCV READ STATUS
6472 041214 013721 002314      MOV      DATA,(R1)          ;SET EXPD WORD #8 = COUNT DATA
6473 041220 011211      MOV      (R2),(R1)           ;SET EXPD WORD #9 = RCV (NOT TESTING)
6474      ; If Data read from FIFO NOT= to Data sent Then Print Error
6475      ; The data is in WORD #8 of the message buffer
6476 041222 005000      CLR      R0                    ;HIGH RCV ADDRESS FOR CKMSG2
6477 041224 012701 047632      MOV      #T17BFR,R1          ;LOW RCV ADDRESS FOR CKMSG2
6478 041230 012702 045772      MOV      #T17EXP,R2          ;EXPD ADDRESS
6479 041234 012703 000022      MOV      #18.,R3             ;NUMBER OF BYTES TO COMPARE
6480 041240 004737 011540      JSR      PC,CKMSG2           ;EXPD EQUAL RCV?
6481 041244      FORCERROR 152$,NOTSSR        ;@@@
6482 041254 103404      BCS      160$                ;BR IF YES
6483 041256      NEXT.ERRNO
6484 041256      152$: ERRHRD  ERRNO,T172CMP,MSGSUB  ;REPORT ERROR
      041256 104456                        TRAP      C#ERHRD
      041260 001144                        .WORD    612
      041262 046637                        .WORD    T172CMP
      041264 014002                        .WORD    MSGSUB
6485 041266      160$: CKLOOP                ;LOOP ON ERROR, IF FLAG SET
      041266 104406                        TRAP      C#CLP1
6486
6487      ; Do a Write Subsystem READ STATUS
6488 041270 004737 047374      JSR      PC,T17SRD           ;SETUP PACKET FOR READ STATUS
6489 041274 012704 047760      MOV      #T17PK2,R4          ;GET WRITE SUBSYSTEM COMMAND PACKET
6490 041300 010465 000000      MOV      R4,TSDB(R5)         ;SET THE PACKET ADDRESS TO EXECUTE
6491 041304 004737 016376      JSR      PC,CHKTSSR          ;WAIT FOR SSR TO SET
6492 041310      FORCERROR 162$                ;@@@FORCE ERROR IF FORCER=1
6493 041324 103407      BCS      170$                ;BR IF CARRY SET (GOOD RETURN)
6494 041326 010001      MOV      R0,R1                ;SAVE CONTENTS OF TSSR
6495 041330      NEXT.ERRNO
6496 041330      162$: ERRDF  ERRNO,T173SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
      041330 104455                        TRAP      C#ERDF
      041332 001145                        .WORD    613

```

TEST 6: SUBTEST 2: FIFO WRITE SINGLE BYTE TEST

```

041334 046336
041336 012106
6497 041340 005237 002222
6498 041344 104406
6499 041344 104406
6500 041346 004737 047556
6501 041352 012701 046012
6502 041356 012702 047652
6503 041362 012221
6504 041364 011211
6505 041366 052711 000020
6506 041372 042711 000040
6507 041376 042711 000200
6508
6509
6510 041402 005000
6511 041404 012701 047632
6512 041410 012702 045772
6513 041414 012703 000024
6514 041420 004737 011540
6515 041424
6516 041434 103404
6517 041436
6518 041436
041436 104456
041440 001146
041442 047017
041444 012410
6519 041446
041446 104406
6520 041450
6521 041460 005237 002314
6522 041464 023727 002314 000377
6523 041472 101002
6524 041474 000137 040566
6525 041500
6526
6527 041500
041500
041500 104403
6528
6529 041502 005737 002222
6530 041506 001402
6531 041510 004737 017242
6532 041514
6533
6534
6535
6536
6537
6538
6539
6540
6541
6542
6543

```

```

170$: INC FATFLG ;SET FATAL ERROR FLAG
CKLOOP ;LOOP ON ERROR, IF FLAG SET
; TRAP C$CLP1
; Set WORDS 0-7 of expd message buffer = to recv since not testing
JSR PC,T17SETEXP ;SET WORDS 0-7 EXPD=RECV
MOV #T17EXSTA,R1 ;GET EXPECTED READ STATUS
MOV #T17BFSTA,R2 ;GET RECV READ STATUS
MOV (R2)+,(R1)+ ;SET EXPD WORD #8 = RECV TEMP
MOV (R2),(R1) ;SET EXPD WORD #9 = RECV TEMP
BIS #S2.INRDY,(R1) ;SET EXP INPUT READY= 1
BIC #S2.OTRDY,(R1) ;SET EXP OUTPUT READY= 0
BIC #S2.DIM,(R1) ;SET EXP DATA IN MISS = 0
; If Input Ready NOT=1 then Print Error
; If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
CLR R0 ;HIGH RECV ADDRESS FOR CKMSG2
MOV #T17BFR,R1 ;LOW RECV ADDRESS FOR CKMSG2
MOV #T17EXP,R2 ;EXPD ADDRESS
MOV #20,R3 ;NUMBER OF BYTES TO COMPARE
JSR PC,CKMSG2 ;EXPD EQUAL RECV?
FORCERROR 172$,NOTSSR ;@@
BCS 180$ ;BR IF YES
NEXT.ERRNO
172$: ERRHRD ERRNO,T174CMP,MSGSTAT ;REPORT ERROR
; TRAP C$ERHRD
; .WORD 614
; .WORD T174CMP
; .WORD MSGSTAT
180$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
; TRAP C$CLP1
FORCEXIT 205$ ;@@
INC DATA ;GET NEXT TEST DATA
CMP DATA,#377 ;DONE 0 TO 377?
BHI 205$ ;BR IF YES
JMP 100$ ;DO ANOTHER TEST PATTERN
205$:
ENDSUB ;////////// END SUBTEST //////////
L10060: TRAP C$ESUB
260$: TST FATFLG ;ANY FATAL ERRORS ?
BEQ 260$ ;BRANCH IF NOT
JSR PC,CKDROP ;TRY TO DROP THE UNIT

```

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

```

; **
; TEST 6: SUBTEST 3:
; SUBTEST DESCRIPTION:
;
; This subtest verifies the ability of the FIFO to correctly
; pass a multiple data bytes from input to output.
; The following sequence is done with various data patterns
;

```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```

6544 ; and byte counts from 2 to 64.
6545 ; 1. Initial FIFO status is checked
6546 ; 2. The Write FIFO function.
6547 ; 3. Read Status is executed and FIFO status is checked.
6548 ; 4. Read FIFO is executed and the data and final status
6549 ; is checked.
6550 ;
6551 ; TEST STEPS:
6552 ;
6553 ; BEGIN
6554 ; Write to TSSR to soft initialize
6555 ; Do a WRITE CHARACTERISTICS to setup a message buffer
6556 ; Do a Write Subsystem READ STATUS
6557 ; If Input Ready NOT=1 Then Print Error
6558 ; If Output Ready NOT=0 Then Print Error
6559 ; If Data In Miss NOT=0 Then Print Error
6560 ; If Last Word NOT=0 Then Print Error
6561 ; REPEAT FOR DATA 0 TO 377, 377 TO 0, FLOATING 1'S,0'S AND ALL 1'S/0'S
6562 ; REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
6563 ; BEGIN
6564 ; Do a Write Subsystem WRITE NPR to set tape direction out
6565 ; Do a Write Subsystem WRITE FIFO
6566 ; Do a Write Subsystem READ STATUS
6567 ; If Input Ready NOT=1 Then Print Error
6568 ; If Output Ready NOT=1 Then Print Error
6569 ; If Data In Miss NOT=0 Then Print Error
6570 ; If Last Word NOT=0 Then Print Error
6571 ; Do Write Subsystem READ FIFO
6572 ; If Data read from FIFO NOT= to Data sent Then Print Error
6573 ; Do a Write Subsystem READ STATUS
6574 ; If Input Ready NOT=1 Then Print Error
6575 ; If Output Ready NOT=0 Then Print Error
6576 ; If Data In Miss NOT=0 Then Print Error
6577 ; If Last Word NOT=0 Then Print Error
6578 ; END
6579 ; END
6580 ;
6581 041514 BGNSUB ;//////////////// BEGIN SUBTEST //////////////////
6582 041514 104402 T6.3: TRAP C#BSUB
6583 ;
6584 041516 ; Write to TSSR register to soft initialize the controller
6585 041516 004737 016034 5$: JSR PC,SOFINIT ;WRITE TO TSSR TO SOFT INITIALIZE
6586 041522 103405 BCS 10$ ;BR IF SOFT INIT OKAY
6587 041524 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
6588 041526 104455 ERRDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
6589 ; TRAP C#ERDF
6590 041536 005037 002222 ;.WORD 614
6591 041542 012704 047610 ;.WORD SFIERR
6592 041546 004737 010662 ;.WORD SFIMSG
6593 041552
6594 041566 103407 10$: Do a WRITE CHARACTERISTICS to setup a message buffer
CLR FATFLG ;CLEAR FATAL ERROR FLAG
MOV #T17PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
JSR PC,WRTCHR ;DO WRITE CHARACTERISTICS COMMAND
FORCERROR 42$ ;@@DFORCE ERROR IF FORCER=1
BCS 50$ ;BR IF CARRY SET (GOOD RETURN)

```


TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```

6637
6638
6639 ; REPEAT FOR BYTE COUNT 2 TO 64 DECIMAL
6640 ; TSTFLAG =1 FOR INCREMENT TEST PATTERN
6641 ; =2 FOR DECREMENT TEST PATTERN
6642 ; =3 FOR TSTBLK TABLE PATTERN
6642 041774 012737 000001 002316 95$: MOV #1,TSTFLAG ;TEST PATTERN FLAG
6643 042002 012737 000002 002312 100$: MOV #2,COUNT ;GET FIRST BYTE COUNT
6644 042002 012737 000002 002312
6645 042010
6646 ; Do a Write Subsystem WRITE NPR to set tape direction out
6647 042010 012700 000100 MOV #NP_OUT,R0 ;SET TAPE DIRECTION OUT
6648 042014 004737 047436 JSR PC,T17SNPR ;SETUP T17PK2 FOR WRITE NPR
6649 042020 012704 047760 MOV #T17PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
6650 042024 010465 000000 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
6651 042030 004737 016376 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
6652 042034 FORCERROR 102$ ;@@DFORCE ERROR IF FORCER=1
6653 042050 103407 BCS 105$ ;BR IF CARRY SET (GOOD RETURN)
6654 042052 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
6655 042054
6656 042054 102$: ERRDF ERRNO,T174SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
        TRAP C#ERDF
        .WORD 618
        .WORD T174SSR
        .WORD PKTSSR
6657 042064 005237 002222 105$: INC FATFLG ;SET FATAL ERROR FLAG
6658 042070 104406 CKLOOP ;LOOP ON ERROR, IF FLAG SET
        TRAP C#CLP1
6659 ; Do a Write Subsystem WRITE FIFO
6660 042072 004737 047536 JSR PC,T17CLEXP ;CLEAR EXPD BUFFER
6661 042076 012701 046114 MOV #T17WFDATA,R1 ;EXPD WRITE FIFO DATA BUFFER
6662 042102 013702 002312 MOV COUNT,R2 ;TEST PATTERN SIZE
6663 042106 022737 000001 002316 CMP #1,TSTFLAG ;INCREMENT PATTERN THIS TIME THRU?
6664 042114 001005 BNE 115$ ;BR IF NO
6665 042116 005000 CLR R0 ;INCREMENT TEST PATTERN
6666 042120 110021 110$: MOVB R0,(R1)+ ;STORE INCREMENT TEST BYTE
6667 042122 005200 INC R0 ;SET NEXT PATTERN
6668 042124 005302 DEC R2 ;DONE?
6669 042126 003374 BGT 110$ ;BR IF NO
6670 042130 022737 000002 002316 115$: CMP #2,TSTFLAG ;DECREMENT PATTERN THIS TIME THRU?
6671 042136 001006 BNE 125$ ;BR IF NO
6672 042140 012700 000377 MOV #377,R0 ;DECREMENT TEST PATTERN
6673 042144 110021 120$: MOVB R0,(R1)+ ;STORE DECREMENT TEST BYTE
6674 042146 005300 DEC R0 ;SET NEXT PATTERN
6675 042150 005302 DEC R2 ;DONE?
6676 042152 003374 BGT 120$ ;BR IF NO
6677 042154 022737 000003 002316 125$: CMP #3,TSTFLAG ;TSTBLK PATTERNS THIS TIME THRU?
6678 042162 001005 BNE 135$ ;BR IF NO
6679 042164 012700 002754 130$: MOV #TSTBLK,R0 ;FLOAT 1'S/O'S ETC. TEST TABLE
6680 042170 112021 130$: MOVB (R0)+,(R1)+ ;STORE A TSTBLK BYTE
6681 042172 005302 DEC R2 ;DONE?
6682 042174 003375 BGT 130$ ;BR IF NO
6683 042176 135$:
6684 042176 013700 002312 MOV COUNT,R0 ;FIFO BYTE COUNT
6685 042202 012701 046114 MOV #T17WFDATA,R1 ;FIFO WRITE DATA ADDRESS
6686 042206 004737 047462 JSR PC,T17WFIF ;SETUP T17PK2 FOR WRITE FIFO
6687 042212 012704 047760 MOV #T17PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
6688 042216 010465 000000 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE

```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```

6689 042222 004737 016376      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
6690 042226                    FORCERROR      142$      ;@DFORCE ERROR IF FORCER=1
6691 042242 103407                    BCS      150$      ;BR IF CARRY SET (GOOD RETURN)
6692 042244 010001                    MOV      R0,R1      ;SAVE CONTENTS OF TSSR
6693 042246                    NEXT,ERRNO
6694 042246 142$:  ERRDF      ERRNO,T175SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      619
                                .WORD      T175SSR
                                .WORD      PKTSSR
6695 042256 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
6696 042262 104406 150$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
6697
6698      ; Do a Write Subsystem READ STATUS
6699 042264 004737 047374      JSR      PC,T17SRD      ;SETUP PACKET FOR READ STATUS
6700 042270 012704 047760      MOV      #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
6701 042274 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
6702 042300 004737 016376      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
6703 042304                    FORCERROR      157$      ;@DFORCE ERROR IF FORCER=1
6704 042320 103407                    BCS      160$      ;BR IF CARRY SET (GOOD RETURN)
6705 042322 010001                    MOV      R0,R1      ;SAVE CONTENTS OF TSSR
6706 042324                    NEXT,ERRNO
6707 042324 157$:  ERRDF      ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD      620
                                .WORD      T173SSR
                                .WORD      PKTSSR
6708 042334 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
6709 042340 104406 160$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
6710
6711      ; Set WORDS 0-7 of expd message buffer = to recv since not testing
6712 042342 004737 047556      JSR      PC,T17SETEXP      ;SET WORDS 0-7 EXPD=RECV
6713 042346 012701 046012      MOV      #T17EXSTA,R1      ;GET EXPECTED READ STATUS
6714 042352 012702 047652      MOV      #T17BFSTA,R2      ;GET RECV READ STATUS
6715 042356 012221                    MOV      (R2), (R1)      ;SET EXPD WORD #8 = RECV TEMP
6716 042360 011211                    MOV      (R2), (R1)      ;SET EXPD WORD #9 = RECV TEMP
6717 042362 052711 000020      BIS      #S2.INRDY,(R1)      ;SET EXP INPUT READY= 1
6718 042366 052711 000040      BIS      #S2.OUTRDY,(R1)      ;SET EXP OUTPUT READY= 1
6719 042372 042711 000200      BIC      #S2.DIM,(R1)      ;SET EXP DATA IN MISS = 0
6720 042376 042711 000100      BIC      #S2.ILW,(R1)      ;SET EXP LAST WORD (ILW)=0
6721
6722      ; If Input Ready NOT=1 then Print Error
6723      ; If Output Ready NOT=1 or Data in Miss NOT=0 Then Print Error
6723 042402 005000                    CLR      R0      ;HIGH RECV ADDRESS FOR CKMSG2
6724 042404 012701 047632      MOV      #T17BFR,R1      ;LOW RECV ADDRESS FOR CKMSG2
6725 042410 012702 045772      MOV      #T17EXP,R2      ;EXPD ADDRESS
6726 042414 012703 000024      MOV      #20,R3      ;NUMBER OF BYTES TO COMPARE
6727 042420 004737 011540      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
6728 042424                    FORCERROR      162$,NOTSSR ;@D
6729 042434 103404                    BCS      170$      ;BR IF YES
6730 042436                    NEXT,ERRNO
6731 042436 162$:  ERRHRD      ERRNO,T173CMP,MSGSTAT ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD      621
                                .WORD      T173CMP
                                .WORD      MSGSTAT
042436 104456
042440 001155
042442 046733
042444 012410

```

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```

6732 042445      170$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      042446 104406      TRAP      C#CLP1
6733
6734      ; Do Write Subsystem READ FIFO
6735 042450 013700 002312      MOV      COUNT,R0      ;SET READ BYTE COUNT
6736 042454 004737 047516      JSR      PC,T17RFIF     ;SETUP T17PK2 FOR READ FIFO
6737 042460 012704 047760      MOV      #T17PK2,R4     ;GET WRITE SUBSYSTEM COMMAND PACKET
6738 042464 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
6739 042470 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
6740 042474      FORCERROR 172$      ;@@@FORCE ERROR IF FORCER=1
6741 042510 103407      BCS      180$           ;BR IF CARRY SET (GOOD RETURN)
6742 042512 010001      MOV      R0,R1         ;SAVE CONTENTS OF TSSR
6743 042514      NEXT.ERRNO
6744 042514      172$: ERRDF  ERRNO,T176SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      042514 104455      TRAP      C#ERDF
      042516 001156      .WORD    622
      042520 046512      .WORD    T176SSR
      042522 012106      .WORD    PKTSSR
6745 042524 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
6746 042530      180$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      042530 104406      TRAP      C#CLP1
6747
6748      ; If Data read from FIFO NOT= to Data sent Then Print Error
6749 042532 005000      CLR      R0            ;HIGH RECV ADDRESS FOR CKMSG2
6750 042534 012702 046114      MOV      #T17WFDATA,R2 ;GET EXPECTED ADDRESS FOR CKMSG2
6751 042540 012701 047652      MOV      #T17BFSTA,R1  ;GET RECEIVED ADDRESS FOR CKMSG2
6752 042544 013703 002312      MOV      COUNT,R3      ;NUMBER OF BYTES TO COMPARE
6753 042550 004737 011540      JSR      PC,CKMSG2     ;EXPD EQUAL RECV?
6754 042554      FORCERROR 192$,NOTSSR ;@@@
6755 042564 103406      BCS      200$           ;BR IF YES
6756 042566      NEXT.ERRNO
6757 042566 013701 002312      192$: MOV      COUNT,R1  ;GET BYTE COUNT
6758 042572      ERRHRD  ERRNO,T175CMP,FIFEXP ;REPORT ERROR
      042572 104456      TRAP      C#ERHRD
      042574 001157      .WORD    623
      042576 047102      .WORD    T175CMP
      042600 012230      .WORD    FIFEXP
6759 042602      200$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      042602 104406      TRAP      C#CLP1
6760
6761      ; Do a Write Subsystem READ STATUS
6762 042604 004737 047374      JSR      PC,T17SRD     ;SETUP PACKET FOR READ STATUS
6763 042610 012704 047760      MOV      #T17PK2,R4     ;GET WRITE SUBSYSTEM COMMAND PACKET
6764 042614 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
6765 042620 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
6766 042624      FORCERROR 212$      ;@@@FORCE ERROR IF FORCER=1
6767 042640 103407      BCS      220$           ;BR IF CARRY SET (GOOD RETURN)
6768 042642 010001      MOV      R0,R1         ;SAVE CONTENTS OF TSSR
6769 042644      NEXT.ERRNO
6770 042644      212$: ERRDF  ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      042644 104455      TRAP      C#ERDF
      042646 001160      .WORD    624
      042650 046336      .WORD    T173SSR
      042652 012106      .WORD    PKTSSR
6771 042654 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
6772 042660      220$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      042660 104406      TRAP      C#CLP1

```

G13

TEST 6: SUBTEST 3: FIFO WRITE MULTIPLE BYTES TEST

```

6773 ; Set WORDS 0-7 of expd message buffer = to rcv since not testing
6774 042662 004737 047556 JSR PC,T17SETEXP ;SET WORDS 0-7 EXPD=RCV
6775 042666 012701 046012 MOV #T17EXSTA,R1 ;GET EXPECTED READ STATUS
6776 042672 012702 047652 MOV #T17BFSTA,R2 ;GET RCV READ STATUS
6777 042676 012221 MOV (R2)+,(R1)+ ;SET EXPD WORD #8 = RCV TEMP
6778 042700 011211 MOV (R2),(R1) ;SET EXPD WORD #9 = RCV TEMP
6779 042702 052711 000020 BIS #S2.INRDY,(R1) ;SET EXP INPUT READY= 1
6780 042706 042711 000040 BIC #S2.OTRDY,(R1) ;SET EXP OUTPUT READY= 0
6781 042712 042711 000200 BIC #S2.DIM,(R1) ;SET EXP DATA IN MISS = 0
6782 042716 042711 000100 BIC #S2.ILW,(R1) ;SET EXP LAST WORD (ILW)=0
6783 ; If Input Ready NOT=1 then Print Error
6784 ; If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
6785 042722 005000 CLR R0 ;HIGH RCV ADDRESS FOR CKMSG2
6786 042724 012701 047632 MOV #T17BFR,R1 ;LOW RCV ADDRESS FOR CKMSG2
6787 042730 012702 045772 MOV #T17EXP,R2 ;EXPD ADDRESS
6788 042734 012703 000024 MOV #20.,R3 ;NUMBER OF BYTES TO COMPARE
6789 042740 004737 011540 JSR PC,CKMSG2 ;EXPD EQUAL RCV?
6790 042744 FORCERROR 232$,NOTSSR ;@@D
6791 042754 103404 BCS 240$ ;BR IF YES
6792 042756 NEXT.ERRNO
6793 042756 232$: ERRHRD ERRNO,T174CMP,MSGSTAT ;REPORT ERROR
;
; TRAP C$ERHRD
; .WORD 625
; .WORD T174CMP
; .WORD MSGSTAT
6794 042766 240$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
; TRAP C$CLP1
6795 042770 FORCEEXIT 250$ ;@@D
6796 043000 005237 002312 INC COUNT ;GET NEXT BYTE COUNT
6797 043004 023727 002312 000077 CMP COUNT,#77 ;DONE 0 TO 77
6798 043012 101002 BHI 250$ ;BR IF YES
6799 043014 000137 042010 JMP 100$ ;DO ANOTHER BYTE COUNT
6800 043020 005237 002316 250$: INC TSTFLAG ;GET NEXT TEST PATTERN CODE
6801 043024 023727 002316 000003 CMP TSTFLAG,#3 ;DONE INC,DEC,TSTBLK PATTERNS?
6802 043032 101002 BHI 255$ ;BR IF YES
6803 043034 000137 042002 JMP 95$ ;DO ANOTHER TEST PATTERN
6804 043040 255$: ENDSUB ;////////////////// END SUBTEST ////////////////////
; L10061: TRAP C$ESUB
6805 043040 104403
6806 043042 005737 002222 TST FATFLG ;ANY FATAL ERRORS ?
6807 043046 001402 BEQ 260$ ;BRANCH IF NOT
6808 043050 004737 017242 JSR PC,CKDROP ;TRY TO DROP THE UNIT
6809 043054 260$:

```

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

```

6811 ;**
6812 ; TEST 6: SUBTEST 4:
6813 ;
6814 ; SUBTEST DESCRIPTION:
6815 ;
6816 ; This subtest verifies that reading the FIFO when it is
6817 ; empty causes the Last Word (ILW) status to assert.
6818 ;
6819 ;
6820 ;
6821 ;
6822 ;

```

TEST 6: SUBTEST 4: FIFO Verify ILW Status

```

6823 ;
6824 ; TEST STEPS:
6825 ;
6826 ; BEGIN
6827 ; Write to TSSR to soft initialize
6828 ; Do Write Subsystem READ FIFO with byte count equal to 1
6829 ; Do a Write Subsystem READ STATUS
6830 ; If Input Ready NOT=1 Then Print Error
6831 ; If Output Ready NOT=0 Then Print Error
6832 ; If Data In Miss NOT=0 Then Print Error
6833 ; If Last Word (ILW) NOT=1 Then Print Error
6834 ;
6835 ; END
6836 ;--
043054 BGNSUB ;//////////////// BEGIN SUBTEST //////////////////
043054 T6.4: TRAP C$BSUB
043054 104402

6837 ;
6838 ; Write to TSSR register to soft initialize the controller
6839 5$: JSR PC,SOFINIT ;WRITE TO TSSR TO SOFT INITIALIZE
6840 043056 004737 016034 BCS 10$ ;BR IF SOFT INIT OKAY
6841 043062 103405 MOV R0,R1 ;SAVE CONTENTS OF TSSR
6842 043064 010001 ERRDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
6843 043066 104455 TRAP C$ERDF
043066 001161 .WORD 625
043070 003654 .WORD SFIERR
043072 012074 .WORD SFIMSG

6844 ;
6845 10$: Do a WRITE CHARACTERISTICS to setup a message buffer
6846 CLR FATFLG ;CLEAR FATAL ERROR FLAG
6847 043102 012704 047610 MOV #T17PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
6848 043106 004737 010662 JSR PC,WRTCHR ;DO WRITE CHARACTERISTICS COMMAND
6849 043112 FORCERROR 42$ ;@DFORCE ERROR IF FORCER=1
6850 043126 103407 BCS 50$ ;BR IF CARRY SET (GOOD RETURN)
6851 043130 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
6852 043132 42$: ERRDF ERRNO,T17SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
043132 104455 TRAP C$ERDF
043134 001162 .WORD 626
043136 046235 .WORD T17SSR
043140 012106 .WORD PKTSSR

6853 043142 005237 002222 INC FATFLG ;SET FATAL ERROR FLAG
6854 043146 50$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
043146 104406 TRAP C$CLP1

6855 ;
6856 ; Do Write Subsystem READ FIFO with byte count equal to 1
6857 MOV #1,R0 ;SET READ BYTE COUNT
6858 043150 012700 000001 JSR PC,T17RFIF ;SETUP T17PK2 FOR READ FIFO
6859 043154 004737 047516 MOV #T17PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
6860 043160 012704 047760 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
6861 043164 010465 000000 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
6862 043174 FORCERROR 142$ ;@DFORCE ERROR IF FORCER=1
6863 043210 103407 BCS 150$ ;BR IF CARRY SET (GOOD RETURN)
6864 043212 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
6865 043214 NEXT.ERRNO
6866 043214 142$: ERRDF ERRNO,T176SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
043214 104455 TRAP C$ERDF
043216 001163 .WORD 627

```

TEST 6: SUBTEST 4: FIFO Verify ILW Status

```

043220 046512
043222 012106
6867 043224 005237 002222
6868 043230
043230 104406
6869
6870
6871 043232 004737 047374
6872 043236 012704 047760
6873 043242 010465 000000
6874 043246 004737 016376
6875 043252
6876 043266 103407
6877 043270 010001
6878 043272
6879 043272
043272 104455
043274 001164
043276 046336
043300 012106
6880 043302 005237 002222
6881 043306
043306 104406
6882
6883 043310 004737 047556
6884 043314 012701 046012
6885 043320 012702 047652
6886 043324 012221
6887 043326 011211
6888 043330 052711 000020
6889 043334 042711 000040
6890 043340 042711 000200
6891 043344 052711 000100
6892
6893
6894
6895 043350 005000
6896 043352 012701 047632
6897 043356 012702 045772
6898 043362 012703 000024
6899 043366 004737 011540
6900 043372
6901 043402 103404
6902 043404
6903 043404
043404 104456
043406 001165
043410 047156
043412 012410
6904 043414
043414 104406
6905
6906 043416
043416
043416 104403
6907
6908 043420 005737 002222

```

```

1504: INC FATFLG ;SET FATAL ERROR FLAG
CKLOOP ;LOOP ON ERROR, IF FLAG SET
TRAP C4CLP1

; Do a Write Subsystem READ STATUS
JSR PC,T17SRD ;SETUP PACKET FOR READ STATUS
MOV #T17PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
FORCERROR 1624 ;@@DFORCE ERROR IF FORCER=1
BCS 1704 ;BR IF CARRY SET (GOOD RETURN)
MOV RO,R1 ;SAVE CONTENTS OF TSSR
NEXT.ERRNO
1624: ERRDF ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
TRAP C4ERDF
.WORD 628
.WORD T173SSR
.WORD PKTSSR

1704: INC FATFLG ;SET FATAL ERROR FLAG
CKLOOP ;LOOP ON ERROR, IF FLAG SET
TRAP C4CLP1

; Set WORDS 0-7 of expd message buffer = to recv since not testing
JSR PC,T17SETEXP ;SET WORDS 0-7 EXPD=RECV
MOV #T17EXSTA,R1 ;GET EXPECTED READ STATUS
MOV #T17BFSTA,R2 ;GET RECV READ STATUS
MOV (R2),R1 ;SET EXPD WORD #8 = RECV TEMP
MOV (R2),R1 ;SET EXPD WORD #9 = RECV TEMP
BIS #S2.INRDY,(R1) ;SET EXP INPUT READY= 1
BIC #S2.OTRDY,(R1) ;SET EXP OUTPUT READY= 0
BIC #S2.DIM,(R1) ;SET EXP DATA IN MISS = 0
BIS #S2.ILW,(R1) ;SET EXP LAST WORD (ILW)=1
; If Input Ready NOT=1 then Print Error
; If Output Ready NOT=0 or Data in Miss NOT=0 Then Print Error
; If Last Word (ILW) NOT=1 Then Print Error
CLR RO ;HIGH RECV ADDRESS FOR CKMSG2
MOV #T17BFR,R1 ;LOW RECV ADDRESS FOR CKMSG2
MOV #T17EXP,R2 ;EXPD ADDRESS
MOV #20,R3 ;NUMBER OF BYTES TO COMPARE
JSR PC,CKMSG2 ;EXPD EQUAL RECV?
FORCERROR 1724,NOTSSR ;@@
BCS 1804 ;BR IF YES
NEXT.ERRNO
1724: ERRHRD ERRNO,T176CMP,MSGSTAT ;REPORT ERROR
TRAP C4ERHRD
.WORD 629
.WORD T176CMP
.WORD MSGSTAT

1804: CKLOOP ;LOOP ON ERROR, IF FLAG SET
TRAP C4CLP1

ENDSUB ;////////// END SUBTEST //////////
L10062: TRAP C4ESUB

TST FATFLG ;ANY FATAL ERRORS ?

```

TEST 6: SUBTEST 4: FIFO Verify ILW Status

6909 043424 001402
6910 043426 004737 017242
6911 043432

BEG 260\$;BRANCH IF NOT
JSR PC,CKDROP ;TRY TO DROP THE UNIT

260\$:

6912
6913
6914
6915
6916
6917
6918
6919
6920
6921
6922
6923
6924
6925
6926
6927
6928
6929
6930
6931
6932
6933
6934
6935
6936
6937
6938
6939
6940
6941
6942
6943
6944
6945
6946
6947
6948
6949
6950
6951

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

;;
; TEST 6: SUBTEST 5:

; SUBTEST DESCRIPTION:

; This subtest verifies that writing 64. bytes into the FIFO
; without reading any out causes the Input Ready status to
; negate. The Subtest then verifies that writing a 65th byte
; into the FIFO causes the Data In Miss status to assert.
; Next it is verified that the original 64 bytes can be read
; out correctly and that the data has not been corrupted.

; TEST STEPS:

; BEGIN

Write to TSSR to soft initialize
Do a WRITE CHARACTERISTICS to setup a message buffer
Do a Write Subsystem WRITE NPR to set tape direction out
Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
Do a Write Subsystem READ STATUS
If Input Ready NOT=0 Then Print Error
If Output Ready NOT=1 Then Print Error
If Data In Miss NOT=0 Then Print Error
Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
Do a Write Subsystem READ STATUS
If Input Ready NOT=0 Then Print Error
If Output Ready NOT=1 Then Print Error
If Data In Miss NOT=1 Then Print Error
Do Write Subsystem READ FIFO
If Data read from FIFO NOT= to Data sent Then Print Error
Do a Write Subsystem READ STATUS
If Input Ready NOT=1 Then Print Error
If Output Ready NOT=0 Then Print Error
If Data In Miss NOT=1 Then Print Error

; END

;;

6952 043432
043432
043432 104402

BGNSUB ;////////// BEGIN SUBTEST ///////////
T6.5: TRAP C#BSUB

6953
6954
6955 043434
6956 043434 004737 016034
6957 043440 103405
6958 043442 010001
6959 043444
043444 104455
043446 001165
043450 003654
043452 012074

; Write to TSSR register to soft initialize the controller

; 5\$:

JSR PC,SOFINIT ;WRITE TO TSSR TO SOFT INITIALIZE
BCS 10\$;BR IF SOFT INIT OKAY
MOV R0,R1 ;SAVE CONTENTS OF TSSR
ERRDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT

TRAP C#ERDF
.WORD 629
.WORD SFIERR
.WORD SFIMSG

TEST 6: SUBTEST 5: FIFO Verify Input Ready

```

6960          : Do a WRITE CHARACTERISTICS to setup a message buffer
6961 043454 005037 002222 10$: CLR FATFLG ;CLEAR FATAL ERROR FLAG
6962 043460 012704 047610 MOV #T17PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
6963 043464 004737 010662 JSR PC,WRTCHR ;DO WRITE CHARACTERISTICS COMMAND
6964 043470 FORCERROR 42$ ;@DFORCE ERROR IF FORCER=1
6965 043504 103407 BCS 50$ ;BR IF CARRY SET (GOOD RETURN)
6966 043506 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
6967 043510 NEXT_ERRNO
6968 043510 42$: ERRDF ERRNO,T17SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP C$ERDF
                                .WORD 630
                                .WORD T17SSR
                                .WORD PKTSSR
6969 043520 005237 002222 50$: INC FATFLG ;SET FATAL ERROR FLAG
6970 043524 104406 CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                TRAP C$CLP1
6971
6972          : Do a Write Subsystem WRITE NPR to set tape direction out
6973 043526 012700 000100 100$: MOV #NP.OUT,RO ;SET TAPE DIRECTION OUT
6974 043532 004737 047436 JSR PC,T17SNPR ;SETUP T17PK2 FOR WRITE NPR
6975 043536 012704 047760 MOV #T17PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
6976 043542 010465 000000 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
6977 043546 004737 016376 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
6978 043552 FORCERROR 102$ ;@DFORCE ERROR IF FORCER=1
6979 043566 103407 BCS 105$ ;BR IF CARRY SET (GOOD RETURN)
6980 043570 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
6981 043572 NEXT_ERRNO
6982 043572 102$: ERRDF ERRNO,T174SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP C$ERDF
                                .WORD 631
                                .WORD T174SSR
                                .WORD PKTSSR
6983 043602 005237 002222 105$: INC FATFLG ;SET FATAL ERROR FLAG
6984 043606 104406 CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                TRAP C$CLP1
6985
6986          : Do a Write Subsystem WRITE FIFO 64. bytes incrementing pattern
6987 043610 012737 000100 002312 : MOV #64.,COUNT ;WRITE 64 BYTES
6988 043616 012701 046114 MOV #T17WFDATA,R1 ;EXPD WRITE FIFO DATA BUFFER
6989 043622 012702 000100 MOV #64.,R2 ;TEST PATTERN SIZE
6990 043626 005000 CLR RO ;INCREMENT TEST PATTERN
6991 043630 110021 110$: MOVB RO,(R1)+ ;STORE INCREMENT TEST BYTE
6992 043632 005200 INC RO ;SET NEXT PATTERN
6993 043634 005302 DEC R2 ;DONE?
6994 043636 003374 BGT 110$ ;BR IF NO
6995 043640 013700 002312 MOV COUNT,RO ;FIFO BYTE COUNT
6996 043644 012701 046114 MOV #T17WFDATA,R1 ;FIFO WRITE DATA ADDRESS
6997 043650 004737 047462 JSR PC,T17WFIF ;SETUP T17PK2 FOR WRITE FIFO
6998 043654 012704 047760 MOV #T17PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
6999 043660 010465 000000 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
7000 043664 004737 016376 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
7001 043670 FORCERROR 142$ ;@DFORCE ERROR IF FORCER=1
7002 043704 103407 BCS 150$ ;BR IF CARRY SET (GOOD RETURN)
7003 043706 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
7004 043710 NEXT_ERRNO
7005 043710 142$: ERRDF ERRNO,T175SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP C$ERDF

```


TEST 6: SUBTEST 5: FIFO Verify Input Ready

```

043712 001170
043714 046446
043716 012106
7006 043720 005237 002222
7007 043724 104406
043724 104406
7008
7009
7010
7011
7012
7013 043726 004737 047374
7014 043732 012704 047760
7015 043736 010465 000000
7016 043742 004737 016376
7017 043746
7018 043762 103407
7019 043764 010001
7020 043766
7021 043766
043766 104455
043770 001171
043772 046336
043774 012106
7022 043776 005237 002222
7023 044002 104406
044002 104406
7024
7025 044004 004737 047556
7026 044010 012701 046012
7027 044014 012702 047652
7028 044020 012221
7029 044022 011211
7030 044024 042711 000020
7031 044030 052711 000040
7032 044034 042711 000200
7033 044040 005000
7034 044042 012701 047632
7035 044046 012702 045772
7036 044052 012703 000024
7037 044056 004737 011540
7038 044062
7039 044072 103404
7040 044074
7041 044074
044074 104456
044076 001172
044100 046733
044102 012410
7042 044104 104406
044104 104406
7043
7044
7045
7046 044106 012700 000001
7047 044112 012701 046114
7048 044116 004737 047462

```

```

                                .WORD 632
                                .WORD T175SSR
                                .WORD PKTSSR
150$: INC FATFLG ;SET FATAL ERROR FLAG
      CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                TRAP C4CLP1
: Do a Write Subsystem READ STATUS
: If Input Ready NOT=0 Then Print Error
: If Output Ready NOT=1 Then Print Error
: If Data In Miss NOT=0 Then Print Error
      JSR PC,T17SRD ;SETUP PACKET FOR READ STATUS
      MOV #T17PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
      MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
      JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
      FORCERROR 157$ ;@@DFORCE ERROR IF FORCER=1
      BCS 160$ ;BR IF CARRY SET (GOOD RETURN)
      MOV R0,R1 ;SAVE CONTENTS OF TSSR
      NEXT.ERRNO
157$: ERRDF ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP C4ERDF
                                .WORD 633
                                .WORD T173SSR
                                .WORD PKTSSR
160$: INC FATFLG ;SET FATAL ERROR FLAG
      CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                TRAP C4CLP1
: Set WORDS 0-7 of expd message buffer = to recv since not testing
      JSR PC,T17SETEXP ;SET WORDS 0-7 EXPD=RECV
      MOV #T17EXSTA,R1 ;GET EXPECTED READ STATUS
      MOV #T17BFSTA,R2 ;GET RECV READ STATUS
      MOV (R2)+,(R1)+ ;SET EXPD WORD #8 = RECV TEMP
      MOV (R2),(R1) ;SET EXPD WORD #9 = RECV TEMP
      BIC #S2.INRDY,(R1) ;SET EXP INPUT READY= 0
      BIS #S2.OTRDY,(R1) ;SET EXP OUTPUT READY= 1
      BIC #S2.DIM,(R1) ;SET EXP DATA IN MISS = 0
      CLR R0 ;HIGH RECV ADDRESS FOR CKMSG2
      MOV #T17BFR,R1 ;LOW RECV ADDRESS FOR CKMSG2
      MOV #T17EXP,R2 ;EXPD ADDRESS
      MOV #20,R3 ;NUMBER OF BYTES TO COMPARE
      JSR PC,CKMSG2 ;EXPD EQUAL RECV?
      FORCERROR 162$,NOTSSR ;@@
      BCS 170$ ;BR IF YES
      NEXT.ERRNO
162$: ERRHRD ERRNO,T173CMP,MSGSTAT ;REPORT ERROR
                                TRAP C4ERHRD
                                .WORD 634
                                .WORD T173CMP
                                .WORD MSGSTAT
170$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                TRAP C4CLP1
: Do a Write Subsystem WRITE FIFO 1 byte for a total of 65. written
      MOV #1,R0 ;FIFO BYTE COUNT
      MOV #T17WFDATA,R1 ;FIFO WRITE DATA ADDRESS
      JSR PC,T17WFIF ;SETUP T17PK2 FOR WRITE FIFO

```

M13

TEST 6: SUBTEST 5: FIFO Verify Input Ready

```

7049 044122 012704 047760      MOV      #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
7050 044126 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
7051 044132 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
7052 044136                    FORCERROR 172$          ;@@DFORCE ERROR IF FORCER=1
7053 044152 103407            BCS      180$          ;BR IF CARRY SET (GOOD RETURN)
7054 044154 010001            MOV      R0,R1         ;SAVE CONTENTS OF TSSR
7055 044156                    NEXT,ERRNO
7056 044156 172$:  ERRDF  ERRNO,T175SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD    635
                                .WORD    T175SSR
                                .WORD    PKTSSR
                                044156 104455
                                044160 001173
                                044162 046446
                                044164 012106
7057 044166 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
7058 044172 180$:  CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
7059 044172 104406
7060 ; Do a Write Subsystem READ STATUS
7061 ; If Input Ready NOT=0 Then Print Error
7062 ; If Output Ready NOT=1 Then Print Error
7063 ; If Data In Miss NOT=1 Then Print Error
7064 044174 004737 047374      JSR      PC,T17SRD     ;SETUP PACKET FOR READ STATUS
7065 044200 012704 047760      MOV      #T17PK2,R4    ;GET WRITE SUBSYSTEM COMMAND PACKET
7066 044204 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
7067 044210 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
7068 044214                    FORCERROR 187$          ;@@DFORCE ERROR IF FORCER=1
7069 044230 103407            BCS      190$          ;BR IF CARRY SET (GOOD RETURN)
7070 044232 010001            MOV      R0,R1         ;SAVE CONTENTS OF TSSR
7071 044234                    NEXT,ERRNO
7072 044234 187$:  ERRDF  ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD    636
                                .WORD    T173SSR
                                .WORD    PKTSSR
                                044234 104455
                                044236 001174
                                044240 046336
                                044242 012106
7073 044244 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
7074 044250 190$:  CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
7075 044250 104406
7076 ; Set WORDS 0-7 of expd message buffer = to recv since not testing
7076 044252 004737 047556      JSR      PC,T17SETEXP  ;SET WORDS 0-7 EXPD=RECV
7077 044256 012701 046012      MOV      #T17EXSTA,R1  ;GET EXPECTED READ STATUS
7078 044262 012702 047652      MOV      #T17BFSTA,R2  ;GET RECV READ STATUS
7079 044266 012221            MOV      (R2), (R1)    ;SET EXPD WORD #8 = RECV TEMP
7080 044270 011211            MOV      (R2), (R1)    ;SET EXPD WORD #9 = RECV TEMP
7081 044272 042711 000020      BIC      #S2.INRDY,(R1) ;SET EXP INPUT READY= 0
7082 044276 052711 000040      BIS      #S2.OTRDY,(R1) ;SET EXP OUTPUT READY= 1
7083 044302 052711 000200      BIS      #S2.DIM,(R1)  ;SET EXP DATA IN MISS = 1
7084 044306 005000            CLR      R0            ;HIGH RECV ADDRESS FOR CKMSG2
7085 044310 012701 047632      MOV      #T17BFR,R1    ;LOW RECV ADDRESS FOR CKMSG2
7086 044314 012702 045772      MOV      #T17EXP,R2    ;EXPD ADDRESS
7087 044320 012703 000024      MOV      #20,R3        ;NUMBER OF BYTES TO COMPARE
7088 044324 004737 011540      JSR      PC,CKMSG2     ;EXPD EQUAL RECV?
7089 044330                    FORCERROR 192$,NOTSSR  ;@@
7090 044340 103404            BCS      200$          ;BR IF YES
7091 044342                    NEXT,ERRNO
7092 044342 192$:  ERRHRD ERRNO,T173CMP,MSGSTAT ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD    637
                                .WORD    T173CMP
                                044342 104456
                                044344 001175
                                044346 046733

```

N13

TEST 6: SUBTEST 5: FIFO Verify Input Ready

```

7093 044350 012410
      044352 104406      200$: CKLOOP ;LOOP ON ERROR, IF FLAG SET .WORD MSGSTAT
      044352 104406      ; TRAP C$CLP1
7094 : Do Write Subsystem READ FIFO
7095 044354 013700 002312 MOV COUNT,R0 ;SET READ BYTE COUNT
7096 044360 004737 047516 JSR PC,T17RFIF ;SETUP T17PK2 FOR READ FIFO
7097 044364 012704 047760 MOV #T17PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
7098 044370 010465 000000 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
7099 044374 004737 016376 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
7100 044400 FORCERROR 212$ ;@@DFORCE ERROR IF FORCER=1
7101 044414 103407 BCS 220$ ;BR IF CARRY SET (GOOD RETURN)
7102 044416 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
7103 044420 NEXT.ERRNO
7104 044420 212$: ERRDF ERRNO,T176SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      044420 104455 TRAP C$ERDF
      044422 001176 .WORD 638
      044424 046512 .WORD T176SSR
      044426 012106 .WORD PKTSSR
7105 044430 005237 002222 INC FATFLG ;SET FATAL ERROR FLAG
7106 044434 104406 220$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
      044434 104406 ; TRAP C$CLP1
7107 :
7108 : If Data read from FIFO NOT= to Data sent Then Print Error
7109 044436 005000 CLR RO ;HIGH RECV ADDRESS FOR CKMSG2
7110 044440 012702 046114 MOV #T17WFDATA,R2 ;GET EXPECTED ADDRESS FOR CKMSG2
7111 044444 012701 047652 MOV #T17BFSTA,R1 ;GET RECEIVED ADDRESS FOR CKMSG2
7112 044450 013703 002312 MOV COUNT,R3 ;NUMBER OF BYTES TO COMPARE
7113 044454 004737 011540 JSR PC,CKMSG2 ;EXPD EQUAL RECV?
7114 044460 FORCERROR 232$,NOTSSR ;@@
7115 044470 103406 BCS 240$ ;BR IF YES
7116 044472 NEXT.ERRNO
7117 044472 013701 002312 232$: MOV COUNT,R1 ;GET BYTE COUNT
7118 044476 104456 ERRHRD ERRNO,T175CMP,FIFEXP ;REPORT ERROR
      044476 104456 TRAP C$ERHRD
      044500 001177 .WORD 639
      044502 047102 .WORD T175CMP
      044504 012230 .WORD FIFEXP
7119 044506 104406 240$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
      044506 104406 ; TRAP C$CLP1
7120 :
7121 : Do a Write Subsystem READ STATUS
7122 : If Input Ready NOT=1 Then Print Error
7123 : If Output Ready NOT=0 Then Print Error
7124 : If Data In Miss NOT=1 Then Print Error
7125 044510 004737 047374 JSR PC,T17SRD ;SETUP PACKET FOR READ STATUS
7126 044514 012704 047760 MOV #T17PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
7127 044520 010465 000000 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
7128 044524 004737 016376 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
7129 044530 FORCERROR 252$ ;@@DFORCE ERROR IF FORCER=1
7130 044544 103407 BCS 260$ ;BR IF CARRY SET (GOOD RETURN)
7131 044546 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
7132 044550 NEXT.ERRNO
7133 044550 252$: ERRDF ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      044550 104455 TRAP C$ERDF
      044552 001200 .WORD 640
      044554 046336 .WORD T173SSR
      044556 012106 .WORD PKTSSR

```

TEST 6: SUBTEST 5: FIFO Verify Input Ready

```

7134 044560 005237 002222          INC      FATFLG          ;SET FATAL ERROR FLAG
7135 044564          260$:      CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                044564 104406          TRAP      C$CLP1
7136          ;          Set WORDS 0-7 of expd message buffer = to recv since not testing
7137 044566 004737 047556          JSR      PC,T17SETEXP    ;SET WORDS 0-7 EXPD=RECV
7138 044572 012701 046012          MOV      #T17EXSTA,R1   ;GET EXPECTED READ STATUS
7139 044576 012702 047652          MOV      #T17BFSTA,R2   ;GET RECV READ STATUS
7140 044602 012221          MOV      (R2)+,(R1)+    ;SET EXPD WORD #8 = RECV TEMP
7141 044604 011211          MOV      (R2),(R1)      ;SET EXPD WORD #9 = RECV TEMP
7142 044606 052711 000020          BIS      #S2.INRDY,(R1) ;SET EXP INPUT READY= 1
7143 044612 042711 000040          BIC      #S2.OURDY,(R1) ;SET EXP OUTPUT READY= 0
7144 044616 052711 000200          BIS      #S2.DIM,(R1)  ;SET EXP DATA IN MISS = 1
7145 044622 005000          CLR      R0             ;HIGH RECV ADDRESS FOR CKMSG2
7146 044624 012701 047632          MOV      #T17BFR,R1    ;LOW RECV ADDRESS FOR CKMSG2
7147 044630 012702 045772          MOV      #T17EXP,R2    ;EXPD ADDRESS
7148 044634 012703 000024          MOV      #20.,R3       ;NUMBER OF BYTES TO COMPARE
7149 044640 004737 011540          JSR      PC,CKMSG2     ;EXPD EQUAL RECV?
7150 044644          FORCERROR 272$,NOTSSR  ;@@D
7151 044654          BCS      280$          ;BR IF YES
                                103404          NEXT,ERRNO
7152 044656          272$:      ERRHRD  ERRNO,T174CMP,MSGSTAT ;REPORT ERROR
                                044656 104456          TRAP      C$ERHRD
                                044660 001201          .WORD    641
                                044662 047017          .WORD    T174CMP
                                044664 012410          .WORD    MSGSTAT
7154 044666          280$:      CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                044666 104406          TRAP      C$CLP1
7155          ENDSUB
7156 044670          ;////////// END SUBTEST //////////
                                044670          L10063:
                                044670 104403          TRAP      C$ESUB
7157          TST      FATFLG          ;ANY FATAL ERRORS ?
7158 044672 005737 002222          BEQ      300$          ;BRANCH IF NOT
7159 044676 001402          JSR      PC,CKDROP     ;TRY TO DROP THE UNIT
7160 044700 004737 017242          300$:
7161 044704

```

```

7162
7163
7164
7165          .SBTTL  TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test
7166

```

```

7167          ;**
7168          ; TEST 6: SUBTEST 6:
7169          ;
7170          ; SUBTEST DESCRIPTION:
7171          ;
7172          ; This subtest verifies that the Reset FIFO function within
7173          ; the Write Miscellaneous Control 1 function initializes
7174          ; the FIFO to correct initial status. The following steps
7175          ; are performed:
7176          ; 1. Reset an already initialized FIFO and check for
7177          ;    proper status.
7178          ; 2. Write a varying number of bytes (1-65.) into the
7179          ;    FIFO and verify that after each block of bytes is
7180          ;    written the FIFO can be reset to it's initial
7181          ;    state.
7182          ;

```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```

7183      ; TEST STEPS:
7184      ;
7185      ; BEGIN
7186      ;   Write to TSSR to soft initialize
7187      ;   Do a WRITE CHARACTERISTICS to setup a message buffer
7188      ;   Do a Write Subsystem Write Misc to Reset FIFO
7189      ;   Do a Write Subsystem READ STATUS
7190      ;   If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
7191      ;   signals NOT=0 Then Print Error
7192      ;   Do a Write Subsystem WRITE NPR to set tape direction out
7193      ;
7194      ; REPEAT FOR BYTE COUNT 1 TO 65.
7195      ; BEGIN
7196      ;   Do a Write Subsystem WRITE FIFO with the current byte count
7197      ;   Do a Write Subsystem Write Misc to Reset FIFO
7198      ;   Do a Write Subsystem READ STATUS
7199      ;   If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
7200      ;   signals NOT=0 Then Print Error
7201      ; END
7202      ;--
7203 044704      BGNSUB                      ;//////////////// BEGIN SUBTEST //////////////////
      044704      T6.6:                      TRAP      C#BSUB
      044704      104402
7204
7205      ;
7206 044706      5$:      Write to TSSR register to soft initialize the controller
7207 044706      004737      016034      JSR      PC,SOFINIT          ;WRITE TO TSSR TO SOFT INITIALIZE
7208 044712      103405      BCS      10$          ;BR IF SOFT INIT OKAY
7209 044714      010001      MOV      R0,R1          ;SAVE CONTENTS OF TSSR
7210 044716      ERRDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
      044716      104455      TRAP      C#ERDF
      044720      001201      .WORD      641
      044722      003654      .WORD      SFIERR
      044724      012074      .WORD      SFIMSG
7211      ;
7212 044726      005037      002222      ; Do a WRITE CHARACTERISTICS to setup a message buffer
7213 044732      012704      047610      CLR      FATFLG          ;CLEAR FATAL ERROR FLAG
7214 044736      004737      010662      MOV      @T17PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
7215 044742      FORCERROR      42$      JSR      PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
7216 044756      103407      BCS      50$          ;@DFORCE ERROR IF FORCER=1
7217 044760      010001      MOV      R0,R1          ;BR IF CARRY SET (GOOD RETURN)
7218 044762      NEXT.ERRNO      ;SAVE CONTENTS OF TSSR
7219 044762      42$:      ERRDF      ERRNO,T17SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      044762      104455      TRAP      C#ERDF
      044764      001202      .WORD      642
      044766      046235      .WORD      T17SSR
      044770      012106      .WORD      PKTSSR
7220 044772      005237      002222      INC      FATFLG          ;SET FATAL ERROR FLAG
7221 044776      50$:      CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      044776      104406      TRAP      C#CLP1
7222      ;
7223 045000      004737      047414      ; Do a Write Subsystem Write Misc to Reset FIFO
7224 045004      012704      047760      JSR      PC,T17RSFIF      ;SETUP PKT FOR WRITE MISC RESET FIFO
7225 045010      010465      000000      MOV      @T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
7226 045014      004737      016376      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
7227 045020      FORCERROR      62$      JSR      PC,CHKTSSR          ;WAIT FOR SSR TO SET
7228 045034      103407      BCS      70$          ;@DFORCE ERROR IF FORCER=1
      ;BR IF CARRY SET (GOOD RETURN)

```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```

7229 045036 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
7230 045040      NEXT.ERRNO
7231 045040      62$:  ERRDF  ERRNO,T172SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      045040 104455      TRAP      C$ERDF
      045042 001203      .WORD    643
      045044 046272      .WORD    T172SSR
      045046 012106      .WORD    PKTSSR
7232 045050 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
7233 045054      70$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      045054 104406      TRAP      C$CLP1

7234
7235      ;      Do a Write Subsystem READ STATUS
7236      ;      If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
7237      ;      signals NOT=0 Then Print Error
7238 045056 004737 047374      JSR      PC,T17SRD      ;SETUP PACKET FOR READ STATUS
7239 045062 012704 047760      MOV      #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
7240 045066 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
7241 045072 004737 016376      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
7242 045076      FORCERROR 77$      ;@@DFORCE ERROR IF FORCER=1
7243 045112 103407      BCS      80$      ;BR IF CARRY SET (GOOD RETURN)
7244 045114 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
7245 045116      NEXT.ERRNO
7246 045116      77$:  ERRDF  ERRNO,T173SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      045116 104455      TRAP      C$ERDF
      045120 001204      .WORD    644
      045122 046336      .WORD    T173SSR
      045124 012106      .WORD    PKTSSR
7247 045126 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
7248 045132      80$:  CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      045132 104406      TRAP      C$CLP1
7249 045134 004737 047556      JSR      PC,T17SETEXP      ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
7250 045140 012701 046012      MOV      #T17EXSTA,R1      ;GET EXPECTED READ STATUS
7251 045144 012702 047652      MOV      #T17BFSTA,R2      ;GET RECV READ STATUS
7252 045150 011211      MOV      (R2),(R1)      ;SET EXPD WORD #8 = RECV TEMP
7253 045152 042711 002000      BIC      #S1.ICER,(R1)      ;SET EXPD ICER =0
7254 045156 042711 001000      BIC      #S1.IFMK,(R1)      ;SET EXPD IFMK =0
7255 045162 042711 000400      BIC      #S1.IHER,(R1)      ;SET EXPD IHER =0
7256 045166 016261 000002 000002      MOV      2(R2),2(R1)      ;SET EXPD WORD #9 = RECV (NOT TESTING)
7257 045174 005000      CLR      R0      ;HIGH RECV ADDRESS FOR CKMSG2
7258 045176 012701 047632      MOV      #T17BFR,R1      ;LOW RECV ADDRESS FOR CKMSG2
7259 045202 012702 045772      MOV      #T17EXP,R2      ;EXPD ADDRESS
7260 045206 012703 000024      MOV      #20.,R3      ;NUMBER OF BYTES TO COMPARE
7261 045212 004737 011540      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
7262 045216      FORCERROR 92$,NOTSSR      ;@@
7263 045226 103404      BCS      100$      ;BR IF YES
7264 045230      NEXT.ERRNO
7265 045230      92$:  ERRHRD  ERRNO,T177CMP,MSGSTAT ;REPORT ERROR
      045230 104456      TRAP      C$ERHRD
      045232 001205      .WORD    645
      045234 047264      .WORD    T177CMP
      045236 012410      .WORD    MSGSTAT
7266 045240      100$: CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      045240 104406      TRAP      C$CLP1

7267
7268      ;      Do a Write Subsystem WRITE NPR to set tape direction out
7269 045242 012700 000100      MOV      #NP.OUT,R0      ;SET TAPE DIRECTION OUT
7270 045246 004737 047436      JSR      PC,T17SNPR      ;SETUP T17PK2 FOR WRITE NPR

```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```

7271 045252 012704 047760      MOV      #T17PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
7272 045256 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
7273 045262 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
7274 045266          FORCERROR 112$      ;@DFORCE ERROR IF FORCER=1
7275 045302 103407          BCS      120$           ;BR IF CARRY SET (GOOD RETURN)
7276 045304 010001          MOV      R0,R1         ;SAVE CONTENTS OF TSSR
7277 045306          NEXT.ERRNO
7278 045306          112$: ERRDF  ERRNO,T174SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD    646
                                .WORD    T174SSR
                                .WORD    PKTSSR
                                045306 104455
                                045310 001206
                                045312 046403
                                045314 012106
7279 045316 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
7280 045322          120$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                045322 104406
7281          ;
7282          ; Setup incrementing pattern in FIFO data buffer
7283 045324 012701 046012      MOV      #T17EXSTA,R1  ;EXPD WRITE FIFO DATA BUFFER
7284 045330 012702 000100      MOV      #64.,R2      ;TEST PATTERN SIZE
7285 045334 005000          CLR      R0           ;INCREMENT TEST PATTERN
7286 045336 110021          130$: MOVB   R0,(R1)+  ;STORE INCREMENT TEST BYTE
7287 045340 005200          INC      R0          ;SET NEXT PATTERN
7288 045342 005302          DEC      R2          ;DONE?
7289 045344 003374          BGT      130$        ;BR IF NO
7290          ;
7291          ; REPEAT FOR BYTE COUNT 1 TO 65.
7292 045346 012737 000001 002312 ;
7293          ; Do a Write Subsystem WRITE FIFO with the current byte count
7294 045354          150$:
7295 045354 013700 002312      MOV      COUNT,R0     ;REPEAT LOOP LABEL
7296 045360 012701 046012      MOV      #T17EXSTA,R1 ;FIFO BYTE COUNT
7297 045364 004737 047462      JSR      PC,T17WFIF   ;FIFO WRITE DATA ADDRESS
7298 045370 012704 047760      MOV      #T17PK2,R4  ;SETUP T17PK2 FOR WRITE FIFO
7299 045374 010465 000000      MOV      R4,TSDB(R5) ;GET WRITE SUBSYSTEM COMMAND PACKET
7300 045400 004737 016376      JSR      PC,CHKTSSR  ;SET THE PACKET ADDRESS TO EXECUTE
7301 045404          FORCERROR 152$      ;WAIT FOR SSR TO SET
7302 045420 103407          BCS      160$        ;@DFORCE ERROR IF FORCER=1
7303 045422 010001          MOV      R0,R1         ;BR IF CARRY SET (GOOD RETURN)
7304 045424          NEXT.ERRNO         ;SAVE CONTENTS OF TSSR
7305 045424          152$: ERRDF  ERRNO,T175SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD    647
                                .WORD    T175SSR
                                .WORD    PKTSSR
                                045424 104455
                                045426 001207
                                045430 046446
                                045432 012106
7306 045434 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
7307 045440          160$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                045440 104406
7308          ;
7309          ; Do a Write Subsystem Write Misc to Reset FIFO
7310 045442 004737 047414      JSR      PC,T17RSFIF  ;SETUP PKT FOR WRITE MISC RESET FIFO
7311 045446 012704 047760      MOV      #T17PK2,R4  ;GET WRITE SUBSYSTEM COMMAND PACKET
7312 045452 010465 000000      MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
7313 045456 004737 016376      JSR      PC,CHKTSSR  ;WAIT FOR SSR TO SET
7314 045462          FORCERROR 162$      ;@DFORCE ERROR IF FORCER=1
7315 045476 103407          BCS      170$        ;BR IF CARRY SET (GOOD RETURN)
7316 045500 010001          MOV      R0,R1         ;SAVE CONTENTS OF TSSR
7317 045502          NEXT.ERRNO

```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```

7318 045502          162$: ERRDF  ERRNO,T172SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
      045502 104455          TRAP  C$ERDF
      045504 001210          .WORD  648
      045506 046272          .WORD  T172SSR
      045510 012106          .WORD  PKTSSR
7319 045512 005237 002222          INC  FATFLG          ;SET FATAL ERROR FLAG
7320 045516          170$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      045516 104406          TRAP  C$CLP1

7321
7322 ; Do a Write Subsystem READ STATUS
7323 ; If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
7324 ; signals NOT=0 Then Print Error
7325 045520 004737 047374          JSR  PC,T17SRD          ;SETUP PACKET FOR READ STATUS
7326 045524 012704 047760          MOV  #T17PK2,R4          ;GET WRITE SUBSYSTEM COMMAND PACKET
7327 045530 010465 000000          MOV  R4,TSDB(R5)          ;SET THE PACKET ADDRESS TO EXECUTE
7328 045534 004737 016376          JSR  PC,CHKTSSR          ;WAIT FOR SSR TO SET
7329 045540          FORCERROR 177$          ;@@@FORCE ERROR IF FORCER=1
7330 045554 103407          BCS  180$          ;BR IF CARRY SET (GOOD RETURN)
7331 045556 010001          MOV  R0,R1          ;SAVE CONTENTS OF TSSR
7332 045560          NEXT.ERRNO
7333 045560          177$: ERRDF  ERRNO,T173SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
      045560 104455          TRAP  C$ERDF
      045562 001211          .WORD  649
      045564 046336          .WORD  T173SSR
      045566 012106          .WORD  PKTSSR
7334 045570 005237 002222          INC  FATFLG          ;SET FATAL ERROR FLAG
7335 045574          180$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      045574 104406          TRAP  C$CLP1
7336 045576 004737 047556          JSR  PC,T17SETEXP          ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
7337 045602 012701 046012          MOV  #T17EXSTA,R1          ;GET EXPECTED READ STATUS
7338 045606 012702 047652          MOV  #T17BFSTA,R2          ;GET RECV READ STATUS
7339 045612 011211          MOV  (R2),(R1)          ;SET EXPD WORD #8 = RECV TEMP
7340 045614 042711 002000          BIC  #S1.ICER,(R1)          ;SET EXPD ICER =0
7341 045620 042711 001000          BIC  #S1.IFMK,(R1)          ;SET EXPD IFMK =0
7342 045624 042711 000400          BIC  #S1.IHER,(R1)          ;SET EXPD IHER =0
7343 045630 016261 000002 000002          MOV  2(R2),2(R1)          ;SET EXPD WORD #9 = RECV (NOT TESTING)
7344 045636 005000          CLR  R0          ;HIGH RECV ADDRESS FOR CKMSG2
7345 045640 012701 047632          MOV  #T17BFR,R1          ;LOW RECV ADDRESS FOR CKMSG2
7346 045644 012702 045772          MOV  #T17EXP,R2          ;EXPD ADDRESS
7347 045650 012703 000024          MOV  #20.,R3          ;NUMBER OF BYTES TO COMPARE
7348 045654 004737 011540          JSR  PC,CKMSG2          ;EXPD EQUAL RECV?
7349 045660          FORCERROR 192$,NOTSSR          ;@@@
7350 045670 103404          BCS  200$          ;BR IF YES
7351 045672          NEXT.ERRNO
7352 045672          192$: ERRHRD  ERRNO,T177CMP,MSGSTAT  ;REPORT ERROR
      045672 104456          TRAP  C$ERHRD
      045674 001212          .WORD  650
      045676 047264          .WORD  T177CMP
      045700 012410          .WORD  MSGSTAT
7353 045702          200$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      045702 104406          TRAP  C$CLP1
7354
7355
7356 045704          250$: FORCEXIT 260$
7357 045704          INC  COUNT
7358 045714 005237 002312          CMP  COUNT, #65.          ;GET NEXT BYTE COUNT
7359 045720 023727 002312 000101          ;DONE ALL BYTES?
    
```


TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```

7360 045726 101002          BHI      260$
7361 045730 000137 045354  JMP      150$          ;BR IF YES
7362 045734          260$:          ;DO ANOTHER BYTE COUNT
7363
7364 045734          ENDSUB          ;//////////////// END SUBTEST //////////////////
          045734          L10064: TRAP C$ESUB
          045734 104403
7365
7366 045736 005737 002222    TST      FATFLG          ;ANY FATAL ERRORS ?
7367 045742 001402          BEQ      300$          ;BRANCH IF NOT
7368 045744 004737 017242    JSR      PC,CKDROP       ;TRY TO DROP THE UNIT
7369 045750 004737 016516    JSR      PC,TSTLOOP      ;DO ITERATIONS?
7370 045754 103002          BCC      305$          ;BR IF NO
7371 045756 000137 040014    JMP      T17LOOP         ;LOOP UNTIL ITERATIONS DONE
7372 045762          305$:
7373
7374 045762          EXIT      TST          ;//////////////// EXIT TEST //////////////////
          045762 104432          TRAP      C$EXIT
          045764 002112          .WORD    L10056-.
7375
7376
7377
7378          ;*
7379          ;LOCAL STORAGE FOR THIS TEST
7380          ;-
7381
7382 045766          T17MSK:          ;MASK OF UNTESTED BITS IN READ STATUS BYTES
7383          ;UNTESTED BITS ARE SET TO 1
7384 045766          377          .BYTE    †C<000>       ;BYTE 0 MASK
7385 045767          037          .BYTE    †C<340>       ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
7386 045770          360          .BYTE    †C<017>       ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
7387 045771          000          .BYTE    0             ;MAKE IT EVEN
7388
7389 045772          T17EXP:          ;BEGIN EXPECTED DATA BUFFER
7390 045772 000000          .WORD    0             ;MESSAGE TYPE
7391 045774 000000          .WORD    0             ;DATA FIELD LENGTH
7392 045776 000000          .WORD    0             ;RBPCR
7393 046000 000000          .WORD    0             ;XST0
7394 046002 000000          .WORD    0             ;XST1
7395 046004 000000          .WORD    0             ;XST2
7396 046006 000000          .WORD    0             ;XST3
7397 046010 000000          .WORD    0             ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
7398 046012          T17EXSTA: .BLKB 66.  ;EXPECTED READ STATUS AND WRITE FIFO DATA
7399 046114          T17EXEND:          ;END EXPECTED DATA BUFFER
7400
7401 046114          T17WFDATA: .BLKB 66.  ;WRITE FIFO EXPECTED DATA BUFFER
7402
7403
7404          ;*
7405          ;LOCAL TEXT MESSAGES FOR TEST
7406          ;-
7407 046216          106          111          106  TST17ID:          .ASCIZ  'FIFO Exerciser'
7408 046235          127          122          111  T17SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
7409 046272          127          122          111  T172SSR: .ASCIZ  'WRITE SUBSYSTEM (Write Misc) Failed'
7410 046336          127          122          111  T173SSR: .ASCIZ  'WRITE SUBSYSTEM (Read Status) Failed'
7411 046403          127          122          111  T174SSR: .ASCIZ  'WRITE SUBSYSTEM (Write Npr) Failed'
7412 046446          127          122          111  T175SSR: .ASCIZ  'WRITE SUBSYSTEM (Write FIFO) Failed'

```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```

7413 046512      127      122      111 T176SSR:.ASCIZ 'WRITE SUBSYSTEM (Read FIFO) Failed'
7414 046555      106      111      106 T171CMP:.ASCIZ 'FIFO Status in WORD #9 Incorrect after Initialize'
7415 046637      122      145      141 T172CMP:.ASCIZ 'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
7416 046733      106      111      106 T173CMP:.ASCIZ 'FIFO Status (In WORD #9) Incorrect after WRITE FIFO'
7417 047017      106      111      106 T174CMP:.ASCIZ 'FIFO Status (In WORD #9) Incorrect after READ FIFO'
7418 047102      122      145      141 T175CMP:.ASCIZ 'Read FIFO Data not equal to Write FIFO Data'
7419 047156      106      111      106 T176CMP:.ASCIZ 'FIFO Status (In WORD #9) Incorrect after READ FIFO from an Empty FIFO'
7420 047264      106      111      106 T177CMP:.ASCIZ 'FIFO Status (In WORD #9) Incorrect after RESET FIFO'
7421
7422
7423
7424
7425
7426 047350
7427 047350
7428 047354 012701 047632
7429 047360 012702 000120
7430 047364 105021
7431 047366 005302
7432 047370 003375
7433 047372 000207
7434
7435
7436
7437
7438 047374
7439 047374 004737 047350
7440 047400 012700 047770
7441 047404 112720 000005
7442 047410 105010
7443 047412 000207
7444
7445
7446
7447
7448 047414
7449 047414 004737 047350
7450 047420 012700 047770
7451 047424 112720 000010
7452 047430 112710 000030
7453 047434 000207
7454
7455
7456
7457
7458
7459
7460
7461
7462
7463 047436
7464 047436 004737 047350
7465 047442 012701 047770
7466 047446 112721 000011
7467 047452 052700 000020
7468 047456 110011
7469 047460 000207

```

```

;*
; CLEAR MESSAGE BUFFER
;-
T17CLRBUF:
        SAVREG
        MOV     #T17BFR,R1
        MOV     #T17BEND-T17BFR,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 T17PK2 PACKET FOR READ STATUS
;-
T17SRD:
        JSR    PC,T17CLRBUF
        MOV     #T17DT2,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 T17PK2 PACKET FOR WRITE MISC RESET FIFO
;-
T17RSFIF:
        JSR    PC,T17CLRBUF
        MOV     #T17DT2,R0
        MOV    #PW.WMISC,(R0)+
        MOV    #MS.RSFIF!MS.RSTAP,(R0)
        RTS    PC
; CLEAR MESSAGE BUFFER
; WRITE SUBSYSTEM DATA BUFFER
; STORE WRITE MISCELLANEOUS IN BSEL0
; STORE BSEL1 CLEAR FIFO CODES
; RETURN

```

```

;*
; SETUP T17PK2 PACKET FOR WRITE NPR
; INPUT:
; RO CONTAINS BSEL1 NPR DATA
;
; SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
;-
T17SNPR:
        JSR    PC,T17CLRBUF
        MOV     #T17DT2,R1
        MOV    #PW.WNPR,(R1)+
        BIS    #NP.WRP,R0
        MOV    R0,(R1)
        RTS    PC
; CLEAR MESSAGE BUFFER
; WRITE SUBSYSTEM DATA BUFFER
; STORE WRITE NPR IN BSEL0
; DON'T WRITE WRONG PARITY
; STORE NPR DATA IN BSEL1
; RETURN

```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```

7470
7471
7472      ;*
7473      ; SETUP T17PK2 PACKET FOR WRITE FIFO
7474      ;
7475      ; INPUT:
7476      ;       R0 CONTAINS BYTE COUNT
7477      ;       R1 CONTAINS DATA PATTERN BLOCK ADDRESS
7478      ;-
7478 047462 T17WFIF:
7479 047462      SAVREG                               ;SAVE R1-R5 UNTIL NEXT RETURN
7480 047466      JSR      PC,T17CLRBUF                ;CLEAR MESSAGE BUFFER
7481 047472      MOV      #T17DT2,R2                 ;WRITE SUBSYSTEM DATA BUFFER
7482 047476      MOVB    #PW.WFIFO,(R2)+            ;STORE WRITE FIFO IN BSEL0
7483 047502      MOVB    R0,(R2)+                    ;STORE BYTE COUNT IN BSEL1
7484 047504      CLR      (R2)+                      ;CLEAR SEL2 (UNUSED)
7485 047506      MOVB    (R1)+,(R2)+                ;STORE DATA PATTERN BYTE
7486 047510      DEC     R0                           ;DONE ALL BYTES?
7487 047512      BGT     10$                          ;BR IF NO
7488 047514      RTS     PC                           ;RETURN
7489
7490
7491      ;*
7492      ; SETUP T17PK2 PACKET FOR READ FIFO
7493      ;
7494      ; INPUT:
7495      ;       R0 CONTAINS SEL2 BYTE COUNT
7496      ;-
7496 047516 T17RFIF:
7497 047516      JSR      PC,T17CLRBUF                ;CLEAR MESSAGE BUFFER
7498 047522      MOV      #T17DT2,R1                 ;WRITE SUBSYSTEM DATA BUFFER
7499 047526      MOVB    #PW.RFIFO,(R1)+            ;STORE READ FIFO IN BSEL0
7500 047532      MOVB    R0,(R1)+                    ;STORE BYTE COUNT IN BSEL1
7501 047534      RTS     PC                           ;RETURN
7502
7503      ;*
7504      ; CLEAR EXPECTED DATA MESSAGE BUFFER
7505      ;-
7505 047536 T17CLEXP:
7506 047536      MOV      #T17EXP,R1                 ;GET EXPD ADDRESS
7507 047542      MOV      #T17EXEND-T17EXP,R0        ;GET EXPD SIZE
7508 047546      CLRB    (R1)+                       ;CLEAR A BYTE
7509 047550      DEC     R0                           ;DONE?
7510 047552      BGT     10$                          ;BR IF NO
7511 047554      RTS     PC                           ;RETURN
7512
7513
7514      ;*
7515      ;Set WORDS 0-7 of expd message buffer = to recv since not testing
7516      ;-
7516 047556 T17SETEXP:
7517 047556      MOV      #T17EXP,R2                 ;GET EXPD
7518 047562      MOV      #T17BFR,R3                 ;GET READ STATUS RECV BUFFER
7519 047566      MOV      #8,R0                      ;SET WORDS 0-7 EXP=RECV
7520 047572      MOVB    (R3)+,(R2)+                ;SET EXPD=RECV
7521 047574      DEC     R0                           ;DONE WORDS 0-7 WORDS?
7522 047576      BGT     5$                          ;BR IF NO
7523 047600      RTS     PC                           ;RETURN
7524
7526      .=<..10>&177770
7528      ;

```

TEST 6: SUBTEST 6: FIFO Verify Reset FIFO Test

```

7529 ;WRITE CHARACTERISTICS COMMAND PACKET
7530 ;
7531 047610 ;T17PACKET: ;COMMAND PACKET FOR TEST
7532 047610 100004 ;.WORD 100004 ;WRITE CHARACTERISTICS COMMAND, WITH ACK
7533 047612 047620 ;.WORD T17DATA ;ADDRESS OF CHARACTERISTICS BLOCK
7534 047614 000000 ;.WORD 0
7535 047616 000012 ;.WORD 10. ;MINIMUM MESSAGE PACKET SIZE
7536 ;
7537 047620 ;T17DATA: ;CHARACTERISTICS DATA BLOCK
7538 047620 047632 ;.WORD T17BFR ;ADDRESS OF MESSAGE BUFFER
7539 047622 000000 ;.WORD 0
7540 047624 000024 ;.WORD 20. ;LENGTH OF MESSAGE BUFFER
7541 047626 000000 ;.WORD 0 ;ESS,ENB,EAI,ERI
7542 047630 000000 ;.WORD 0 ;EXTENDED FEATURES UNIT NO. ETC.
7543 ;
7544 ;
7545 ;MESSAGE BUFFER FOR ALL TEST 6 COMMANDS
7546 ;
7547 047632 ;T17BFR: ;BEGIN MESSAGE BUFFER
7548 047632 000000 ;.WORD 0 ;MESSAGE TYPE
7549 047634 000000 ;.WORD 0 ;DATA FIELD LENGTH
7550 047636 000000 ;.WORD 0 ;RBPCR
7551 047640 000000 ;.WORD 0 ;XST0
7552 047642 000000 ;.WORD 0 ;XST1
7553 047644 000000 ;.WORD 0 ;XST2
7554 047646 000000 ;.WORD 0 ;XST3
7555 047650 000000 ;.WORD 0 ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
7556 047652 ;T17BFSTA: .BLKB 64. ;READ STATUS AND WRITE FIFO BUFFER
7557 047752 ;T17BEND: ;END OF MESSAGE BUFFER
7558 ;
7559 ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
7560 ;
7561 ;
7562 047760 ;.=<.10>E177770
7563 ;
7564 047760 ;T17PK2: ;WRITE SUBSYSTEM WITH ACK
7565 047760 100006 ;.WORD P.WRTSUB!P.ACK ;LOW ADDRESS OF DATA BLOCK
7566 047762 047770 ;.WORD T17DT2 ;HIGH ADDRESS OF DATA BLOCK
7567 047764 000000 ;.WORD 0 ;MINIMUM MESSAGE PACKET SIZE
7568 047766 000012 ;.WORD 10.
7569 ;
7570 047770 ;T17DT2: ;DATA BLOCK
7571 047770 000 ;.BYTE 0 ;BSELO
7572 047771 000 ;.BYTE 0 ;BSEL1
7573 047772 000000 ;.WORD 0 ;SEL2
7574 047774 ;.BLKB 66. ;WRITE FIFO DATA OUTPUT BUFFER
7575 ;
7576 050076 ;ENDTST
7577 ;
7578 ;
7579 ;
7580 ;
7581 ;
7582 ;
7583 ;
7584 ;
7585 ;

```

L10056: TRAP C4ETST

.SBTTL TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```

;*
; TEST DESCRIPTION:
;
; TEST STEPS:
;
; REPEAT FOR LOOPCNT
;

```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```

7586      : BEGIN
7587      : Write to TSSR register to soft initialize the controller
7588      : Do WRITE CHARACTERISTICS to check for Extended Features Switch
7589      : If Extended Features Hardware Switch Clear then:
7590      :   Do Write Subsystem Write Miscellaneous to Set Extended Features.
7591      : Do WRITE CHARACTERISTICS to select reserved unit 7
7592      : Do a Write Subsystem READ STATUS
7593      : If any transport interface signals are asserted then Print Error
7594      : END
7595      :--
7596
7597
7598 050100      BGNTST
7599 050100
7603 050100 012700 050606      MOV      #TST18ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
7604 050104 004737 016550      JSR      PC,TSTSETUP      ;DO INITIAL TEST SETUP
7605 050110 012737 000012 002216  MOV      #10.,LOOPCNT      ;PERFORM 10 ITERATIONS
7606 050116      T18LOOP:
7607      : Write to TSSR register to soft initialize the controller
7608 050116      5$:
7609 050116 004737 016034      JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
7610 050122 103405      BCS      10$             ;BR IF SOFT INIT OKAY
7611 050124 010001      MOV      R0,R1           ;SAVE CONTENTS OF TSSR
7612 050126      ERRDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
7613      050126 104455      TRAP      C$ERDF
7614      050130 001274      .WORD      700
7615      050132 003654      .WORD      SFIERR
7616      050134 012074      .WORD      SFIMSG
7617
7618      : Do WRITE CHARACTERISTICS to check for Extended Features Switch
7619 050136 005037 002222      10$: CLR      FATFLG          ;CLEAR FATAL ERROR FLAG
7620 050142 012704 051270      MOV      #T18PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
7621 050146 004737 010662      JSR      PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
7622 050152      FORCERROR 12$      ;@@DFORCE ERROR IF FORCER=1
7623 050166 103407      BCS      15$             ;BR IF CARRY SET (GOOD RETURN)
7624 050170 010001      MOV      R0,R1           ;SAVE CONTENTS OF TSSR
7625 050172      NEXT,ERRNO
7626 050172 12$: ERRDF      ERRNO,T18SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
7627      050172 104455      TRAP      C$ERDF
7628      050174 001275      .WORD      701
7629      050176 050645      .WORD      T18SSR
7630      050200 012106      .WORD      PKTSSR
7631 050202 005237 002222      15$: INC      FATFLG          ;SET FATAL ERROR FLAG
7632 050206 104406      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
7633      050206 104406      TRAP      C$CLP1
7634
7635      : If Extended Features Hardware Switch Clear then:
7636      :   Do Write Subsystem Write Miscellaneous to Set Extended Features.
7637 050210 012701 051312      MOV      #T18BFR,R1        ;MESSAGE BUFFER ADDRESS
7638 050214 032761 000200 000012  BIT      #X2.EXTF,XST2(R1)  ;EXTENDED FEATURES SWITCH SET?
7639 050222 001026      BNE      30$             ;BR IF YES
7640 050224 004737 051136      JSR      PC,T18SMISC       ;SETUP PACKET FOR WRITE MISCELLANEOUS
7641 050230 012704 051340      MOV      #T18PK2,R4       ;GET WRITE SUBSYSTEM COMMAND PACKET
7642 050234 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
7643 050240 004737 016376      JSR      PC,CHKTSSR       ;WAIT FOR SSR TO SET
7644 050244      FORCERROR 22$      ;@@DFORCE ERROR IF FORCER=1
7645 050260 103407      BCS      30$             ;BR IF CARRY SET (GOOD RETURN)

```

L14

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```

7637 050262 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
7638 050264          NEXT.ERRNO
7639 050264          22$:  ERRDF  ERRNO,T182SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
      050264 104455          TRAP    C#ERDF
      050266 001276          .WORD  702
      050270 050702          .WORD  T182SSR
      050272 012106          .WORD  PKTSSR
7640 050274 005237 002222  INC    FATFLG        ;SET FATAL ERROR FLAG
7641 050300          30$:  CKLOOP        ;LOOP ON ERROR, IF FLAG SET
      050300 104406          TRAP    C#CLP1
7642
7643
7644          :          Do WRITE CHARACTERISTICS to select reserved unit 7
7645 050302 005037 002222  CLR    FATFLG        ;CLEAR FATAL ERROR FLAG
7646 050306 012704 051270  MOV    #T18PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
7647 050312 004737 010662  JSR    PC,WRTCHR     ;DO WRITE CHARACTERISTICS COMMAND
7648 050316          FORCERROR 42$      ;@@DFORCE ERROR IF FORCER=1
7649 050332 103407          BCS    50$          ;BR IF CARRY SET (GOOD RETURN)
7650 050334 010001          MOV    R0,R1        ;SAVE CONTENTS OF TSSR
7651 050336          NEXT.ERRNO
7652 050336          42$:  ERRDF  ERRNO,T18SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
      050336 104455          TRAP    C#ERDF
      050340 001277          .WORD  703
      050342 050645          .WORD  T18SSR
      050344 012106          .WORD  PKTSSR
7653 050346 005237 002222  INC    FATFLG        ;SET FATAL ERROR FLAG
7654 050352          50$:  CKLOOP        ;LOOP ON ERROR, IF FLAG SET
      050352 104406          TRAP    C#CLP1
7655
7656          :          Clear message buffer
7657 050354 012701 051312  MOV    #T18BFR,R1   ;GET MESSAGE BUFFER ADDRESS
7658 050360 013700 051304  MOV    T18DATA-4,R0 ;SIZE OF MESSAGE BUFFER IN BYTES
7659 050364 105021          60$:  CLRB  (R1)-    ;CLEAR A BYTE
7660 050366 005300          DEC    R0          ;DONE?
7661 050370 003375          BGT    60$        ;BR IF NO
7662          :          Do a Write Subsystem READ STATUS
7663 050372 004737 051116  JSR    PC,T18SRD    ;SETUP PACKET FOR READ STATUS
7664 050376 012704 051340  MOV    #T18PK2,R4   ;GET WRITE SUBSYSTEM COMMAND PACKET
7665 050402 010465 000000  MOV    R4,TSDB(R5)  ;SET THE PACKET ADDRESS TO EXECUTE
7666 050406 004737 016376  JSR    PC,CHKTSSR   ;WAIT FOR SSR TO SET
7667 050412          FORCERROR 62$      ;@@DFORCE ERROR IF FORCER=1
7668 050426 103407          BCS    70$        ;BR IF CARRY SET (GOOD RETURN)
7669 050430 010001          MOV    R0,R1        ;SAVE CONTENTS OF TSSR
7670 050432          NEXT.ERRNO
7671 050432          62$:  ERRDF  ERRNO,T183SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
      050432 104455          TRAP    C#ERDF
      050434 001300          .WORD  704
      050436 050746          .WORD  T183SSR
      050440 012106          .WORD  PKTSSR
7672 050442 005237 002222  INC    FATFLG        ;SET FATAL ERROR FLAG
7673 050446          70$:  CKLOOP        ;LOOP ON ERROR, IF FLAG SET
      050446 104406          TRAP    C#CLP1
7674
7675
7676          :          Set first 8 words of expd message buffer = to recv since not testing
7677          :          Set unused bits in Read Status expd equal rcvd
7678 050450 004737 051160  JSR    PC,T18SETEXP ;SET SOME EXPD TO RECV

```

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```

7679          :      If any transport interface signals are asserted then Print Error
7680 050454 005000      CLR      R0          ;HIGH RECV ADDRESS FOR CKMSG2
7681 050456 012701 051312  MOV     #T18BFR,R1      ;LOW RECV ADDRESS FOR CKMSG2
7682 050462 012702 050556  MOV     #T18EXP,R2      ;EXPD ADDRESS
7683 050466 012703 000012  MOV     #10.,R3        ;NUMBER OF WORDS TO COMPARE
7684 050472 004737 011540  JSR     PC,CKMSG2      ;EXPD EQUAL RECV?
7685 050476          FORCERROR 82$,NOTSSR ;@@D
7686 050506 103404      BCS    90$           ;BR IF YES
7687 050510          NEXT.ERRNO
7688 050510 82$:      ERRHRD  ERRNO,T18CMP,MSGSTAT ;REPORT ERROR
          050510 104456          TRAP    C$ERHRD
          050512 001301          .WORD  705
          050514 051013          .WORD  T18CMP
          050516 012410          .WORD  MSGSTAT
7689 050520 90$:      CKLOOP          ;LOOP ON ERROR, IF FLAG SET
          050520 104406          TRAP    C$CLP1
7690
7691 050522 005737 002222  TST     FATFLG        ;ANY FATAL ERRORS ?
7692 050526 001402      BEQ     160$      ;BRANCH IF NOT
7693 050530 004737 017242  JSR     PC,CKDROP     ;TRY TO DROP THE UNIT
7694 050534 004737 016516  JSR     PC,TSTLOOP    ;DO ITERATIONS?
7695 050540 103002      BCC    165$      ;BR IF NO
7696 050542 000137 050116  JMP     T18LOOP       ;LOOP UNTIL ITERATIONS DONE
7697 050546          165$:      EXIT    TST
7698 050546          TRAP    C$EXIT
          050546 104432          .WORD  L10065-.
          050550 000606
7699
7700
7701          ;*
7702          ;LOCAL STORAGE FOR THIS TEST
7703          ;-
7704
7705 050552          T18MSK:      ;MASK OF UNUSED BITS IN READ STATUS BYTES
7706 050552          .BYTE  †C<000> ;BYTE 0 MASK
          377
7707 050553          .BYTE  †C<340> ;BYTE 1
          037
7708 050554          .BYTE  †C<277> ;BYTE 2
          100
7709 050555          .BYTE  0       ;MAKE IT EVEN
          000
7710
7711 050556          T18EXP:      ;EXPECTED DATA BUFFER
7712 050556          .WORD  0       ;MESSAGE TYPE
          000000
7713 050560          .WORD  0       ;DATA FIELD LENGTH
          000000
7714 050562          .WORD  0       ;RBPCCR
          000000
7715 050564          .WORD  0       ;XST0
          000000
7716 050566          .WORD  0       ;XST1
          000000
7717 050570          .WORD  0       ;XST2
          000000
7718 050572          .WORD  0       ;XST3
          000000
7719 050574          .WORD  0       ;XST4 (ALWAYS PRESENT FOR WRITE SUB)
          000000
7720 050576          .WORD  0       ;READ STATUS BYTE 1/0
          000000
7721 050600          .WORD  0       ;READ STATUS BYTE 2
          000000
7722
7723 050602          T18XS:      .BYTE  377,020 ;READ STATUS BYTE 0/1 EXPECTED BASE
          377      020
7724 050604          .WORD  0       ;READ STATUS BYTE 2 EXPECTED BASE
          000000
7725
7726          ;*
7727          ;LOCAL TEXT MESSAGES FOR TEST
7728          ;-

```

N14

TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```

7729
7730 050606      123      164      141  TST18ID:      .ASCIZ  'Static Transport Bus Interface'
7731 050645      127      122      111  T18SSR: .ASCIZ  'WRITE CHARACTERISTICS Failed'
7732 050702      127      122      111  T182SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Misc) Failed'
7733 050746      127      122      111  T183SSR: .ASCIZ 'WRITE SUBSYSTEM (Read Status) Failed'
7734 051013      124      162      141  T18CMP:  .ASCIZ  'Transport Bus Interface Signals NOT Negated After Unit 7 Selected'
7735
7736
7737
7738
7739
7740 051116
7741 051116
7742 051122      012700   051350
7743 051126      112720   000005
7744 051132      105010
7745 051134      000207
7746
7747
7748
7749
7750 051136
7751 051136
7752 051142      012700   051350
7753 051146      112720   000010
7754 051152      112710   000200
7755 051156      000207
7756
7757
7758
7759
7760
7761 051160
7762 051160      012702   050556
7763 051164      012703   051312
7764 051170      012700   000010
7765 051174      012322
7766 051176      005300
7767 051200      003375
7768 051202      012701   050552
7769 051206      013712   050602
7770 051212      013762   050604   000002
7771 051220      011300
7772 051222      041100
7773 051224      040012
7774 051226      050012
7775 051230      016300   000002
7776 051234      046100   000002
7777 051240      040062   000002
7778 051244      050062   000002
7779 051250      105062   000003
7780 051254      105063   000003
7781 051260      000207
7782
7783
7784      051270
7785
7786
7787

; *
; SETUP T18PK2 PACKET FOR READ STATUS
; -
T18SRD:
  SAVREG      ;SAVE R1-R5 UNTIL NEXT RETURN
  MOV         #T18DT2,R0 ;WRITE SUBSYSTEM DATA BUFFER
  MOV         #PW.RDSTATUS,(R0) ;STORE READ STATUS COMMAND IN BSELO
  CLRB       (R0) ;CLEAR BSEL1
  RTS        PC ;RETURN

; *
; SETUP T18PK2 PACKET FOR WRITE MISC.
; -
T18SMISC:
  SAVREG      ;SAVE R1-R5 UNTIL NEXT RETURN
  MOV         #T18DT2,R0 ;WRITE SUBSYSTEM DATA BUFFER
  MOV         #PW.WMISC,(R0) ;STORE WRITE MISCELLANEOUS IN BSELO
  MOV         #MS.EXT,(R0) ;STORE INVERT EXTENDED FEATURES IN BSEL1
  RTS        PC ;RETURN

; *
; Set first 8 words of expd message buffer = to rcv since not testing
; Set unused bits in Read Status expd equal rcvd
; -
T18SETEXP:
  MOV         #T18EXP,R2 ;GET EXPD
  MOV         #T18BFR,R3 ;GET READ STATUS RECV BUFFER
  MOV         #8,R0 ;SET FIRST 8 WORDS EXP=RCV
  S4: MOV      (R3)+,(R2) ;SET EXPD=RCV
      DEC     R0 ;DONE FIRST 8 WORDS?
      BGT     S4 ;BR IF NO
  MOV         #T18MSK,R1 ;GET UNUSED BIT MASK
  MOV         T18XS,(R2) ;SETUP BASE EXPECTED BYTE 1/0
  MOV         T18XS+2,2(R2) ;SETUP BASE EXPECTED BYTE 2
  MOV         (R3),R0 ;GET RECV BYTE 1 AND BYTE 0
  BIC        (R1),R0 ;CLEAR ALL BUT UNUSED
  BIC        R0,(R2) ;CLEAR UNUSED IN EXP
  BIS        R0,(R2) ;SET UNUSED EXPD=RCV FOR COMPARE
  MOV         2(R3),R0 ;GET RECV BYTE 2
  BIC        2(R1),R0 ;CLEAR ALL BUT UNUSED
  BIC        R0,2(R2) ;CLEAR UNUSED IN EXPD
  BIS        R0,2(R2) ;SET UNUSED EXPD=RCV FOR COMPARE
  CLRB      3(R2) ;CLEAR EXPD BYTE 3 (UNUSED)
  CLRB      3(R3) ;CLEAR RECV BYTE 3 (UNUSED)
  RTS        PC ;RETURN

; *
; WRITE CHARACTERISTICS COMMAND PACKET

```


TEST 7: STATIC TRANSPORT BUS INTERFACE TEST

```

7788
7789 051270          ;T18PACKET:          ;COMMAND PACKET FOR TEST
7790 051270 100004   .WORD 100004   ;WRITE CHARACTERISTICS COMMAND, WITH ACK
7791 051272 051300   .WORD T18DATA   ;ADDRESS OF CHARACTERISTICS BLOCK
7792 051274 000000   .WORD 0          ;
7793 051276 000012   .WORD 10.        ;MESSAGE PACKET MINIMUM SIZE
7794
7795 051300          T18DATA:          ;CHARACTERISTICS DATA BLOCK
7796 051300 051312   .WORD T18BFR     ;ADDRESS OF MESSAGE BUFFER
7797 051302 000000   .WORD 0          ;
7798 051304 000024   .WORD 20.        ;LENGTH OF MESSAGE BUFFER
7799 051306 000000   .WORD 0          ;ESS,ENB,EAI,ERI
7800 051310 000007   .WORD 7          ;SELECT RESERVED UNIT 7
7801
7802
7803 051312          T18BFR:          ;MESSAGE BUFFER
7804 051312 000000   .WORD 0          ;MESSAGE TYPE
7805 051314 000000   .WORD 0          ;DATA FIELD LENGTH
7806 051316 000000   .WORD 0          ;RBPCR
7807 051320 000000   .WORD 0          ;XST0
7808 051322 000000   .WORD 0          ;XST1
7809 051324 000000   .WORD 0          ;XST2
7810 051326 000000   .WORD 0          ;XST3
7811 051330 000000   .WORD 0          ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
7812 051332 000000   .WORD 0          ;READ STATUS BYTE 1/O RETURNED
7813 051334 000000   .WORD 0          ;READ STATUS BYTE 2
7814
7815          ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
7816
7817          ;
7818          ;.=<.10>&177770
7819          ;
7820 051340          T18PK2:          ;WRITE SUBSYSTEM WITH ACK
7821 051340 100006   .WORD P.WRTSUB!P.ACK ;LOW ADDRESS OF DATA BLOCK
7822 051342 051350   .WORD T18DT2     ;HIGH ADDRESS OF DATA BLOCK
7823 051344 000000   .WORD 0          ;BUFFER EXTENT
7824 051346 000010   .WORD 8.        ;
7825
7826 051350          T18DT2:          ;DATA BLOCK
7827 051350          .BYTE 0          ;BSELO
7828 051351          .BYTE 0          ;BSEL1
7829 051352 000000   .WORD 0          ;SEL2
7830 051354 000000   .WORD 0          ;DATA
7831
7832
7833 051356          ENDTST
7834 051356          L10065: TRAP C$ETST
7835 051356 104401
7836          .SBTTL TEST 8: TRANSPORT BUS INTERFACE LOOPBACK TEST
7837
7838          ;**
7839          ; TEST DESCRIPTION:
7840          ;
7841          ; This test verifies the controller's Transport Bus
7842          ; drivers, receivers, and signal loopback logic. Note
7843          ; that the Static Transport Bus test must have run
7844          ; correctly for this test to provide meaningful results.
7845          ;
7846          ; TEST STEPS:
7847          ;

```

TEST 8: TRANSPORT BUS INTERFACE LOOPBACK TEST

```

7845      ; REPEAT FOR LOOPCNT
7846      ; BEGIN
7847      ; Do Subtest 1 - Loopback Control signals test
7848      ; Do Subtest 2 - Loopback Read/Write signals test
7849      ; Do Subtest 3 - Loopback Write Strobe test
7850      ; Do Subtest 4 - Loopback Read Strobe test
7851      ; END
7852      ;--
7853
7854
7855 051360      BGNTST
7856 051360
7860 051360 012700 057572      MOV #TST19ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
7861 051364 004737 016550      JSR PC,TSTSETUP      ;DO INITIAL TEST SETUP
7862 051370 012737 000012 002216      MOV #10.,LOOPCNT      ;PERFORM 10 ITERATIONS
7863 051376      T19LOOP:
7864
7865      .SBTTL TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST
7866
7867      ;**
7868      ; TEST 8: SUBTEST 1:
7869      ;
7870      ; SUBTEST DESCRIPTION:
7871      ;
7872      ; This subtest verifies the Transport Control loopback
7873      ; path can transmit and receive correctly. The
7874      ; control signals are all loopback signals other
7875      ; than the read/write data (IW<7:0> and IR<7:0>).
7876      ;
7877      ; TEST STEPS:
7878      ;
7879      ; The loopback signals IFAD,ITADO,ITAD1 are the tape unit select
7880      ; lines. Since reserved unit 7 must remain selected these signals
7881      ; are always set low. This further means the signals they drive
7882      ; (ISPEED,IRDY,IONL) are only tested in the low state.
7883      ;
7884      ; BEGIN
7885      ; Write to TSSR register to soft initialize the controller
7886      ; Do WRITE CHARACTERISTICS to check for Extended Features Switch
7887      ; If Extended Features Hardware Switch Clear then:
7888      ; Do Write Subsystem Write Miscellaneous to Set Extended Features.
7889      ; Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
7890      ; Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
7891      ; Do Write Subsystem Write Control to CLEAR loopback signals group 1.
7892      ; Do Write Subsystem Write Format to CLEAR loopback signals group 2.
7893      ; (the loopback signals have to be cleared here due to the flip-flops
7894      ; that are set on a 1 to 0 transition (IHER,IFMK,ICER))
7895      ; Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
7896      ; Do a Write Subsystem READ STATUS
7897      ; If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
7898      ; signals NOT=0 Then Print Error
7899      ;
7900      ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
7901      ; BEGIN
7902      ; Do Write Subsystem Write Control to Drive loopback signals group 1.
7903      ; Do Write Subsystem Write Format to Drive loopback signals group 2.
7904      ; Do a Write Subsystem READ STATUS

```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

7905      :      If loopback data NOT= data sent Then Print Error
7906      :      Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
7907      :      Do a Write Subsystem READ STATUS
7908      :      If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
7909      :      signals NOT=0 Then Print Error
7910      :      END
7911      :--
7912 051376      BGNSUB      ;//////////////// BEGIN SUBTEST //////////////////
      051376      T8.1:      TRAP      C#BSUB
      051376 104402
7913
7914      :      Write to TSSR register to soft initialize the controller
7915 051400      5$:
7916 051400 004737 016034      JSR      PC,SOFINIT      ;WRITE TO TSSR TO SOFT INITIALIZE
7917 051404 103405      BCS      10$      ;BR IF SOFT INIT OKAY
7918 051406 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
7919 051410      ERRDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
      051410 104455      TRAP      C#ERDF
      051412 001440      .WORD      800
      051414 003654      .WORD      SFIERR
      051416 012074      .WORD      SFIMSG
7920      :      Do WRITE CHARACTERISTICS to check for Extended Features Switch
7921 051420 005037 002222      10$:      CLR      FATFLG      ;CLEAR FATAL ERROR FLAG
7922 051424 012704 061720      MOV      #T19PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
7923 051430 004737 010662      JSR      PC,WRTCHR      ;DO WRITE CHARACTERISTICS COMMAND
7924 051434      FORCERROR      12$      ;@@DFORCE ERROR IF FORCER=1
7925 051450 103407      BCS      15$      ;BR IF CARRY SET (GOOD RETURN)
7926 051452 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
7927 051454      NEXT.ERRNO
7928 051454      12$:      ERRDF      ERRNO,T19SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      051454 104455      TRAP      C#ERDF
      051456 001441      .WORD      801
      051460 057633      .WORD      T19SSR
      051462 012106      .WORD      PKTSSR
7929 051464 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
7930 051470      15$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      051470 104406      TRAP      C#CLP1
7931      :      If Extended Features Hardware Switch Clear then:
7932      :      Do Write Subsystem Write Miscellaneous to Set Extended Features.
7933 051472 012701 061742      MOV      #T19BFR,R1      ;MESSAGE BUFFER ADDRESS
7934 051476 032761 000200 000012      BIT      #X2.EXTF,XST2(R1)      ;EXTENDED FEATURES SWITCH SET?
7935 051504 001026      BNE      30$      ;BR IF YES
7936 051506 004737 061572      JSR      PC,T19SEXT      ;SETUP PACKET FOR WRITE MISC INVERT
7937 051512 012704 062070      MOV      #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
7938 051516 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
7939 051522 004737 016376      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
7940 051526      FORCERROR      22$      ;@@DFORCE ERROR IF FORCER=1
7941 051542 103407      BCS      30$      ;BR IF CARRY SET (GOOD RETURN)
7942 051544 010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
7943 051546      NEXT.ERRNO
7944 051546      22$:      ERRDF      ERRNO,T192SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      051546 104455      TRAP      C#ERDF
      051550 001442      .WORD      802
      051552 057670      .WORD      T192SSR
      051554 012106      .WORD      PKTSSR
7945 051556 005237 002222      INC      FATFLG      ;SET FATAL ERROR FLAG
7946 051562      30$:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET

```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

051562 104406                                TRAP    C$CLP1
7947      ; Do WRITE CHARACTERISTICS to select reserved unit 7
7948 051564 005037 002222                    CLR    FATFLG                    ;CLEAR FATAL ERROR FLAG
7949 051570 012704 061720                    MOV    #T19PACKET,R4             ;GET THE ADDRESS OF COMMAND PACKET
7950 051574 004737 010662                    JSR    PC,WRTCHR                 ;DO WRITE CHARACTERISTICS COMMAND
7951 051600                                FORCERROR 42$                    ;@DFORCE ERROR IF FORCER=1
7952 051614 103407                    BCS    50$                       ;BR IF CARRY SET (GOOD RETURN)
7953 051616 010001                    MOV    R0,R1                     ;SAVE CONTENTS OF TSSR
7954 051620                                NEXT.ERRNO
7955 051620 42$: ERRDF  ERRNO,T19SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                TRAP    C$ERDF
                                .WORD   803
                                .WORD   T19SSR
                                .WORD   PKTSSR
051620 104455
051622 001443
051624 057633
051626 012106
7956 051630 005237 002222                    INC    FATFLG                    ;SET FATAL ERROR FLAG
7957 051634 50$: CKLOOP                       ;LOOP ON ERROR, IF FLAG SET
                                TRAP    C$CLP1
7958      ; Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
7959 051636 012700 000100                    MOV    #NP.OUT,R0               ;SET TAPE DIRECTION OUT
7960 051642 052700 000040                    BIS    #NP.LOOP,R0              ;SET LOOPBACK ENABLE
7961 051646 004737 061432                    JSR    PC,T19SNPR               ;SETUP T19PK2 FOR WRITE NPR
7962 051652 012704 062070                    MOV    #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
7963 051656 010465 000000                    MOV    R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7964 051662 004737 016376                    JSR    PC,CHKTSSR              ;WAIT FOR SSR TO SET
7965 051666                                FORCERROR 62$                    ;@DFORCE ERROR IF FORCER=1
7966 051702 103407                    BCS    70$                       ;BR IF CARRY SET (GOOD RETURN)
7967 051704 010001                    MOV    R0,R1                     ;SAVE CONTENTS OF TSSR
7968 051706                                NEXT.ERRNO
7969 051706 62$: ERRDF  ERRNO,T194SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                                TRAP    C$ERDF
                                .WORD   804
                                .WORD   T194SSR
                                .WORD   PKTSSR
051706 104455
051710 001444
051712 060001
051714 012106
7970 051716 005237 002222                    INC    FATFLG                    ;SET FATAL ERROR FLAG
7971 051722 70$: CKLOOP                       ;LOOP ON ERROR, IF FLAG SET
                                TRAP    C$CLP1
7972      ; Do Write Subsystem Write Control to CLEAR loopback signals group 1.
7973      ; Do Write Subsystem Write Format to CLEAR loopback signals group 2.
7974      ; (the loopback signals have to be cleared here due to the flip-flops
7975      ; that are set on a 1 to 0 transition (IHER,IFMK,ICER))
7976 051724 005000                                CLR    R0                        ;WRITE 0'S
7977 051726 042700 000200                    BIC    #WC.IFAD,R0              ;IFAD MUST ALWAYS =0
7978 051732 042700 000100                    BIC    #WC.IOTAD,R0            ;ITADO MUST ALWAYS =0
7979 051736 042700 000040                    BIC    #WC.ITAD1,R0            ;ITAD1 MUST ALWAYS =0
7980 051742 004737 061532                    JSR    PC,T19WCTL               ;SETUP PACKET FOR WRITE CONTROL
7981 051746 012704 062070                    MOV    #T19PK2,R4              ;GET WRITE SUBSYSTEM COMMAND PACKET
7982 051752 010465 000000                    MOV    R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
7983 051756 004737 016376                    JSR    PC,CHKTSSR              ;WAIT FOR SSR TO SET
7984 051762                                FORCERROR 82$                    ;@DFORCE ERROR IF FORCER=1
7985 051776 103407                    BCS    90$                       ;BR IF CARRY SET (GOOD RETURN)
7986 052000 010001                    MOV    R0,R1                     ;SAVE CONTENTS OF TSSR
7987 052002                                NEXT.ERRNO
7988 052002 82$: ERRDF  ERRNO,T197SSR,PKTSSR    ;DEVICE FATAL SSR FAILED TO SET
                                TRAP    C$ERDF
                                .WORD   805
                                .WORD   T197SSR
                                .WORD   PKTSSR
052002 104455
052004 001445
052006 060153
052010 012106

```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

7989 052012 005237 002222          INC    FATFLG          ;SET FATAL ERROR FLAG
7990 052016          90$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      052016 104406          TRAP    C$CLP1
7991 052020 005000          CLR    R0              ;SET FORMAT DRIVE DATA=0
7992 052022 004737 061552          JSR   PC,T19WFMT       ;SETUP PACKET FOR WRITE FORMAT
7993 052026 012704 062070          MOV   #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
7994 052032 010465 000000          MOV   R4,TSDB(R5)     ;SET THE PACKET ADDRESS TO EXECUTE
7995 052036 004737 016376          JSR   PC,CHKTSSR      ;WAIT FOR SSR TO SET
7996 052042          FORCERROR 102$      ;@@DFORCE ERROR IF FORCER=1
7997 052056 103407          BCS   110$            ;BR IF CARRY SET (GOOD RETURN)
7998 052060 010001          MOV   R0,R1          ;SAVE CONTENTS OF TSSR
7999 052062          NEXT.ERRNO
8000 052062          102$: ERRDF  ERRNO,T198SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      052062 104455          TRAP    C$ERDF
      052064 001446          .WORD  806
      052066 060222          .WORD  T198SSR
      052070 012106          .WORD  PKTSSR
8001 052072 005237 002222          INC    FATFLG          ;SET FATAL ERROR FLAG
8002 052076          110$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      052076 104406          TRAP    C$CLP1
8003          ; Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
8004 052100 004737 061410          JSR   PC,T19SFIF      ;SETUP PKT FOR WRITE MISC Reset Tape Status F-FLOPS
8005 052104 012704 062070          MOV   #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
8006 052110 010465 000000          MOV   R4,TSDB(R5)     ;SET THE PACKET ADDRESS TO EXECUTE
8007 052114 004737 016376          JSR   PC,CHKTSSR      ;WAIT FOR SSR TO SET
8008 052120          FORCERROR 122$      ;@@DFORCE ERROR IF FORCER=1
8009 052134 103407          BCS   130$            ;BR IF CARRY SET (GOOD RETURN)
8010 052136 010001          MOV   R0,R1          ;SAVE CONTENTS OF TSSR
8011 052140          NEXT.ERRNO
8012 052140          122$: ERRDF  ERRNO,T192SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      052140 104455          TRAP    C$ERDF
      052142 001447          .WORD  807
      052144 057670          .WORD  T192SSR
      052146 012106          .WORD  PKTSSR
8013 052150 005237 002222          INC    FATFLG          ;SET FATAL ERROR FLAG
8014 052154          130$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      052154 104406          TRAP    C$CLP1
8015          ; Do a Write Subsystem READ STATUS
8016          ; If all Tape Status 2 (ICER,IFMK,IHER) flip-flop
8017          ; signals NOT=0 Then Print Error
8018 052156 004737 061370          JSR   PC,T19SRD       ;SETUP PACKET FOR READ STATUS
8019 052162 012704 062070          MOV   #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
8020 052166 010465 000000          MOV   R4,TSDB(R5)     ;SET THE PACKET ADDRESS TO EXECUTE
8021 052172 004737 016376          JSR   PC,CHKTSSR      ;WAIT FOR SSR TO SET
8022 052176          FORCERROR 132$      ;@@DFORCE ERROR IF FORCER=1
8023 052212 103407          BCS   140$            ;BR IF CARRY SET (GOOD RETURN)
8024 052214 010001          MOV   R0,R1          ;SAVE CONTENTS OF TSSR
8025 052216          NEXT.ERRNO
8026 052216          132$: ERRDF  ERRNO,T193SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      052216 104455          TRAP    C$ERDF
      052220 001450          .WORD  808
      052222 057734          .WORD  T193SSR
      052224 012106          .WORD  PKTSSR
8027 052226 005237 002222          INC    FATFLG          ;SET FATAL ERROR FLAG
8028 052232          140$: CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      052232 104406          TRAP    C$CLP1
8029 052234 004737 061630          JSR   PC,T19SETEXP    ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)

```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

8030 052240 012701 057472      MOV      #T19EXSTA,R1      ;GET EXPECTED READ STATUS
8031 052244 012702 061762      MOV      #T19BFSTA,R2     ;GET RECV READ STATUS
8032 052250 011211              MOV      (R2),(R1)        ;SET EXPD WORD #8 = RECV TEMP
8033 052252 042711 002000      BIC      #S1.ICER,(R1)    ;SET EXPD ICER =0
8034 052256 042711 001000      BIC      #S1.IFMK,(R1)    ;SET EXPD IFMK =0
8035 052262 042711 000400      BIC      #S1.IHER,(R1)    ;SET EXPD IHER =0
8036 052266 016261 000002      MOV      2(R2),2(R1)     ;SET EXPD WORD #9 = RECV (NOT TESTING)
8037 052274 005000              CLR      R0                ;HIGH RECV ADDRESS FOR CKMSG2
8038 052276 012701 061742      MOV      #T19BFR,R1      ;LOW RECV ADDRESS FOR CKMSG2
8039 052302 012702 057452      MOV      #T19EXP,R2      ;EXPD ADDRESS
8040 052306 012703 000024      MOV      #20,R3          ;NUMBER OF BYTES TO COMPARE
8041 052312 004737 011540      JSR      PC,CKMSG2        ;EXPD EQUAL RECV?
8042 052316              FORCERROR 152$,NOTSSR    ;@@D
8043 052326 103404              BCS     160$             ;BR IF YES
8044 052330              NEXT.ERRNO
8045 052330 152$:  ERRHRD  ERRNO,T197CMP,MSGLOOP ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD    809
                                .WORD    T197CMP
                                .WORD    MSGLOOP
8046 052340 160$:  CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
8047              ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
8048 052342 005037 057404      CLR      T19PREV         ;INIT 1-0 TRANSITION FLAG
8049 052346 012703 002754      MOV      #TSTBLK,R3      ;GET FIRST PATTERN ADDRESS
8050 052352 012300 200$:  MOV      (R3)+,R0        ;GET A TEST PATTERN
8051 052354 010337 002320      MOV      R3,TSTPTR      ;SAVE POINTER INTO TSTBLK
8052 052360 042700 000200      BIC      #WC.IFAD,R0     ;IFAD MUST ALWAYS =0
8053 052364 042700 000100      BIC      #WC.IOTAD,R0    ;ITADO MUST ALWAYS =0
8054 052370 042700 000040      BIC      #WC.IITAD,R0    ;ITAD1 MUST ALWAYS =0
8055 052374 010037 002314      MOV      R0,DATA        ;SET DATA PATTERN
8056              ; Do Write Subsystem Write Control to Drive loopback signals group 1.
8057              ;@@D CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
8058 052400 013700 002314      MOV      DATA,R0        ;GET TEST PATTERN
8059 052404 004737 061654      JSR      PC,T19CNVT      ;CONVERT PATTERN TO CONTROL DRIVE MASK
8060              ;R0 CONTAINS WRITE CONTROL DATA HERE
8061 052410 004737 061532      JSR      PC,T19WCTL      ;SETUP PACKET FOR WRITE CONTROL
8062 052414 012704 062070      MOV      #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
8063 052420 010465 000000      MOV      R4,TSDB(R5)     ;SET THE PACKET ADDRESS TO EXECUTE
8064 052424 004737 016376      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
8065 052430              FORCERROR 212$          ;@@DFORCE ERROR IF FORCER=1
8066 052444 103407              BCS     220$             ;BR IF CARRY SET (GOOD RETURN)
8067 052446 010001              MOV      R0,R1           ;SAVE CONTENTS OF TSSR
8068 052450              NEXT.ERRNO
8069 052450 212$:  ERRDF  ERRNO,T197SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD    810
                                .WORD    T197SSR
                                .WORD    PKTSSR
8070 052460 005237 002222      INC      FATFLG          ;SET FATAL ERROR FLAG
8071 052464 104406 220$:  CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
8072              ; Do Write Subsystem Write Format to Drive loopback signals group 2.
8073              ;@@D CALL T19CNVT TO SETUP WRITE CONTROL PATTERN
8074              ;
8075 052466 013700 002314      MOV      DATA,R0        ;GET TEST PATTERN
8076 052472 004737 061654      JSR      PC,T19CNVT      ;CONVERT PATTERN TO FORMAT DRIVE MASK

```

H15

SEQ 0189

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

8077 052476 000300 SWAB R0 ;WRITE FORMAT DATA RETURNED IN HIGH BYTE
8078 052500 004737 061552 JSR PC,T19WFMT ;SETUP PACKET FOR WRITE FORMAT
8079 052504 012704 062070 MOV #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8080 052510 010465 000000 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8081 052514 004737 016376 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
8082 052520 FORCERROR 232$ ;@DFORCE ERROR IF FORCER=1
8083 052534 103407 BCS 240$ ;BR IF CARRY SET (GOOD RETURN)
8084 052536 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
8085 052540 NEXT.ERRNO
8086 052540 232$: ERRDF ERRNO,T198SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      052540 104455 TRAP C$ERDF
      052542 001453 .WORD 811
      052544 060222 .WORD T198SSR
      052546 012106 .WORD PKTSSR
8087 052550 005237 002222 INC FATFLG ;SET FATAL ERROR FLAG
8088 052554 240$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
      052554 104406 TRAP C$CLP1
; Do a Write Subsystem READ STATUS
8089 052556 004737 061370 JSR PC,T19SRD ;SETUP PACKET FOR READ STATUS
8090 052562 012704 062070 MOV #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8091 052566 010465 000000 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8092 052572 004737 016376 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
8093 052576 FORCERROR 252$ ;@DFORCE ERROR IF FORCER=1
8094 052576 BCS 260$ ;BR IF CARRY SET (GOOD RETURN)
8095 052612 103407 MOV RO,R1 ;SAVE CONTENTS OF TSSR
8096 052614 010001 NEXT.ERRNO
8097 052616 252$: ERRDF ERRNO,T193SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      052616 104455 TRAP C$ERDF
      052620 001454 .WORD 812
      052622 057734 .WORD T193SSR
      052624 012106 .WORD PKTSSR
8099 052626 005237 002222 INC FATFLG ;SET FATAL ERROR FLAG
8100 052632 260$: CKLOOP ;LOOP ON ERROR, IF FLAG SET
      052632 104406 TRAP C$CLP1
; If loopback data NOT= data sent Then Print Error
8101 052634 004737 061630 JSR PC,T19SETEXP ;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
8102 052640 012701 057472 MOV #T19EXSTA,R1 ;GET EXPECTED READ STATUS
8103 052644 012702 061762 MOV #T19BFSTA,R2 ;GET RCV READ STATUS
8104 052650 013711 002314 MOV DATA,(R1) ;SET EXPD WORD #8 TO TEST DATA FIRST
8105 052654 013700 057404 MOV T19PREV,R0 ;GET PREVIOUS DATA PATTERN
8106 052660 013703 002314 MOV DATA,R3 ;GET CURRENT PATTERN
8107 052664 012704 000400 MOV #S1.IHER,R4 ;SETUP IHER EXPECTED
8108 052670 040411 BIC R4,(R1) ;SET EXPD IHER =0
8109 052672 030400 BIT R4,R0 ;PREVIOUS =1?
8110 052674 001403 BEQ 275$ ;BR IF NO
8111 052676 030403 BIT R4,R3 ;CURRENT =0?
8112 052700 001001 BNE 275$ ;BR IF NO
8113 052702 050411 BIS R4,(R1) ;SET EXPD IHER =1
8114 052704 012704 001000 275$: MOV #S1.IFMK,R4 ;SETUP IFMK EXPECTED
8115 052710 040411 BIC R4,(R1) ;SET EXPD IFMK =0
8116 052712 030400 BIT R4,R0 ;PREVIOUS =1?
8117 052714 001403 BEQ 280$ ;BR IF NO
8118 052716 030403 BIT R4,R3 ;CURRENT =0?
8119 052720 001001 BNE 280$ ;BR IF NO
8120 052722 050411 BIS R4,(R1) ;SET EXPD IFMK =1
8121 052724 012704 002000 280$: MOV #S1.ICER,R4 ;SETUP ICER EXPECTED
8122 052730 040411 BIC R4,(R1) ;SET EXPD ICER =0

```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

8124 052732 030400          BIT      R4,R0          ;PREVIOUS =1?
8125 052734 001403          BEQ      285$          ;BR IF NO
8126 052736 030403          BIT      R4,R3          ;CURRENT =0?
8127 052740 001001          BNE      285$          ;BR IF NO
8128 052742 050411          BIS      R4,(R1)        ;SET EXPD ICER =1
8129 052744 011100          285$:  MOV      (R1),R0    ;GET EXPD WORD
8130                                ;
8131 052746 012704 004000    ; If previous IIDENT=1 and current is IIDENT=1 then EXPD= 0 else 1
8132 052752 050400          MOV      #S1.IIDENT,R4  ;IIDENT
8133 052754 030437 057404    BIS      R4,R0          ;ASSUME EXPD=1
8134 052760 001403          BIT      R4,T19PREV     ;PREVIOUS IIDENT=1?
8135 052762 030403          BEQ      288$          ;BR IF NO
8136 052764 001401          BIT      R4,R3          ;IS CURRENT IIDENT=1?
8137 052766 040400          BEQ      288$          ;BR IF NO
8138 052770 052700 040000    288$:  BIC      R4,R0          ;SET EXPD=0
8139 052774 052700 020000    BIS      #S1.I2RES,R0   ;IRESV2 EXPD ALWAYS=1
8140 053000 042700 100000    BIS      #S1.I1RES,R0   ;IRESV1 EXPD ALWAYS=1
8141 053004 032712 100000    BIC      #S1.PARERR,R0   ;IGNORE PARERR
8142 053010 001402          BIT      #S1.PARERR,(R2) ;IS PARERR SET IN RECV?
8143 053012 052700 100000    BEQ      290$          ;BR IF NO
8144 053016 010011          BIS      #S1.PARERR,R0   ;SET IN EXPD
8145 053020 016261 000002 000002 290$:  MOV      R0,(R1)        ;SETUP FINAL EXPD IN WORD #8
8146 053026 005000          MOV      2(R2),2(R1)    ;SET EXPD WORD #9 = RECV (NOT TESTING)
8147 053030 012701 061742    CLR      R0            ;HIGH RECV ADDRESS FOR CKMSG2
8148 053034 012702 057452    MOV      #T19BFR,R1     ;LOW RECV ADDRESS FOR CKMSG2
8149 053040 012703 000024    MOV      #T19EXP,R2     ;EXPD ADDRESS
8150 053044 004737 011540    MOV      #20.,R3        ;NUMBER OF BYTES TO COMPARE
8151 053050          JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
8152 053060 103404          FORCERROR 302$,NOTSSR   ;@@D
8153 053062          BCS      310$          ;BR IF YES
8154 053062          302$:  ERRHRD  ERRNO,T198CMP,MSGLOOP ;REPORT ERROR
8155 053072          TRAP   C$ERHRD
8156 053072 104406          .WORD  813
8157 053074 004737 061410    .WORD  T198CMP
8158 053100 012704 062070    .WORD  MSGLOOP
8159 053104 010465 000000    .WORD  C$CLP1
8160 053110 004737 016376    TRAP   C$CLP1
8161 053114          ; Do a Write Subsystem Write Misc to Reset Tape Status F-FLOPS
8162 053130 103407          JSR      PC,T19RSFIF    ;SETUP PKT FOR WRITE MISC Reset STATUS
8163 053132 010001          MOV      #T19PK2,R4    ;GET WRITE SUBSYSTEM COMMAND PACKET
8164 053134          MOV      R4,TSDB(R5)   ;SET THE PACKET ADDRESS TO EXECUTE
8165 053134          JSR      PC,CHKTSSR    ;WAIT FOR SSR TO SET
8166 053134          FORCERROR 322$      ;@@DFORCE ERROR IF FORCER=1
8167 053134          BCS      330$          ;BR IF CARRY SET (GOOD RETURN)
8168 053134          MOV      R0,R1        ;SAVE CONTENTS OF TSSR
8169 053134          322$:  NEXT.ERRNO
8170 053134          ERRDF  ERRNO,T192SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
8171 053134          TRAP   C$ERDF
8172 053134          .WORD  814
8173 053134          .WORD  T192SSR
8174 053134          .WORD  PKTSSR
8175 053144 005237 002222    330$:  INC      FATFLG      ;SET FATAL ERROR FLAG
8176 053150          CKLOOP              ;LOOP ON ERROR, IF FLAG SET
8177 053150          TRAP   C$CLP1
8178 053152          ; Do a Write Subsystem READ STATUS
8179 053152 004737 061370    JSR      PC,T19SRD      ;SETUP PACKET FOR READ STATUS
8180 053156 012704 062070    MOV      #T19PK2,R4    ;GET WRITE SUBSYSTEM COMMAND PACKET

```


TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

```

8171 053162 010465 000000      MOV      R4,TSD8(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
8172 053166 004737 016376      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
8173 053172                    FORCERROR 342$          ;@@@FORCE ERROR IF FORCER=1
8174 053206 103407                    BCS     350$          ;BR IF CARRY SET (GOOD RETURN)
8175 053210 010001                    MOV     RO,R1          ;SAVE CONTENTS OF TSSR
8176 053212                    NEXT.ERRNO
8177 053212 342$:  ERRDF  ERRNO,T193SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD    815
                                .WORD    T193SSR
                                .WORD    PKTSSR
8178 053222 005237 002222      INC     FATFLG          ;SET FATAL ERROR FLAG
8179 053226 104406 350$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
8180 053230 004737 061630      JSR     PC,T19SETEXP    ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
8181 053234 012701 057472      MOV     #T19EXSTA,R1   ;GET EXPECTED READ STATUS
8182 053240 012702 061762      MOV     #T198FSTA,R2   ;GET RECV READ STATUS
8183 053244 011211                    MOV     (R2),(R1)      ;SET EXPD WORD #8 = RECV TEMP
8184 053246 042711 002000      BIC     #S1.ICER,(R1)   ;SET EXPD ICER =0
8185 053252 042711 001000      BIC     #S1.IFMK,(R1)  ;SET EXPD IFMK =0
8186 053256 042711 000400      BIC     #S1.IHER,(R1)  ;SET EXPD IHER =0
8187 053262 016261 000002 000002  MOV     2(R2),2(R1)     ;SET EXPD WORD #9 = RECV (NOT TESTING)
8188 053270 005000                    CLR     RO             ;HIGH RECV ADDRESS FOR CKMSG2
8189 053272 012701 061742      MOV     #T198FR,R1     ;LOW RECV ADDRESS FOR CKMSG2
8190 053276 012702 057452      MOV     #T19EXP,R2     ;EXPD ADDRESS
8191 053302 012703 000024      MOV     #20,R3         ;NUMBER OF BYTES TO COMPARE
8192 053306 004737 011540      JSR     PC,CKMSG2      ;EXPD EQUAL RECV?
8193 053312                    FORCERROR 362$,NOTSSR  ;@@@
8194 053322 103404                    BCS     370$          ;BR IF YES
8195 053324                    NEXT.ERRNO
8196 053324 362$:  ERRHRD ERRNO,T197CMP,MSGSTAT ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD    816
                                .WORD    T197CMP
                                .WORD    MSGSTAT
8197 053334 104406 370$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
8198 053336 013737 002314 057404  MOV     DATA,T19PREV  ;SETUP PREVIOUS DATA FOR EXPD CALC.
8200 053344 013703 002320      MOV     TSTPTR,R3      ;RESTORE CURRENT TSTBLK POINTER
8201 053350 020327 003064      CMP     R3,#TBLEND     ;END OF TSTBLK?
8202 053354 103002                    BHS    400$          ;BR IF YES
8203 053356 000137 052352      JMP     200$          ;DO NEXT TSTBLK PATTERN
8204 053362 400$:
8205
8206 053362                    ENDSUB
                                ;////////// END SUBTEST ////////////
                                L10067:
                                TRAP      C$ESUB
8207 053364 005737 002222      TST     FATFLG          ;ANY FATAL ERRORS ?
8208 053370 001402                    BEQ    460$          ;BRANCH IF NOT
8209 053372 004737 017242      JSR     PC,CKDROP      ;TRY TO DROP THE UNIT
8210 053376 460$:
8211
8212
8213
8214
8215

```

TEST 8: SUBTEST 1: LOOPBACK CONTROL SIGNAL TEST

8216
8217
8218
8219
8220
8221
8222
8223
8224
8225
8226
8227
8228
8229
8230
8231
8232
8233
8234
8235
8236
8237
8238
8239
8240
8241
8242
8243
8244
8245
8246
8247
8248
8249
8250
8251
8252
8253
8254
8255
8256
8257
8258
8259
8260
8261
8262
8263
8264
8265
8266

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

TEST 8: SUBTEST 2:
SUBTEST DESCRIPTION:
This subtest verifies the Read/Write data loopback path.
The Read/Write data signals are IR<7:0> and IW<7:0>
respectively.

TEST STEPS:
REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
BEGIN
Write to TSSR register to soft initialize the controller
Do WRITE CHARACTERISTICS to check for Extended Features Switch
If Extended Features Hardware Switch Clear then:
Do Write Subsystem Write Miscellaneous to Set Extended Features.
Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
Do a WRITE NPR to set loopback and tape direction OUT
Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
Do a READ FIFO with tape direction OUT to load tape out write latch
Do a WRITE NPR to set loopback and tape direction IN
Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
to strobe loopback data into FIFO.
Do a READ FIFO with tape direction IN to read data
If Data read from FIFO NOT= to Data sent Then Print Error
Do a Write Subsystem READ STATUS
If Input Ready NOT=1 Then Print Error
If Output Ready NOT=0 Then Print Error
If Data In Miss NOT=0 Then Print Error
END

BGNSUB ;//////////////// BEGIN SUBTEST //////////////////
T8.2: TRAP C#BSUB
Write to TSSR register to soft initialize the controller
5+: JSR PC,SOFINIT ;WRITE TO TSSR TO SOFT INITIALIZE
BCS 10# ;BR IF SOFT INIT OKAY
MOV R0,R1 ;SAVE CONTENTS OF TSSR
ERRDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
TRAP C#ERDF
.WORD 816
.WORD SFIERR
.WORD SFIMSG
10+: Do WRITE CHARACTERISTICS to check for Extended Features Switch
CLR FATFLG ;CLEAR FATAL ERROR FLAG
MOV #T19PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
JSR PC,WRTCHR ;DO WRITE CHARACTERISTICS COMMAND
FORCERROR 12# ;@@DFORCE ERROR IF FORCER=1
BCS 15# ;BR IF CARRY SET (GOOD RETURN)
MOV R0,R1 ;SAVE CONTENTS OF TSSR

053376
053376 104402
053400
053400 004737 016034
053404 103405
053406 010001
053410
053410 104455
053412 001460
053414 003654
053416 012074
053420 005037 002222
053424 012704 061720
053430 004737 010662
053434
053450 103407
053452 010001

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

```

8267 053454          NEXT_ERRNO
8268 053454          124:  ERRDF  ERRNO,T19SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
      053454 104455          TRAP  C4ERDF
      053456 001461          .WORD 817
      053460 057633          .WORD T19SSR
      053462 012106          .WORD PKTSSR
8269 053464 005237 002222          INC  FATFLG  ;SET FATAL ERROR FLAG
8270 053470          154:  CKLOOP  ;LOOP ON ERROR, IF FLAG SET
      053470 104406          TRAP  C4CLP1
8271          ;      If Extended Features Hardware Switch Clear then:
8272          ;      Do Write Subsystem Write Miscellaneous to Set Extended Features.
8273 053472 012701 061742          MOV  @T19BFR,R1  ;MESSAGE BUFFER ADDRESS
8274 053476 032761 000200 000012          BIT  @X2.EXTF,XST2(R1) ;EXTENDED FEATURES SWITCH SET?
8275 053504 001026          BNE  304      ;BR IF YES
8276 053506 004737 061572          JSR  PC,T19SEXT ;SETUP PACKET FOR WRITE MISC INVERT
8277 053512 012704 062070          MOV  @T19PK2,R4  ;GET WRITE SUBSYSTEM COMMAND PACKET
8278 053516 010465 000000          MOV  R4,TSD8(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8279 053522 004737 016376          JSR  PC,CHKTSSR ;WAIT FOR SSR TO SET
8280 053526          FORCERROR 224 ;BADFORCE ERROR IF FORCER=1
8281 053542 103407          BCS  304      ;BR IF CARRY SET (GOOD RETURN)
8282 053544 010001          MOV  R0,R1      ;SAVE CONTENTS OF TSSR
8283 053546          NEXT_ERRNO
8284 053546          224:  ERRDF  ERRNO,T192SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      053546 104455          TRAP  C4ERDF
      053550 001462          .WORD 818
      053552 057670          .WORD T192SSR
      053554 012106          .WORD PKTSSR
8285 053556 005237 002222          INC  FATFLG  ;SET FATAL ERROR FLAG
8286 053562          304:  CKLOOP  ;LOOP ON ERROR, IF FLAG SET
      053562 104406          TRAP  C4CLP1
8287          ;      Do WRITE CHARACTERISTICS to select reserved unit 7
8288 053564 012704 061720          MOV  @T19PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
8289 053570 004737 010662          JSR  PC,WATCHR  ;DO WRITE CHARACTERISTICS COMMAND
8290 053574          FORCERROR 424 ;BADFORCE ERROR IF FORCER=1
8291 053610 103407          BCS  504      ;BR IF CARRY SET (GOOD RETURN)
8292 053612 010001          MOV  R0,R1      ;SAVE CONTENTS OF TSSR
8293 053614          NEXT_ERRNO
8294 053614          424:  ERRDF  ERRNO,T19SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      053614 104455          TRAP  C4ERDF
      053616 001463          .WORD 819
      053620 057633          .WORD T19SSR
      053622 012106          .WORD PKTSSR
8295 053624 005237 002222          INC  FATFLG  ;SET FATAL ERROR FLAG
8296 053630          504:  CKLOOP  ;LOOP ON ERROR, IF FLAG SET
      053630 104406          TRAP  C4CLP1
8297
8298
8299          ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
8300 053632 012703 002754          MOV  @TSTBLK,R3 ;GET FIRST PATTERN ADDRESS
8301 053636 012337 002314          1004: MOV  (R3),DATA ;GET A TEST PATTERN
8302 053642 042737 177400 002314          BIC  @C<377>,DATA ;DATA IS BYTE
8303 053650 010337 002320          MOV  R3,TSTPTR ;SETUP CURRENT TSTBLK POINTER
8304          ;      Do a WRITE NPR to set loopback and tape direction OUT
8305 053654 012700 000100          MOV  @NP.OUT,R0 ;SET TAPE DIRECTION OUT
8306 053660 052700 000040          BIS  @NP.LOOP,R0 ;SET LOOPBACK
8307 053664 004737 061432          JSR  PC,T19SNPR ;SETUP T19PK2 FOR WRITE NPR
8308 053670 012704 062070          MOV  @T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET

```

M15

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

```

8309 053674 010465 000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
8310 053700 004737 016376      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
8311 053704                      FORCERROR 102$           ;@@DFORCE ERROR IF FORCER=1
8312 053720 103407                      BCS     105$           ;BR IF CARRY SET (GOOD RETURN)
8313 053722 010001                      MOV     RO,R1          ;SAVE CONTENTS OF TSSR
8314 053724                      NEXT.ERRNO
8315 053724 102$:  ERRDF  ERRNO,T194SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     820
                                .WORD     T194SSR
                                .WORD     PKTSSR
8316 053734 005237 002222      INC     FATFLG          ;SET FATAL ERROR FLAG
8317 053740 104406 105$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
8318 ; Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
8319 053742 012700 000001      MOV     #1,RO          ;WRITE 1 BYTE
8320 053746 012701 002314      MOV     @DATA,R1       ;FIFO WRITE DATA ADDRESS
8321 053752 004737 061476      JSR     PC,T19WFIF     ;SETUP T19PK2 FOR WRITE FIFO
8322 053756 012704 062070      MOV     #T19PK2,R4     ;GET WRITE SUBSYSTEM COMMAND PACKET
8323 053762 010465 000000      MOV     R4,TSDB(R5)   ;SET THE PACKET ADDRESS TO EXECUTE
8324 053766 004737 016376      JSR     PC,CHKTSSR    ;WAIT FOR SSR TO SET
8325 053772                      FORCERROR 107$           ;@@DFORCE ERROR IF FORCER=1
8326 054006 103407                      BCS     110$           ;BR IF CARRY SET (GOOD RETURN)
8327 054010 010001                      MOV     RO,R1          ;SAVE CONTENTS OF TSSR
8328 054012                      NEXT.ERRNO
8329 054012 107$:  ERRDF  ERRNO,T195SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     821
                                .WORD     T195SSR
                                .WORD     PKTSSR
8330 054022 005237 002222      INC     FATFLG          ;SET FATAL ERROR FLAG
8331 054026 104406 110$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
8332 ; Do a READ FIFO with tape direction OUT to load tape out write latch
8333 054030 012700 000001      MOV     #1,RO          ;SET READ BYTE COUNT
8334 054034 004737 061456      JSR     PC,T19RFIF     ;SETUP T19PK2 FOR READ FIFO
8335 054040 012704 062070      MOV     #T19PK2,R4     ;GET WRITE SUBSYSTEM COMMAND PACKET
8336 054044 010465 000000      MOV     R4,TSDB(R5)   ;SET THE PACKET ADDRESS TO EXECUTE
8337 054050 004737 016376      JSR     PC,CHKTSSR    ;WAIT FOR SSR TO SET
8338 054054                      FORCERROR 122$           ;@@DFORCE ERROR IF FORCER=1
8339 054070 103407                      BCS     130$           ;BR IF CARRY SET (GOOD RETURN)
8340 054072 010001                      MOV     RO,R1          ;SAVE CONTENTS OF TSSR
8341 054074                      NEXT.ERRNO
8342 054074 122$:  ERRDF  ERRNO,T196SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     822
                                .WORD     T196SSR
                                .WORD     PKTSSR
8343 054104 005237 002222      INC     FATFLG          ;SET FATAL ERROR FLAG
8344 054110 104406 130$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
8345 ; Do a WRITE NPR to set loopback and tape direction IN
8346 054112 005000      CLR     RO              ;CLR NP.OUT TO SET TAPE DIRECTION IN
8347 054114 052700 000040      BIS     #NP.LOOP,RO    ;SET LOOPBACK
8348 054120 004737 061432      JSR     PC,T19SNPR     ;SETUP T19PK2 FOR WRITE NPR
8349 054124 012704 062070      MOV     #T19PK2,R4     ;GET WRITE SUBSYSTEM COMMAND PACKET
8350 054130 010465 000000      MOV     R4,TSDB(R5)   ;SET THE PACKET ADDRESS TO EXECUTE

```

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

```

8351 054134 004737 016376      JSR      PC,CHKTSSR          ;WAIT FOR SSR TO SET
8352 054140                    FORCERROR          142$      ;@DFORCE ERROR IF FORCER=1
8353 054154 103407            BCS      150$          ;BR IF CARRY SET (GOOD RETURN)
8354 054156 010001            MOV      RO,R1          ;SAVE CONTENTS OF TSSR
8355 054160                    NEXT,ERRNO
8356 054160 142$:            ERRDF      ERRNO,T194SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     823
                                .WORD     T194SSR
                                .WORD     PKTSSR
8357 054170 005237 002222      INC      FATFLG          ;SET FATAL ERROR FLAG
8358 054174 104406            CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
8359 ; Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
8360 054176 012700 000001      MOV      #1,RO          ;WRITE 1 BYTE
8361 054202 012701 002314      MOV      @DATA,R1       ;FIFO WRITE DATA ADDRESS
8362 054206 004737 061476      JSR      PC,T19WFIF      ;SETUP T19PK2 FOR WRITE FIFO
8363 054212 012704 062070      MOV      @T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
8364 054216 010465 000000      MOV      R4,TSDB(R5)     ;SET THE PACKET ADDRESS TO EXECUTE
8365 054222 004737 016376      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
8366 054226                    FORCERROR          162$      ;@DFORCE ERROR IF FORCER=1
8367 054242 103407            BCS      170$          ;BR IF CARRY SET (GOOD RETURN)
8368 054244 010001            MOV      RO,R1          ;SAVE CONTENTS OF TSSR
8369 054246                    NEXT,ERRNO
8370 054246 162$:            ERRDF      ERRNO,T195SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     824
                                .WORD     T195SSR
                                .WORD     PKTSSR
8371 054256 005237 002222      INC      FATFLG          ;SET FATAL ERROR FLAG
8372 054262 104406            CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
8373 ; Do a READ FIFO with tape direction IN to read data
8374 ; If Data read from FIFO NOT= to Data sent Then Print Error
8375 054264 012700 000001      MOV      #1,RO          ;SET READ BYTE COUNT
8376 054270 004737 061456      JSR      PC,T19RFIF      ;SETUP T19PK2 FOR READ FIFO
8377 054274 012704 062070      MOV      @T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
8378 054300 010465 000000      MOV      R4,TSDB(R5)     ;SET THE PACKET ADDRESS TO EXECUTE
8379 054304 004737 016376      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
8380 054310                    FORCERROR          182$      ;@DFORCE ERROR IF FORCER=1
8381 054324 103407            BCS      190$          ;BR IF CARRY SET (GOOD RETURN)
8382 054326 010001            MOV      RO,R1          ;SAVE CONTENTS OF TSSR
8383 054330                    NEXT,ERRNO
8384 054330 182$:            ERRDF      ERRNO,T196SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     825
                                .WORD     T196SSR
                                .WORD     PKTSSR
8385 054340 005237 002222      INC      FATFLG          ;SET FATAL ERROR FLAG
8386 054344 104406            CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
8387 054346 004737 061630      JSR      PC,T19SETEXP     ;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
8388 054352 012701 057472      MOV      @T19EXSTA,R1    ;GET EXPECTED READ STATUS
8389 054356 012702 061762      MOV      @T19BFSTA,R2    ;GET RCV READ STATUS
8390 054362 013711 002314      MOV      DATA,(R1)      ;SET EXPD WORD #8 = DATA
8391 054366 016261 000002 000002 MOV      2(R2),2(R1)      ;SET EXPD WORD #9 = RCV (NOT TESTING)
8392 054374 005000            CLR      RO              ;HIGH RCV ADDRESS FOR CKMSG2
    
```

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

```

8393 054376 012701 061742      MOV      #T19BFR,R1      ;LOW RECV ADDRESS FOR CKMSG2
8394 054402 012702 057452      MOV      #T19EXP,R2     ;EXPD ADDRESS
8395 054406 012703 000022      MOV      #18.,R3       ;NUMBER OF BYTES TO COMPARE
8396 054412 004737 011540      JSR      PC,CKMSG2     ;EXPD EQUAL RECV?
8397 054416                      FORCERROR 202$,NOTSSR   ;@@D
8398 054426 103404                      BCS      210$         ;BR IF YES
8399 054430                      NEXT.ERRNO
8400 054430 202$:  ERRHRD  ERRNO,T199CMP,MSGSUB ;REPORT ERROR
      054430 104456                      TRAP    C$ERHRD
      054432 001472                      .WORD  826
      054434 061050                      .WORD  T199CMP
      054436 014002                      .WORD  MSGSUB
8401 054440 210$:  CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
      051440 104406                      TRAP    C$CLP1
8402  ; Do a Write Subsystem READ STATUS
8403  ; If Input Ready NOT=1 Then Print Error
8404  ; If Output Ready NOT=0 Then Print Error
8405  ; If Data In Miss NOT=0 Then Print Error
8406 054442 004737 061370      JSR      PC,T19SRD     ;SETUP PACKET FOR READ STATUS
8407 054446 012704 062070      MOV      #T19PK2,R4    ;GET WRITE SUBSYSTEM COMMAND PACKET
8408 054452 010465 000000      MOV      R4,TSDB(R5)   ;SET THE PACKET ADDRESS TO EXECUTE
8409 054456 004737 016376      JSR      PC,CHKTSSR    ;WAIT FOR SSR TO SET
8410 054462                      FORCERROR 212$         ;@@DFORCE ERROR IF FORCER=1
8411 054476 103407                      BCS      220$         ;BR IF CARRY SET (GOOD RETURN)
8412 054500 010001                      MOV      RO,R1         ;SAVE CONTENTS OF TSSR
8413 054502                      NEXT.ERRNO
8414 054502 212$:  ERRDF  ERRNO,T193SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      054502 104455                      TRAP    C$ERDF
      054504 001473                      .WORD  827
      054506 057734                      .WORD  T193SSR
      054510 012106                      .WORD  PKTSSR
8415 054512 005237 002222      220$:  INC      FATFLG ;SET FATAL ERROR FLAG
8416 054516 220$:  CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
      054516 104406                      TRAP    C$CLP1
8417 054520 004737 061630      JSR      PC,T19SETEXP  ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
8418 054524 012701 057472      MOV      #T19EXSTA,R1 ;GET EXPECTED READ STATUS
8419 054530 012702 061762      MOV      #T19BFSTA,R2 ;GET RECV READ STATUS
8420 054534 012221                      MOV      (R2),.(R1)    ;SET EXPD WORD #8 = RECV TEMP
8421 054536 011211                      MOV      (R2),(R1)     ;SET EXPD WORD #9 = RECV TEMP
8422 054540 052711 000020      BIS      #S2.INRDY,(R1) ;SET EXP INPUT READY= 1
8423 054544 042711 000040      BIC      #S2.OUTRDY,(R1) ;SET EXP OUTPUT READY= 0
8424 054550 042711 000200      BIC      #S2.DIM,(R1)  ;SET EXP DATA IN MISS = 0
8425 054554 005000                      CLR      RO            ;HIGH RECV ADDRESS FOR CKMSG2
8426 054556 012701 061742      MOV      #T19BFR,R1   ;LOW RECV ADDRESS FOR CKMSG2
8427 054562 012702 057452      MOV      #T19EXP,R2   ;EXPD ADDRESS
8428 054566 012703 000024      MOV      #20.,R3     ;NUMBER OF BYTES TO COMPARE
8429 054572 004737 011540      JSR      PC,CKMSG2     ;EXPD EQUAL RECV?
8430 054576                      FORCERROR 232$,NOTSSR ;@@D
8431 054606 103404                      BCS      240$         ;BR IF YES
8432 054610                      NEXT.ERRNO
8433 054610 232$:  ERRHRD  ERRNO,T196CMP,MSGSTAT ;REPORT ERROR
      054610 104456                      TRAP    C$ERHRD
      054612 001474                      .WORD  828
      054614 060610                      .WORD  T196CMP
      054616 012410                      .WORD  MSGSTAT
8434 054620 240$:  CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
      054620 104406                      TRAP    C$CLP1

```

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

```

8435
8436
8437
8438 054622          FORCEEXIT          255$          :@aD
8439 054632 013703 002320      MOV      TSTPTR,R3          ;RESTORE CURRENT TSTBLK POINTER
8440 054636 020327 003064      CMP      R3,#TBLEND        ;END OF TSTBLK?
8441 054642 103002          BHIS     255$          ;BR IF YES
8442 054644 000137 053636      JMP      100$             ;DO ANOTHER TSTBLK PATTERN
8443 054650          255$:
8444
8445 054650          ENDSUB                      ;//////////////// END SUBTEST //////////////////
      054650                                     L10070:
      054650 104403                                     TRAP      C$ESUB
8446
8447 054652 005737 002222      TST     FATFLG           ;ANY FATAL ERRORS ?
8448 054656 001402          BEQ     260$             ;BRANCH IF NOT
8449 054660 004737 017242      JSR     PC,CKDROP        ;TRY TO DROP THE UNIT
8450 054664          260$:
8451
8452          .SBTTL TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST
8453
8454
8455      ;**
8456      ; TEST 8: SUBTEST 3:
8457      ;
8458      ; SUBTEST DESCRIPTION:
8459      ;
8460      ; This subtest verifies the Write Strobe loopback path
8461      ; can strobe data from the FIFO to the Data lines.
8462      ; The signal IRESV3 drives IWSTR (write strobe) to write
8463      ; data from the FIFO to the tape data out latch.
8464      ;
8465      ; TEST STEPS:
8466      ;
8467      ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
8468      ; BEGIN
8469      ; Write to TSSR register to soft initialize the controller
8470      ; Do WRITE CHARACTERISTICS to check for Extended Features Switch
8471      ; If Extended Features Hardware Switch Clear then:
8472      ; Do Write Subsystem Write Miscellaneous to Set Extended Features.
8473      ; Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
8474      ; Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
8475      ; Do a WRITE NPR to set loopback and tape direction OUT
8476      ; Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
8477      ; Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
8478      ; Do a WRITE FORMAT to set IRESV3==>IWSTR = 0 to load write data latch
8479      ; Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
8480      ; Do a WRITE NPR to set loopback and tape direction IN
8481      ; Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
8482      ; to strobe loopback data into FIFO.
8483      ; Do a READ FIFO with tape direction IN to read data
8484      ; If Data read from FIFO NOT= to Data sent Then Print Error
8485      ; END
8486      ;--
8487      ;
8488 054664          BGNSUB                      ;//////////////// BEGIN SUBTEST //////////////////
      054664                                     T8.3:

```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```

054664 104402
8489 ; Write to TSSR register to soft initialize the controller TRAP C4BSUB
8490 054666 5$:
8491 054666 004737 016034 JSR PC,SOFINIT ;WRITE TO TSSR TO SOFT INITIALIZE
8492 054672 103405 BCS 10$ ;BR IF SOFT INIT OKAY
8493 054674 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
8494 054676 ERRDF ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
054676 104455 TRAP C4ERDF
054700 001474 .WORD 828
054702 003654 .WORD SFIERR
054704 012074 .WORD SFIMSG
8495 ; Do WRITE CHARACTERISTICS to check for Extended Features Switch
8496 054706 005037 002222 10$: CLR FATFLG ;CLEAR FATAL ERROR FLAG
8497 054712 012704 061720 MOV #T19PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
8498 054716 004737 010662 JSR PC,WRTCHR ;DO WRITE CHARACTERISTICS COMMAND
8499 054722 FORCERROR 12$ ;@@DFORCE ERROR IF FORCER=1
8500 054736 103407 BCS 15$ ;BR IF CARRY SET (GOOD RETURN)
8501 054740 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
8502 054742 NEXT.ERRNO
8503 054742 12$: ERRDF ERRNO,T19SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
054742 104455 TRAP C4ERDF
054744 001475 .WORD 829
054746 057633 .WORD T19SSR
054750 012106 .WORD PKTSSR
8504 054752 005237 002222 15$: INC FATFLG ;SET FATAL ERROR FLAG
8505 054756 CKLOOP ;LOOP ON ERROR, IF FLAG SET TRAP C4CLP1
054756 104406
8506 ; If Extended Features Hardware Switch Clear then:
8507 ; Do Write Subsystem Write Miscellaneous to Set Extended Features.
8508 054760 012701 061742 MOV #T19BFR,R1 ;MESSAGE BUFFER ADDRESS
8509 054764 032761 000200 000012 BIT #X2.EXTF,XST2(R1) ;EXTENDED FEATURES SWITCH SET?
8510 054772 001026 BNE 30$ ;BR IF YES
8511 054774 004737 061572 JSR PC,T19SEXT ;SETUP PACKET FOR WRITE MISC INVERT
8512 055000 012704 062070 MOV #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8513 055004 010465 000000 MOV R4,TSD8(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8514 055010 004737 016376 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
8515 055014 FORCERROR 22$ ;@@DFORCE ERROR IF FORCER=1
8516 055030 103407 BCS 30$ ;BR IF CARRY SET (GOOD RETURN)
8517 055032 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
8518 055034 NEXT.ERRNO
8519 055034 22$: ERRDF ERRNO,T192SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
055034 104455 TRAP C4ERDF
055036 001476 .WORD 830
055040 057670 .WORD T192SSR
055042 012106 .WORD PKTSSR
8520 055044 005237 002222 30$: INC FATFLG ;SET FATAL ERROR FLAG
8521 055050 CKLOOP ;LOOP ON ERROR, IF FLAG SET TRAP C4CLP1
055050 104406
8522 ; Do WRITE CHARACTERISTICS to select reserved unit 7
8523 055052 012704 061720 MOV #T19PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
8524 055056 004737 010662 JSR PC,WRTCHR ;DO WRITE CHARACTERISTICS COMMAND
8525 055062 FORCERROR 42$ ;@@DFORCE ERROR IF FORCER=1
8526 055076 103407 BCS 50$ ;BR IF CARRY SET (GOOD RETURN)
8527 055100 010001 MOV R0,R1 ;SAVE CONTENTS OF TSSR
8528 055102 NEXT.ERRNO
8529 055102 42$: ERRDF ERRNO,T19SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
055102 104455 TRAP C4ERDF

```


TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```

055104 001477                                .WORD 831
055106 057633                                .WORD T19SSR
055110 012106                                .WORD PKTSSR
8530 055112 005237 002222                    50$: INC FATFLG ;SET FATAL ERROR FLAG
8531 055116 104406                            CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                           TRAP C$CLP1
8532
8533 ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
8534 055120 012703 002754                    100$: MOV #TSTBLK,R3 ;GET FIRST PATTERN ADDRESS
8535 055124 012337 002314                    MOV (R3),DATA ;GET A TEST PATTERN
8536 055130 042737 177400 002314            BIC #1C<377>,DATA ;DATA IS BYTE
8537 055136 010337 002320                    MOV R3,TSTPTR ;SETUP CURRENT TSTBLK POINTER
8538 ; Do a WRITE NPR to set loopback and tape direction OUT
8539 055142 012700 000100                    MOV #NP.OUT,R0 ;SET TAPE DIRECTION OUT
8540 055146 052700 000040                    BIS #NP.LOOP,R0 ;SET LOOPBACK
8541 055152 004737 061432                    JSR PC,T19SNPR ;SETUP T19PK2 FOR WRITE NPR
8542 055156 012704 062070                    MOV #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8543 055162 010465 000000                    MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8544 055166 004737 016376                    JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
8545 055172                                FORCERROR 102$ ;@DFORCE ERROR IF FORCER=1
8546 055206 103407                            BCS 105$ ;BR IF CARRY SET (GOOD RETURN)
8547 055210 010001                            MOV R0,R1 ;SAVE CONTENTS OF TSSR
8548 055212                                NEXT.ERRNO
8549 055212 104455                            102$: ERRDF ERRNO,T194SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                           TRAP C$ERDF
055212 104455                                .WORD 832
055214 001500                                .WORD T194SSR
055216 060001                                .WORD PKTSSR
055220 012106
8550 055222 005237 002222                    105$: INC FATFLG ;SET FATAL ERROR FLAG
8551 055226 104406                            CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                           TRAP C$CLP1
8552 ; Do a WRITE FORMAT to set IRESV3=>IWSTR = 1
8553 055230 012700 000002                    MOV #WF.I3RES,R0 ;IRESV3=>IWSTR=1
8554 055234 004737 061552                    JSR PC,T19WFMT ;SETUP T19PK2 FOR WRITE FORMAT
8555 055240 012704 062070                    MOV #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8556 055244 010465 000000                    MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8557 055250 004737 016376                    JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
8558 055254                                FORCERROR 112$ ;@DFORCE ERROR IF FORCER=1
8559 055270 103407                            BCS 120$ ;BR IF CARRY SET (GOOD RETURN)
8560 055272 010001                            MOV R0,R1 ;SAVE CONTENTS OF TSSR
8561 055274                                NEXT.ERRNO
8562 055274 104455                            112$: ERRDF ERRNO,T198SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                           TRAP C$ERDF
055274 104455                                .WORD 833
055276 001501                                .WORD T198SSR
055300 060222                                .WORD PKTSSR
055302 012106
8563 055304 005237 002222                    120$: INC FATFLG ;SET FATAL ERROR FLAG
8564 055310 104406                            CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                           TRAP C$CLP1
8565 ; Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
8566 055312 012700 000001                    MOV #1,R0 ;WRITE 1 BYTE
8567 055316 012701 002314                    MOV #DATA,R1 ;FIFO WRITE DATA ADDRESS
8568 055322 004737 061476                    JSR PC,T19WFIF ;SETUP T19PK2 FOR WRITE FIFO
8569 055326 012704 062070                    MOV #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8570 055332 010465 000000                    MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8571 055336 004737 016376                    JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
8572 055342                                FORCERROR 132$ ;@DFORCE ERROR IF FORCER=1

```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```

8573 055356 103407          BCS      140$          ;BR IF CARRY SET (GOOD RETURN)
8574 055360 010001          MOV      RO,R1        ;SAVE CONTENTS OF TSSR
8575 055362                NEXT,ERRNO
8576 055362                ERRDF   ERRNO,T195SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP   C$ERDF
                                .WORD  834
                                .WORD  T195SSR
                                .WORD  PKTSSR
                                132$:
8577 055372 005237 002222    INC      FATFLG        ;SET FATAL ERROR FLAG
8578 055376 005237 002222    CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
                                TRAP   C$CLP1
8579                ; Do a WRITE FORMAT to set IRESV3==>IWSTR = 0
8580 055400 005000          CLR      RO            ;SET IRESV3==>IWSTR=0
8581 055402 004737 061552    JSR     PC,T19WFMT     ;SETUP T9PK2 FOR WRITE FORMAT
8582 055406 012704 062070    MOV     #T19PK2,R4     ;GET WRITE SUBSYSTEM COMMAND PACKET
8583 055412 010465 000000    MOV     R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
8584 055416 004737 016376    JSR     PC,CHKTSSR     ;WAIT FOR SSR TO SET
8585 055422                FORCERROR 152$          ;@@DFORCE ERROR IF FORCER=1
8586 055436 103407          BCS      160$          ;BR IF CARRY SET (GOOD RETURN)
8587 055440 010001          MOV      RO,R1        ;SAVE CONTENTS OF TSSR
8588 055442                NEXT,ERRNO
8589 055442                ERRDF   ERRNO,T198SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP   C$ERDF
                                .WORD  835
                                .WORD  T198SSR
                                .WORD  PKTSSR
                                152$:
8590 055452 005237 002222    INC      FATFLG        ;SET FATAL ERROR FLAG
8591 055456 005237 002222    CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
                                TRAP   C$CLP1
8592                ; Do a WRITE FORMAT to set IRESV3==>IWSTR = 1
8593 055460 012700 000002    MOV     #WF.I3RES,RO   ;IRESV3==>IWSTR=1
8594 055464 004737 061552    JSR     PC,T19WFMT     ;SETUP T9PK2 FOR WRITE FORMAT
8595 055470 012704 062070    MOV     #T19PK2,R4     ;GET WRITE SUBSYSTEM COMMAND PACKET
8596 055474 010465 000000    MOV     R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
8597 055500 004737 016376    JSR     PC,CHKTSSR     ;WAIT FOR SSR TO SET
8598 055504                FORCERROR 172$          ;@@DFORCE ERROR IF FORCER=1
8599 055520 103407          BCS      180$          ;BR IF CARRY SET (GOOD RETURN)
8600 055522 010001          MOV      RO,R1        ;SAVE CONTENTS OF TSSR
8601 055524                NEXT,ERRNO
8602 055524                ERRDF   ERRNO,T198SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP   C$ERDF
                                .WORD  836
                                .WORD  T198SSR
                                .WORD  PKTSSR
                                172$:
8603 055534 005237 002222    INC      FATFLG        ;SET FATAL ERROR FLAG
8604 055540 005237 002222    CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
                                TRAP   C$CLP1
8605                ;
8606                ; Do a WRITE NPR to set loopback and tape direction IN
8607 055542 005000          CLR      RO            ;CLR NP.OUT TO SET TAPE DIRECTION IN
8608 055544 052700 000040    BIS     #NP.LOOP,RO    ;SET LOOPBACK
8609 055550 004737 061432    JSR     PC,T19SNPR     ;SETUP T19PK2 FOR WRITE NPR
8610 055554 012704 062070    MOV     #T19PK2,R4     ;GET WRITE SUBSYSTEM COMMAND PACKET
8611 055560 010465 000000    MOV     R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
8612 055564 004737 016376    JSR     PC,CHKTSSR     ;WAIT FOR SSR TO SET
8613 055570                FORCERROR 182$          ;@@DFORCE ERROR IF FORCER=1
8614 055604 103407          BCS      190$          ;BR IF CARRY SET (GOOD RETURN)

```

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```

8615 055606 010001      MOV     R0,R1      ;SAVE CONTENTS OF TSSR
8616 055610             NEXT.ERRNO
8617 055610             182$:  ERRDF  ERRNO,T194SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP  C$ERDF
                                .WORD 837
                                .WORD T194SSR
                                .WORD PKTSSR
8618 055620 002222      INC     FATFLG     ;SET FATAL ERROR FLAG
8619 055624 104406      190$:  CKLOOP     ;LOOP ON ERROR, IF FLAG SET
                                TRAP  C$CLP1
8620 ; Do a WRITE FIFO with byte count equal to 1 and Tape direction IN
8621 055626 012700 000001 MOV     #1,R0      ;WRITE 1 BYTE
8622 055632 012701 002314 MOV     #DATA,R1   ;FIFO WRITE DATA ADDRESS
8623 055636 004737 061476 JSR     PC,T19WFIF ;SETUP T19PK2 FOR WRITE FIFO
8624 055642 012704 062070 MOV     #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8625 055646 010465 000000 MOV     R4,TSDB(R5);SET THE PACKET ADDRESS TO EXECUTE
8626 055652 004737 016376 JSR     PC,CHKTSSR ;WAIT FOR SSR TO SET
8627 055656             FORCERROR 202$ ;@@DFORCE ERROR IF FORCER=1
8628 055672 103407      BCS    210$      ;BR IF CARRY SET (GOOD RETURN)
8629 055674 010001      MOV     R0,R1      ;SAVE CONTENTS OF TSSR
8630 055676             NEXT.ERRNO
8631 055676             202$:  ERRDF  ERRNO,T195SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP  C$ERDF
                                .WORD 838
                                .WORD T195SSR
                                .WORD PKTSSR
8632 055706 005237 002222 INC     FATFLG     ;SET FATAL ERROR FLAG
8633 055712 104406      210$:  CKLOOP     ;LOOP ON ERROR, IF FLAG SET
                                TRAP  C$CLP1
8634 ; Do a READ FIFO with tape direction IN to read data
8635 055714 012700 000001 MOV     #1,R0      ;SET READ BYTE COUNT
8636 055720 004737 061456 JSR     PC,T19RFIF ;SETUP T19PK2 FOR READ FIFO
8637 055724 012704 062070 MOV     #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8638 055730 010465 000000 MOV     R4,TSDB(R5);SET THE PACKET ADDRESS TO EXECUTE
8639 055734 004737 016376 JSR     PC,CHKTSSR ;WAIT FOR SSR TO SET
8640 055740             FORCERROR 222$ ;@@DFORCE ERROR IF FORCER=1
8641 055754 103407      BCS    230$      ;BR IF CARRY SET (GOOD RETURN)
8642 055756 010001      MOV     R0,R1      ;SAVE CONTENTS OF TSSR
8643 055760             NEXT.ERRNO
8644 055760             222$:  ERRDF  ERRNO,T196SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP  C$ERDF
                                .WORD 839
                                .WORD T196SSR
                                .WORD PKTSSR
8645 055770 005237 002222 INC     FATFLG     ;SET FATAL ERROR FLAG
8646 055774 104406      230$:  CKLOOP     ;LOOP ON ERROR, IF FLAG SET
                                TRAP  C$CLP1
8647 ; If Data read from FIFO NOT= to Data sent Then Print Error
8648 055776 004737 061630 JSR     PC,T19SETEXP ;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
8649 056002 012701 057472 MOV     #T19EXSTA,R1 ;GET EXPECTED READ STATUS
8650 056006 012702 061762 MOV     #T19BFSTA,R2 ;GET RCV READ STATUS
8651 056012 013711 002314 MOV     DATA,(R1) ;SET EXPD WORD #8 = DATA
8652 056016 016261 000002 000002 MOV     2(R2),2(R1) ;SET EXPD WORD #9 = RCV (NOT TESTING)
8653 056024 005000      CLR     R0          ;HIGH RCV ADDRESS FOR CKMSG2
8654 056026 012701 061742 MOV     #T19BFR,R1 ;LOW RCV ADDRESS FOR CKMSG2
8655 056032 012702 057452 MOV     #T19EXP,R2 ;EXPD ADDRESS
8656 056036 012703 000022 MOV     #18.,R3    ;NUMBER OF BYTES TO COMPARE

```

H16

TEST 8: SUBTEST 3: LOOPBACK WRITE STROBE TEST

```

8657 056042 004737 011540      JSR      PC,CKMSG2      ;EXPD EQUAL RECV?
8658 056046                    FORCERROR 242$,NOTSSR   ;@@D
8659 056056 103404            BCS      250$          ;BR IF YES
P660 056060                    NEXT.ERRNO
8661 056060 242$:            ERRHRD  ERRNO,T19WSTR,MSGSUB ;REPORT ERROR
                                TRAP      C$ERHRD
                                .WORD    840
                                .WORD    T19WSTR
                                .WORD    MSGSUB
8662 056070 104456                    ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
8663 056070 104406                    CKLOOP
8664
8665 056072                    FORCEEXIT 255$          ;@@D
8666 056102 013703 002320      MOV      TSTPTR,R3     ;RESTORE CURRENT TSTBLK POINTER
8667 056106 020327 003064      CMP      R3,#TBLEND   ;END OF TSTBLK?
8668 056112 103002                    BHS      255$          ;BR IF YES
8669 056114 000137 055124      JMP      100$         ;DO ANOTHER TSTBLK PATTERN
8670 056120 255$:
8671
8672 056120                    ENDSUB
                                ;////////// END SUBTEST //////////
                                L10071:
                                TRAP      C$ESUB
8673 056120 104403
8674 056122 005737 002222      TST      FATFLG       ;ANY FATAL ERRORS ?
8675 056126 001402                    BEQ      260$         ;BRANCH IF NOT
8676 056130 004737 017242      JSR      PC,CKDROP    ;TRY TO DROP THE UNIT
8677 056134 260$:
8678 .SBTTL TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST
8679
8680
8681 ;**
8682 ; TEST 8: SUBTEST 4:
8683 ;
8684 ; SUBTEST DESCRIPTION:
8685 ;
8686 ; This subtest verifies the Read Strobe loopback path
8687 ; can strobe the data from the Data lines to the FIFO.
8688 ; The signal IRESV4 drives IRSTR (read strobe) to write
8689 ; from the data lines to the FIFO.
8690 ;
8691 ; TEST STEPS:
8692 ;
8693 ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
8694 ; BEGIN
8695 ; Write to TSSR register to soft initialize the controller
8696 ; Do WRITE CHARACTERISTICS to check for Extended Features Switch
8697 ; If Extended Features Hardware Switch Clear then:
8698 ; Do Write Subsystem Write Miscellaneous to Set Extended Features.
8699 ; Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
8700 ; Do a Write Subsystem WRITE NPR to set tape direction out and Loopback
8701 ; Do a WRITE NPR to set loopback and tape direction OUT
8702 ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
8703 ; Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
8704 ; Do a READ FIFO with tape direction OUT to load tape out write latch
8705 ; Do a WRITE NPR to set loopback and tape direction IN
8706 ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 to write loop data to FIFO

```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

8707      ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
8708      ; (to strobe loopback data into FIFO.)
8709      ; Do a READ FIFO with tape direction IN to read data
8710      ; If Data read from FIFO NOT= to Data sent Then Print Error
8711      ; END
8712      ;--
8713 056134      BGNSUB                      ;////////// BEGIN SUBTEST //////////
      056134                      T8.4:
      056134 104402                      TRAP      C$BSUB
8714      ; Write to TSSR register to soft initialize the controller
8715 056136      5$:
8716 056136 004737 016034      JSR      PC,SOFINIT          ;WRITE TO TSSR TO SOFT INITIALIZE
8717 056142 103405      BCS      10$          ;BR IF SOFT INIT OKAY
8718 056144 010001      MOV      R0,R1          ;SAVE CONTENTS OF TSSR
8719 056146      ERRDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL DURING INIT
      056146 104455                      TRAP      C$ERDF
      056150 001510                      .WORD      840
      056152 003654                      .WORD      SFIERR
      056154 012074                      .WORD      SFIMSG
8720      ; Do WRITE CHARACTERISTICS to check for Extended Features Switch
8721 056156 005037 002222      10$:
8722 056162 012704 061720      CLR      FATFLG          ;CLEAR FATAL ERROR FLAG
8723 056166 004737 010662      MOV      @T19PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
8724 056172      JSR      PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
8725 056206 103407      FORCERROR 12$          ;@DFORCE ERROR IF FORCER=1
8726 056210 010001      BCS      15$          ;BR IF CARRY SET (GOOD RETURN)
8727 056212      MOV      R0,R1          ;SAVE CONTENTS OF TSSR
8728 056212      NEXT,ERRNO
      056212 104455      ERRDF      ERRNO,T19SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      056214 001511                      TRAP      C$ERDF
      056216 057633                      .WORD      841
      056220 012106                      .WORD      T19SSR
      056222 005237 002222                      .WORD      PKTSSR
8729 056222 005237 002222      15$:
8730 056226      INC      FATFLG          ;SET FATAL ERROR FLAG
      056226 104406      CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      056226 104406                      TRAP      C$CLP1
8731      ; If Extended Features Hardware Switch Clear then:
8732      ; Do Write Subsystem Write Miscellaneous to Set Extended Features.
8733 056230 012701 061742      MOV      @T19BFR,R1          ;MESSAGE BUFFER ADDRESS
8734 056234 032761 000200 000012      BIT      @X2.EXTF,XST2(R1)      ;EXTENDED FEATURES SWITCH SET?
8735 056242 001026      BNE      30$          ;BR IF YES
8736 056244 004737 061572      JSR      PC,T19SEXT          ;SETUP PACKET FOR WRITE MISC INVERT
8737 056250 012704 062070      MOV      @T19PK2,R4          ;GET WRITE SUBSYSTEM COMMAND PACKET
8738 056254 010465 000000      MOV      R4,TSDB(R5)          ;SET THE PACKET ADDRESS TO EXECUTE
8739 056260 004737 016376      JSR      PC,CHKTSSR          ;WAIT FOR SSR TO SET
8740 056264      FORCERROR 22$          ;@DFORCE ERROR IF FORCER=1
8741 056300 103407      BCS      30$          ;BR IF CARRY SET (GOOD RETURN)
8742 056302 010001      MOV      R0,R1          ;SAVE CONTENTS OF TSSR
8743 056304      NEXT,ERRNO
8744 056304      ERRDF      ERRNO,T192SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      056304 104455                      TRAP      C$ERDF
      056306 001512                      .WORD      842
      056310 057670                      .WORD      T192SSR
      056312 012106                      .WORD      PKTSSR
8745 056314 005237 002222      30$:
8746 056320      INC      FATFLG          ;SET FATAL ERROR FLAG
      056320 104406      CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      056320 104406                      TRAP      C$CLP1
8747      ; Do WRITE CHARACTERISTICS to select reserved unit 7

```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

8748 056322 012704 061720      MOV      #T19PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
8749 056326 004737 010662      JSR      PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
8750 056332                FORCERROR 42$              ;@DFORCE ERROR IF FORCER=1
8751 056346 103407                BCS     50$                ;BR IF CARRY SET (GOOD RETURN)
8752 056350 010001                MOV     RO,R1              ;SAVE CONTENTS OF TSSR
8753 056352                NEXT.ERRNO
8754 056352 42$:  ERRDF  ERRNO,T19SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     843
                                .WORD     T19SSR
                                .WORD     PKTSSR
                                056352 104455
                                056354 001513
                                056356 057633
                                056360 012106
8755 056362 005237 002222      INC     FATFLG             ;SET FATAL ERROR FLAG
8756 056366 50$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                056366 104406
8757
8758 ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
8759 056370 012703 002754      MOV     #TSTBLK,R3        ;GET FIRST PATTERN ADDRESS
8760 056374 012337 002314 100$:  MOV     (R3)+,DATA        ;GET A TEST PATTERN
8761 056400 042737 177400 002314 BIC     #C<377>,DATA      ;DATA IS BYTE
8762 056406 010337 002320      MOV     R3,TSTPTR        ;SETUP CURRENT TSTBLK POINTER
8763 ; Do a WRITE NPR to set loopback and tape direction OUT
8764 056412 012700 000100      MOV     #NP.OUT,RO       ;SET TAPE DIRECTION OUT
8765 056416 052700 000040      BIS     #NP.LOOP,RO      ;SET LOOPBACK
8766 056422 004737 061432      JSR     PC,T19SNPR       ;SETUP T19PK2 FOR WRITE NPR
8767 056426 012704 062070      MOV     #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
8768 056432 010465 000000      MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS TO EXECUTE
8769 056436 004737 016376      JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
8770 056442                FORCERROR 102$            ;@DFORCE ERROR IF FORCER=1
8771 056456 103407                BCS     105$              ;BR IF CARRY SET (GOOD RETURN)
8772 056460 010001                MOV     RO,R1              ;SAVE CONTENTS OF TSSR
8773 056462                NEXT.ERRNO
8774 056462 102$:  ERRDF  ERRNO,T194SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     844
                                .WORD     T194SSR
                                .WORD     PKTSSR
                                056462 104455
                                056464 001514
                                056466 060001
                                056470 012106
8775 056472 005237 002222      INC     FATFLG             ;SET FATAL ERROR FLAG
8776 056476 105$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                056476 104406
8777 ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
8778 056500 012700 000001      MOV     #WF.I4RES,RO     ;IRESV4==>IRSTR=1
8779 056504 004737 061552      JSR     PC,T19WFMT      ;SETUP T9PK2 FOR WRITE FORMAT
8780 056510 012704 062070      MOV     #T19PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
8781 056514 010465 000000      MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS TO EXECUTE
8782 056520 004737 016376      JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
8783 056524                FORCERROR 112$            ;@DFORCE ERROR IF FORCER=1
8784 056540 103407                BCS     120$              ;BR IF CARRY SET (GOOD RETURN)
8785 056542 010001                MOV     RO,R1              ;SAVE CONTENTS OF TSSR
8786 056544                NEXT.ERRNO
8787 056544 112$:  ERRDF  ERRNO,T198SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     845
                                .WORD     T198SSR
                                .WORD     PKTSSR
                                056544 104455
                                056546 001515
                                056550 060222
                                056552 012106
8788 056554 005237 002222      INC     FATFLG             ;SET FATAL ERROR FLAG
8789 056560 120$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                056560 104406

```

K16

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

8790 ; Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
8791 056562 012700 000001 MOV #1,R0 ;WRITE 1 BYTE
8792 056566 012701 002314 MOV #DATA,R1 ;FIFO WRITE DATA ADDRESS
8793 056572 004737 061476 JSR PC,T19WFIF ;SETUP T19PK2 FOR WRITE FIFO
8794 056576 012704 062070 MOV #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8795 056602 010465 000000 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8796 056606 004737 016376 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
8797 056612 FORCERROR 132$ ;@DFORCE ERROR IF FORCER=1
8798 056626 103407 BCS 140$ ;BR IF CARRY SET (GOOD RETURN)
8799 056630 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
8800 056632 NEXT.ERRNO
8801 056632 132$: ERRDF ERRNO,T195SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      TRAP C$ERDF
      .WORD 846
      .WORD T195SSR
      .WORD PKTSSR
8802 056642 005237 002222 INC FATFLG ;SET FATAL ERROR FLAG
8803 056646 104406 CKLOOP ;LOOP ON ERROR, IF FLAG SET
      TRAP C$CLP1
8804 ; Do a READ FIFO with tape direction OUT to load tape out write latch
8805 056650 012700 000001 MOV #1,R0 ;SET READ BYTE COUNT
8806 056654 004737 061456 JSR PC,T19RFIF ;SETUP T19PK2 FOR READ FIFO
8807 056660 012704 062070 MOV #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8808 056664 010465 000000 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8809 056670 004737 016376 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
8810 056674 FORCERROR 152$ ;@DFORCE ERROR IF FORCER=1
8811 056710 103407 BCS 160$ ;BR IF CARRY SET (GOOD RETURN)
8812 056712 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
8813 056714 NEXT.ERRNO
8814 056714 152$: ERRDF ERRNO,T196SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      TRAP C$ERDF
      .WORD 847
      .WORD T196SSR
      .WORD PKTSSR
8815 056724 005237 002222 INC FATFLG ;SET FATAL ERROR FLAG
8816 056730 104406 CKLOOP ;LOOP ON ERROR, IF FLAG SET
      TRAP C$CLP1
8817 ; Do a WRITE NPR to set loopback and tape direction IN
8818 056732 005000 CLR RO ;CLR NP.OUT TO SET TAPE DIRECTION IN
8819 056734 052700 000040 BIS #NP.LOOP,RO ;SET LOOPBACK
8820 056740 004737 061432 JSR PC,T19SNPR ;SETUP T19PK2 FOR WRITE NPR
8821 056744 012704 062070 MOV #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8822 056750 010465 000000 MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8823 056754 004737 016376 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
8824 056760 FORCERROR 182$ ;@DFORCE ERROR IF FORCER=1
8825 056774 103407 BCS 190$ ;BR IF CARRY SET (GOOD RETURN)
8826 056776 010001 MOV RO,R1 ;SAVE CONTENTS OF TSSR
8827 057000 NEXT.ERRNO
8828 057000 182$: ERRDF ERRNO,T194SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      TRAP C$ERDF
      .WORD 848
      .WORD T194SSR
      .WORD PKTSSR
8829 057010 005237 002222 INC FATFLG ;SET FATAL ERROR FLAG
8830 057014 104406 CKLOOP ;LOOP ON ERROR, IF FLAG SET
      TRAP C$CLP1
8831 ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 0

```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

8832 057016 005000          CLR      RO          ;SET IRESV4==>IRSTR=0
8833 057020 004737 061552  JSR      PC,T19WFMT ;SETUP T9PK2 FOR WRITE FORMAT
8834 057024 012704 062070  MOV      #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8835 057030 010465 000000  MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8836 057034 004737 016376  JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
8837 057040          FORCERROR 202$ ;@DFORCE ERROR IF FORCER=1
8838 057054 103407          BCS     210$ ;BR IF CARRY SET (GOOD RETURN)
8839 057056 010001          MOV      RO,R1 ;SAVE CONTENTS OF TSSR
8840 057060          NEXT.ERRNO
8841 057060 202$:  ERRDF  ERRNO,T198SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     849
                                .WORD     T198SSR
                                .WORD     PKTSSR
8842 057070 005237 002222          INC      FATFLG ;SET FATAL ERROR FLAG
8843 057074 104406          CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
;
; Do a WRITE FORMAT to set IRESV4==>IRSTR = 1
8845 057076 012700 000001  MOV      #WF.I4RES,RO ;IRESV4==>IRSTR=1
8846 057102 004737 061552  JSR      PC,T19WFMT ;SETUP T9PK2 FOR WRITE FORMAT
8847 057106 012704 062070  MOV      #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8848 057112 010465 000000  MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8849 057116 004737 016376  JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
8850 057122          FORCERROR 222$ ;@DFORCE ERROR IF FORCER=1
8851 057136 103407          BCS     230$ ;BR IF CARRY SET (GOOD RETURN)
8852 057140 010001          MOV      RO,R1 ;SAVE CONTENTS OF TSSR
8853 057142          NEXT.ERRNO
8854 057142 222$:  ERRDF  ERRNO,T198SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     850
                                .WORD     T198SSR
                                .WORD     PKTSSR
8855 057152 005237 002222          INC      FATFLG ;SET FATAL ERROR FLAG
8856 057156 104406          CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
;
; Do a READ FIFO with tape direction IN to read data
8858 057160 012700 000001  MOV      #1,RO ;SET READ BYTE COUNT
8859 057164 004737 061456  JSR      PC,T19RFIF ;SETUP T19PK2 FOR READ FIFO
8860 057170 012704 062070  MOV      #T19PK2,R4 ;GET WRITE SUBSYSTEM COMMAND PACKET
8861 057174 010465 000000  MOV      R4,TSDB(R5) ;SET THE PACKET ADDRESS TO EXECUTE
8862 057200 004737 016376  JSR      PC,CHKTSSR ;WAIT FOR SSR TO SET
8863 057204          FORCERROR 282$ ;@DFORCE ERROR IF FORCER=1
8864 057220 103407          BCS     290$ ;BR IF CARRY SET (GOOD RETURN)
8865 057222 010001          MOV      RO,R1 ;SAVE CONTENTS OF TSSR
8866 057224          NEXT.ERRNO
8867 057224 282$:  ERRDF  ERRNO,T196SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C$ERDF
                                .WORD     851
                                .WORD     T196SSR
                                .WORD     PKTSSR
8868 057234 005237 002222          INC      FATFLG ;SET FATAL ERROR FLAG
8869 057240 104406          CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
;
; If Data read from FIFO NOT= to Data sent Then Print Error
8871 057242 004737 061630  JSR      PC,T19SETEXP ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
8872 057246 012701 057472  MOV      #T19EXSTA,R1 ;GET EXPECTED READ STATUS
8873 057252 012702 061762  MOV      #T19BFSTA,R2 ;GET RECV READ STATUS

```


M16

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

8874 057256 013711 002314      MOV      DATA,(R1)          ;SET EXPD WORD #8 = DATA
8875 057262 016261 000002 000002  MOV      2(R2),2(R1)        ;SET EXPD WORD #9 = RECV (NOT TESTING)
8876 057270 005000              CLR      R0                 ;HIGH RECV ADDRESS FOR CKMSG2
8877 057272 012701 061742      MOV      #T19BFR,R1        ;LOW RECV ADDRESS FOR CKMSG2
8878 057276 012702 057452      MOV      #T19EXP,R2       ;EXPD ADDRESS
8879 057302 012703 000022      MOV      #18.,R3          ;NUMBER OF BYTES TO COMPARE
8880 057306 004737 011540      JSR      PC,CKMSG2        ;EXPD EQUAL RECV?
8881 057312              FORCERROR 302$,NOTSSR      ;@@D
8882 057322 103404              BCS     310$              ;BR IF YES
8883 057324              NEXT,ERRNO
8884 057324 302$:  ERRHRD  ERRNO,T19RSTR,MSGSUB ;REPORT ERROR
      057324 104456              TRAP   C$ERHRD
      057326 001524              .WORD 852
      057330 061240              .WORD T19RSTR
      057332 014002              .WORD MSGSUB
8885 057334 310$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      057334 104406              TRAP  C$CLP1
8886
8887
8888 057336              FORCEXIT 355$             ;@@D
8889 057346 013703 002320      MOV      TSTPTR,R3        ;RESTORE CURRENT TSTBLK POINTER
8890 057352 020327 003064      CMP      R3,#TBLEND      ;END OF TSTBLK?
8891 057356 103002              BHIS    355$             ;BR IF YES
8892 057360 000137 056374      JMP      100$            ;DO ANOTHER TSTBLK PATTERN
8893 057364 355$:
8894
8895 057364              ENDSUB                  ;////////// END SUBTEST //////////
      057364              L10072:
      057364 104403              TRAP   C$ESUB
8896
8897 057366 005737 002222      TST      FATFLG          ;ANY FATAL ERRORS ?
8898 057372 001402              BEQ     360$            ;BRANCH IF NOT
8899 057374 004737 017242      JSR      PC,CKDROP       ;TRY TO DROP THE UNIT
8900 057400 360$:
8901
8902 057400              EXIT  TST              ;////////// EXIT TEST //////////
      057400 104432              TRAP   C$EXIT
      057402 002602              .WORD L10066-.
8903
8904
8905
8906 ;*
8907 ;LOCAL STORAGE FOR THIS TEST
8908 ;-
8909 057404 000000      T19PREV: .WORD 0          ;DRIVE SIGNAL 1-0 TRANSITION FLAG
8910 ;*
8911 ; LOOPBACK DRIVE SIGNAL TABLE
8912 ; THIS TABLE IS USED BY T19CNVT TO SETUP
8913 ; A DRIVE PATTERN FROM THE TEST DATA INPUT PATTERN.
8914 ;
8915 ; WRITE CONTROL SIGNALS ARE OF FORM WC.XXX
8916 ; WRITE FORMAT SIGNALS ARE OF FORM WF.XXXX
8917 ;-
8918 057406      T19BCTL:                ;WRITE CONTROL DRIVE SIGNALS
8919 057406 000001      WC.IGO                ;IGO==>IFPT DATA<0>
8920 057410 000002      WC.IFEN               ;IFEN==>IFBY DATA<1>
8921 057412 000004      WC.IRWU               ;IRWU==>IRWD DATA<2>

```

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

8922 057414 000010 WC.IREW ;IREW==>IDBY DATA<3>
8923 057416 002000 WF.IERASE*256. ;IFAD==>ILDLP DATA<4>
8924 057420 000040 WC.IITAD ;ITAD1==>IONL DATA<5>
8925 057422 000100 WC.IOTAD ;ITADO==>IRDY DATA<6>
8926 057424 000200 WC.IFAD ;IERASE==>ISPEED DATA<7>
8927 057426 004000 WF.IEDIT*256. ;IEDIT==>IHER DATA<8>
8928 057430 010000 WF.IWFM*256. ;IWFM==>IFMK DATA<9>
8929 057432 020000 WF.IREV*256. ;IREV==>ICER DATA<10>
8930 057434 040000 WF.IWRT*256. ;IWRT==>IIDENT DATA<11>
8931 057436 100000 WF.IHISP*256. ;IHISP==>IEOT DATA<12>
8932 057440 000000 .WORD 0 ;IRESV2 (UNUSED)DATA<13>
8933 057442 000000 .WORD 0 ;IRESV1 (UNUSED)DATA<14>
8934 057444 000000 .WORD 0 ;PARERR (UNTESTED)DATA<15>
8935
8936 057446 T19MSK: ;MASK OF UNTESTED BITS IN READ STATUS BYTES
8937 ;UNTESTED BITS ARE SET TO 1
8938 057446 377 .BYTE †C<000> ;BYTE 0 MASK
8939 057447 037 .BYTE †C<340> ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
8940 057450 360 .BYTE †C<017> ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
8941 057451 000 .BYTE 0 ;MAKE IT EVEN
8942
8943 057452 T19EXP: ;BEGIN EXPECTED DATA BUFFER
8944 057452 000000 .WORD 0 ;MESSAGE TYPE
8945 057454 000000 .WORD 0 ;DATA FIELD LENGTH
8946 057456 000000 .WORD 0 ;RBPCCR
8947 057460 000000 .WORD 0 ;XST0
8948 057462 000000 .WORD 0 ;XST1
8949 057464 000000 .WORD 0 ;XST2
8950 057466 000000 .WORD 0 ;XST3
8951 057470 000000 .WORD 0 ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
8952 057472 T19EXSTA: .BLKB 64. ;EXPECTED READ STATUS AND WRITE FIFO DATA
8953 057572 T19EXEND: ;END EXPECTED DATA BUFFER
8954
8955 ;*
8956 ;LOCAL TEXT MESSAGES FOR TEST
8957 ;-
8958 057572 124 162 141 TST19ID: .ASCIZ 'Transport Bus Interface Loopback'
8959 057633 127 122 111 T19SSR: .ASCIZ 'WRITE CHARACTERISTICS Failed'
8960 057670 127 122 111 T192SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Misc) Failed'
8961 057734 127 122 111 T193SSR: .ASCIZ 'WRITE SUBSYSTEM (Read Status) Failed'
8962 060001 127 122 111 T194SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Npr) Failed'
8963 060044 127 122 111 T195SSR: .ASCIZ 'WRITE SUBSYSTEM (Write FIFO) Failed'
8964 060110 127 122 111 T196SSR: .ASCIZ 'WRITE SUBSYSTEM (Read FIFO) Failed'
8965 060153 127 122 111 T197SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Control) Failed'
8966 060222 127 122 111 T198SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Format) Failed'
8967 060270 106 111 106 T191CMP: .ASCIZ 'FIFO Status in WORD #9 Incorrect after Initialize'
8968 060352 122 145 141 T192CMP: .ASCIZ 'Read FIFO Data not equal to Write FIFO , Data is in WORD #8'
8969 060446 124 141 160 T193CMP: .ASCIZ 'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
8970 060534 122 145 141 T195CMP: .ASCIZ 'Read FIFO Data not equal to Write FIFO Data'
8971 060610 106 111 106 T196CMP: .ASCIZ 'FIFO Status (in WORD #9) Incorrect after READ FIFO'
8972 060673 124 141 160 T197CMP: .ASCIZ 'Tape Status 2 (in WORD #8) Incorrect after RESET TAPE'
8973 060761 103 157 156 T198CMP: .ASCIZ 'Control Signal Loopback Data Error, Data is in WORD #8'
8974 061050 122 145 141 T199CMP: .ASCIZ 'Read/Write Loopback Data Error, Data is in WORD #8'
8975 061133 114 157 157 T19WSTR: .ASCIZ 'Loopback Data Error when strobed by Write strobe, Data is in WORD #8'
8976 061240 114 157 157 T19RSTR: .ASCIZ 'Loopback Data Error when strobed by Read Strobe, Data is in WORD #8'
8977
8978 .EVEN

```

C1

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

8979
8980
8981      ;* CLEAR MESSAGE BUFFER
8982      ;-
8983 061344 T19CLRBUF:
8984 061344      SAVREG
8985 061350      MOV      #T19BFR,R1      ;SAVE R1-R5 UNTIL NEXT RETURN
8986 061354 012701 061742      MOV      #T19BEND-T19BFR,R2 ;GET MESSAGE BUFFER ADDRESS
8987 061360 105021      CLR      (R1)+      ;SIZE OF MESSAGE BUFFER IN BYTES
8988 061362 005302      DEC      R2      ;CLEAR A BYTE
8989 061364 003375      BGT      104      ;DONE?
8990 061366 000207      RTS      PC      ;BR IF NO
8991      ;RETURN
8992
8993      ;* SETUP T19PK2 PACKET FOR READ STATUS
8994      ;-
8995 061370 T19SRD:
8996 061370 004737 061344      JSR      PC,T19CLRBUF      ;CLEAR MESSAGE BUFFER
8997 061374 012700 062100      MOV      #T19DT2,R0      ;WRITE SUBSYSTEM DATA BUFFER
8998 061400 112720 000005      MOVB   #PW.RDSTATUS,(R0)+ ;STORE READ STATUS COMMAND IN BSELO
8999 061404 105010      CLR      (R0)      ;CLEAR BSEL1
9000 061406 000207      RTS      PC      ;RETURN
9001
9002      ;* SETUP T19PK2 PACKET FOR WRITE MISC Reset Tape Status F-FLOPS
9003      ;-
9004      ;-
9005 061410 T19RSFIF:
9006 061410 004737 061344      JSR      PC,T19CLRBUF      ;CLEAR MESSAGE BUFFER
9007 061414 012700 062100      MOV      #T19DT2,R0      ;WRITE SUBSYSTEM DATA BUFFER
9008 061420 112720 000010      MOVB   #PW.WMISC,(R0)+ ;STORE WRITE MISCELLANEOUS IN BSELO
9009 061424 112710 000030      MOVB   #MS.RSFIF!MS.RSTAP,(R0) ;STORE BSEL1 CLEAR FIFO CODES
9010 061430 000207      RTS      PC      ;RETURN
9011
9012      ;* SETUP T19PK2 PACKET FOR WRITE NPR
9013      ;-
9014      ;-
9015      ; INPUT:
9016      ; RO CONTAINS BSEL1 NPR DATA
9017      ;
9018      ; SETS NP.WRP SINCE IF 0 IT WRITES WRONG PARITY.
9019      ;-
9020 061432 T19SNPR:
9021 061432 004737 061344      JSR      PC,T19CLRBUF      ;CLEAR MESSAGE BUFFER
9022 061436 012701 062100      MOV      #T19DT2,R1      ;WRITE SUBSYSTEM DATA BUFFER
9023 061442 112721 000011      MOVB   #PW.WNPR,(R1)+ ;STORE WRITE NPR IN BSELO
9024 061446 052700 000020      BIS      #NP.WRP,R0      ;DON'T WRITE WRONG PARITY
9025 061452 110011      MOVB   R0,(R1)      ;STORE NPR DATA IN BSEL1
9026 061454 000207      RTS      PC      ;RETURN
9027
9028      ;* SETUP T19PK2 PACKET FOR READ FIFO
9029      ;-
9030      ;-
9031      ; INPUT:
9032      ; RO CONTAINS SEL2 BYTE COUNT
9033      ;-
9034 061456 T19RFIF:
9035 061456 004737 061344      JSR      PC,T19CLRBUF      ;CLEAR MESSAGE BUFFER

```

D1

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

9036 061462 012701 062100      MOV      #T19DT2,R1          ;WRITE SUBSYSTEM DATA BUFFER
9037 061466 112721 000003      MOVB    #PW.RFIFO,(R1)+     ;STORE READ FIFO IN BSELO
9038 061472 110021              MOVB    RO,(R1)+           ;STORE BYTE COUNT IN BSEL1
9039 061474 000207              RTS      PC                ;RETURN
9040
9041      ;*
9042      ; SETUP T19PK2 PACKET FOR WRITE FIFO
9043      ;
9044      ; INPUT:
9045      ;   RO CONTAINS BYTE COUNT
9046      ;   R1 CONTAINS DATA PATTERN BLOCK ADDRESS
9047      ;-
9047 061476      T19WFIF:
9048 061476      SAVREG                    ;SAVE R1-R5 UNTIL NEXT RETURN
9049 061502 004737 061344      JSR     PC,T19CLRBUF        ;CLEAR MESSAGE BUFFER
9050 061506 012702 062100      MOV     #T19DT2,R2         ;WRITE SUBSYSTEM DATA BUFFER
9051 061512 112722 000004      MOVB   #PW.WFIFO,(R2)+     ;STORE WRITE FIFO IN BSELO
9052 061516 110022              MOVB   RO,(R2)+           ;STORE BYTE COUNT IN BSEL1
9053 061520 005022              CLR    (R2)+              ;CLEAR SEL2 (UNUSED)
9054 061522 112122      10$:   MOVB   (R1)+,(R2)+       ;STORE DATA PATTERN BYTE
9055 061524 005300              DEC    RO                 ;DONE ALL BYTES?
9056 061526 003375              BGT    10$                ;BR IF NO
9057 061530 000207              RTS      PC                ;RETURN
9058
9059      ;*
9059      ; SETUP T19PK2 FOR WRITE CONTROL
9060      ;
9061      ; INPUT:
9062      ;   RO CONTAINS DRIVING DATA PATTERN
9063      ;-
9064 061532      T19WCTL:
9065 061532 004737 061344      JSR     PC,T19CLRBUF        ;CLEAR MESSAGE BUFFER
9066 061536 012701 062100      MOV     #T19DT2,R1         ;WRITE SUBSYSTEM DATA BUFFER
9067 061542 112721 000006      MOVB   #PW.WCTL,(R1)+     ;STORE WRITE CONTROL IN BSELO
9068 061546 110021              MOVB   RO,(R1)+           ;STORE DATA WORD IN BSEL1
9069 061550 000207              RTS      PC                ;RETURN
9070
9071      ;*
9071      ; SETUP T19PK2 FOR WRITE FORMAT TRANSPORT REGISTER
9072      ;
9073      ; INPUT:
9074      ;   RO CONTAINS DRIVING DATA PATTERN
9075      ;-
9076 061552      T19WFMT:
9077 061552 004737 061344      JSR     PC,T19CLRBUF        ;CLEAR MESSAGE BUFFER
9078 061556 012701 062100      MOV     #T19DT2,R1         ;WRITE SUBSYSTEM DATA BUFFER
9079 061562 112721 000007      MOVB   #PW.WFMT,(R1)+     ;STORE WRITE FORMAT IN BSELO
9080 061566 110021              MOVB   RO,(R1)+           ;STORE DATA WORD IN BSEL1
9081 061570 000207              RTS      PC                ;RETURN
9082
9083      ;*
9083      ; SETUP T19PK2 PACKET FOR WRITE MISC. INVERT EXTENDED FEATURES SWITCH
9084      ;-
9085 061572      T19SEXT:
9086 061572 012700 062100      MOV     #T19DT2,RO         ;WRITE SUBSYSTEM DATA BUFFER
9087 061576 112720 000010      MOVB   #PW.WMISC,(RO)+    ;STORE WRITE MISCELLANEOUS IN BSELO
9088 061602 112710 000200      MOVB   #MS.EXT,(RO)       ;STORE INVERT EXTENDED FEATURES IN BSEL1
9089 061606 000207              RTS      PC                ;RETURN
9090
9091      ;*
9091      ; CLEAR EXPECTED DATA MESSAGE BUFFER
9092      ;-

```

E1

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

9093 061610
 9094 061610 012701 057452
 9095 061614 012700 000120
 9096 061620 105021
 9097 061622 005300
 9098 061624 003375
 9099 061626 000207
 9100
 9101
 9102
 9103
 9104 061630
 9105 061630 012702 057452
 9106 061634 012703 061742
 9107 061640 012700 000010
 9108 061644 012322
 9109 061646 005300
 9110 061650 003375
 9111 061652 000207
 9112
 9113
 9114
 9115
 9116
 9117
 9118
 9119
 9120
 9121
 9122
 9123
 9124
 9125
 9126
 9127
 9128
 9129 061654
 9130 061654
 9131 061660 012701 057406
 9132 061664 005002
 9133 061666 012703 000020
 9134 061672 006000
 9135 061674 103001
 9136 061676 051102
 9137 061700 005721
 9138 061702 005303
 9139 061704 003372
 9140 061706 010200
 9141 061710 000207
 9142
 9143
 9144
 9146 061720
 9148
 9149
 9150
 9151 061720

```

T19CLEXP:
MOV #T19EXP,R1 ;GET EXPD ADDRESS
MOV #T19XEND-T19EXP,R0 ;GET EXPD SIZE
104: CLR (R1)+ ;CLEAR A BYTE
DEC R0 ;DONE?
BGT 104 ;BR IF NO
RTS PC ;RETURN

;*
;Set WORDS 0-7 of expd message BUFFER = to recv since not testing
;
T19SETEXP:
MOV #T19EXP,R2 ;GET EXPD
MOV #T19BFR,R3 ;GET READ STATUS RECV BUFFER
MOV #8,R0 ;SET WORDS 0-7 EXP=RECV
54: MOV (R3)+,(R2)+ ;SET EXPD=RECV
DEC R0 ;DONE WORDS 0-7 WORDS?
BGT 54 ;BR IF NO
RTS PC ;RETURN

;*
; CONVERT A TEST PATTERN DATA WORD TO LOOPBACK DRIVE SIGNALS
;
; INPUTS:
; R0 TEST PATTERN
;
; IMPLICIT INPUTS:
; T19BCTL - CONTAINS WRITE CONTROL / WRITE FORMAT CONVERSION BITS
;
; OUTPUTS:
; R0 - LOW BYTE CONTAINS WRITE CONTROL DATA
; - HIGH BYTE CONTAINS WRITE FORMAT DATA
;
T19CNVT:
SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
MOV #T19BCTL,R1 ;CONVERSION TABLE ADDRESS
CLR R2 ;INIT RESULT OF CONVERSION
MOV #16,R3 ;BIT COUNT
104: ROR R0 ;IS THIS BIT EQUAL TO 1?
BCC 204 ;BR IF NO
BIS (R1),R2 ;SET CONVERTED BIT
204: TST (R1)+ ;POINT TO NEXT BIT IN CONVERSION TABLE
DEC R3 ;DONE?
BGT 104 ;BR IF NO
MOV R2,R0 ;COPY RESULT
RTS PC ;RETURN

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

F1

TEST 8: SUBTEST 4 LOOPBACK READ STROBE TEST

```

9152 061720 100004      .WORD 100004      ;WRITE CHARACTERISTICS COMMAND, WITH ACK
9153 061722 061730      .WORD T19DATA    ;ADDRESS OF CHARACTERISTICS BLOCK
9154 061724 000000      .WORD 0           ;
9155 061726 000012      .WORD 10.        ;MINIMUM MESSAGE PACKET SIZE
9156
9157 061730            T19DATA:          ;CHARACTERISTICS DATA BLOCK
9158 061730 061742      .WORD T19BFR     ;ADDRESS OF MESSAGE BUFFER
9159 061732 000000      .WORD 0           ;
9160 061734 000024      .WORD 20.        ;LENGTH OF MESSAGE BUFFER
9161 061736 000000      .WORD 0           ;ESS,ENB,EAI,ERI
9162 061740 000007      .WORD 7          ;EXTENDED FEATURES UNIT NO.
9163
9164
9165                    ;MESSAGE BUFFER FOR ALL TEST 8 COMMANDS
9166
9167 061742            T19BFR:          ;BEGIN MESSAGE BUFFER
9168 061742 000000      .WORD 0           ;MESSAGE TYPE
9169 061744 000000      .WORD 0           ;DATA FIELD LENGTH
9170 061746 000000      .WORD 0           ;RBPGR
9171 061750 000000      .WORD 0           ;XST0
9172 061752 000000      .WORD 0           ;XST1
9173 061754 000000      .WORD 0           ;XST2
9174 061756 000000      .WORD 0           ;XST3
9175 061760 000000      .WORD 0           ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
9176 061762            T19BFSTA: .BLKB 64. ;READ STATUS AND WRITE FIFO BUFFER
9177 062062            T19BEND:          ;END OF MESSAGE BUFFER
9178
9179                    ;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
9180
9182                    ;
9184 062070            .=<..10>&177770
9185 062070 100006      T19PK2:          ;WRITE SUBSYSTEM WITH ACK
9186 062072 062100      .WORD P.WRTSUB!P.ACK ;LOW ADDRESS OF DATA BLOCK
9187 062074 000000      .WORD T19DT2        ;HIGH ADDRESS OF DATA BLOCK
9188 062076 000012      .WORD 0           ;MINIMUM MESSAGE PACKET SIZE
9189
9190 062100            T19DT2:          ;DATA BLOCK
9191 062100 000         .BYTE 0           ;BSELO
9192 062101 000         .BYTE 0           ;BSEL1
9193 062102 000000      .WORD 0           ;SEL2
9194 062104            .BLKB 64.        ;WRITE FIFO DATA OUTPUT BUFFER
9195
9196
9197 062204            ENDTST
9198 062204
9199 062204 104401

```

L10066: TRAP C4ETST

.SBTTL TEST 9: READ/WRITE DATA PARITY TEST

```

9198
9199
9200 ; TEST DESCRIPTION:
9201 ;
9202 ; This test verifies that the Write Data Parity generator
9203 ; and the Read Data Parity checker operate properly. The
9204 ; Transport Bus signal loopback mode is enabled and a
9205 ; Set Wrong parity function is executed. Then various
9206 ; Write Subsystem Memory functions are performed to
9207 ; write data to and from the FIFO in loopback mode.
9208 ; The program then checks to insure a Read Data parity

```

G1

TEST 9: READ/WRITE DATA PARITY TEST

```

9209 : error occurred.
9210 : A Reset FIFO is done and the Read Data parity
9211 : error bit is again tested to insure it cleared.
9212 : Finally a Clear wrong parity function is done
9213 : and it is verified the data word can pass in loopback
9214 : mode without setting Read Data parity error.
9215 :
9216 : TEST STEPS:
9217 :
9218 : REPEAT FOR LOOPCNT
9219 : BEGIN
9220 : Write to TSSR register to soft initialize the controller
9221 : Do WRITE CHARACTERISTICS to check for Extended Features Switch
9222 : If Extended Features Hardware Switch Clear then:
9223 : Do Write Subsystem Write Miscellaneous to Set Extended Features.
9224 : Do WRITE CHARACTERISTICS to select reserved unit 7 and setup BUFFER
9225 : REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
9226 : BEGIN
9227 : (* Verify Write Wrong Parity Sets Parity Error *)
9228 : Do a WRITE NPR to set loopback and tape direction OUT
9229 : and SET Write Wrong Parity.
9230 : Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
9231 : Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
9232 : Do a READ FIFO with tape direction OUT to load tape out write latch
9233 : (this is when wrong parity (IWP) is set)
9234 : Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 (sets read strobe low)
9235 : (Read Strobe sets PAR IN H [Parity Error])
9236 : Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
9237 : Do a Write Subsystem READ STATUS
9238 : If Read Data parity error NOT=1 Then Print Error
9239 : Do a Write Misc to RESET FIFO
9240 : Do a Write Subsystem READ STATUS
9241 : If Read Data parity error NOT=0 Then Print Error
9242 :
9243 : (* Verify Data can be transferred without a Parity Error *)
9244 : Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
9245 : Do a WRITE NPR to set loopback and tape direction OUT
9246 : and CLEAR Write Wrong Parity.
9247 : Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
9248 : Do a READ FIFO with tape direction OUT to load tape out write latch
9249 : Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 (sets read strobe low)
9250 : (Read Strobe should NOT set PAR IN H [Parity Error] here)
9251 : Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
9252 : Do a Write Subsystem READ STATUS
9253 : If Read Data parity error NOT=0 Then Print Error
9254 :
9255 : END
9256 :
9257 :
9258 :
9259 :

```

```

9259 062206 BGNTST
062206
9264 062206 012700 064572 MOV #TST20ID,R0 T9::
9265 062212 004737 016550 JSR PC,TSTSETUP ;ASCII MESSAGE TO IDENTIFY TEST
9266 062216 012737 000012 002216 MOV #10.,LOOPCNT ;DO INITIAL TEST SETUP
9267 062224 T20LOOP: ;PERFORM 10 ITERATIONS
9268

```

H1

TEST 9: READ/WRITE DATA PARITY TEST

```

9269 062224          BGNSUB                ;//////////////// BEGIN SUBTEST //////////////////
      062224          T9.1:                TRAP    C4BSUB
      062224 104402          Write to TSSR register to soft initialize the controller
9270          ;
9271 062226          ;5:
9272 062226 004737 016034  JSR      PC,SOFINIT          ;WRITE TO TSSR TO SOFT INITIALIZE
9273 062232 103405          BCS      10$                ;BR IF SOFT INIT OKAY
9274 062234 010001          MOV      R0,R1              ;SAVE CONTENTS OF TSSR
9275 062236          ERRDF   ERRNO,SFIERR,SFIMSG ;DEVICE FATAL DURING INIT
      062236 104455          TRAP    C4ERDF
      062240 001604          .WORD   900
      062242 003654          .WORD   SFIERR
      062244 012074          .WORD   SFIMSG
9276          ;
9277 062246 005037 002222  ;10$: Do WRITE CHARACTERISTICS to check for Extended Features Switch
9278 062252 012704 065770  CLR      FATFLG            ;CLEAR FATAL ERROR FLAG
9279 062256 004737 010662  MOV      #T2OPACKET,R4    ;GET THE ADDRESS OF COMMAND PACKET
9280 062262          JSR      PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
9281 062276 103407          FORCERROR 12$            ;@DFORCE ERROR IF FORCER=1
9282 062300 010001          BCS      15$                ;BR IF CARRY SET (GOOD RETURN)
9283 062302          MOV      R0,R1              ;SAVE CONTENTS OF TSSR
9284 062302          12$: ERRDF   ERRNO,T2OSSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      062302 104455          TRAP    C4ERDF
      062304 001605          .WORD   901
      062306 064621          .WORD   T2OSSR
      062310 012106          .WORD   PKTSSR
9285 062312 005237 002222  ;15$: INC      FATFLG            ;SET FATAL ERROR FLAG
9286 062316          CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      062316 104406          TRAP    C4CLP1
9287          ;
9288          ; If Extended Features Hardware Switch Clear then:
9289 062320 012701 066012  ; Do Write Subsystem Write Miscellaneous to Set Extended Features.
9290 062324 032761 000200 000012  MOV      #T20BFR,R1        ;MESSAGE BUFFER ADDRESS
9291 062332 001026          BIT      #X2.EXTF,XST2(R1) ;EXTENDED FEATURES SWITCH SET?
9292 062334 004737 065706          BNE      30$                ;BR IF YES
9293 062340 012704 066140          JSR      PC,T20SEXT        ;SETUP PACKET FOR WRITE MISC INVERT
9294 062344 010465 000000          MOV      #T20PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
9295 062350 004737 016376          MOV      R4,TSDB(R5)     ;SET THE PACKET ADDRESS TO EXECUTE
9296 062354          JSR      PC,CHKTSSR        ;WAIT FOR SSR TO SET
9297 062370 103407          FORCERROR 22$            ;@DFORCE ERROR IF FORCER=1
9298 062372 010001          BCS      30$                ;BR IF CARRY SET (GOOD RETURN)
9299 062374          MOV      R0,R1              ;SAVE CONTENTS OF TSSR
9300 062374          22$: ERRDF   ERRNO,T202SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      062374 104455          TRAP    C4ERDF
      062376 001606          .WORD   902
      062400 064656          .WORD   T202SSR
      062402 012106          .WORD   PKTSSR
9301 062404 005237 002222  ;30$: INC      FATFLG            ;SET FATAL ERROR FLAG
9302 062410          CKLOOP          ;LOOP ON ERROR, IF FLAG SET
      062410 104406          TRAP    C4CLP1
9303          ;
9304 062412 012704 065770  ; Do WRITE CHARACTERISTICS to select reserved unit 7
9305 062416 004737 010662  MOV      #T2OPACKET,R4    ;GET THE ADDRESS OF COMMAND PACKET
9306 062422          JSR      PC,WRTCHR          ;DO WRITE CHARACTERISTICS COMMAND
9307 062436 103407          FORCERROR 42$            ;@DFORCE ERROR IF FORCER=1
9308 062440 010001          BCS      50$                ;BR IF CARRY SET (GOOD RETURN)
9309 062442          MOV      R0,R1              ;SAVE CONTENTS OF TSSR
      NEXT.ERRNO

```


TEST 9: READ/WRITE DATA PARITY TEST

```

9310 062442          424:  ERRDF  ERRNO,T20SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      062442 104455                                     TRAP      C4ERDF
      062444 001607                                     .WORD    903
      062446 064621                                     .WORD    T20SSR
      062450 012106                                     .WORD    PKTSSR
9311 062452 005237 002222          INC      FATFLG      ;SET FATAL ERROR FLAG
9312 062456          504:  CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
      062456 104406                                     TRAP      C4CLP1
9313
9314
9315          ; REPEAT FOR ALL TEST PATTERNS IN TSTBLK TABLE
9316 062460 012703 002754          MOV      #TSTBLK,R3      ;GET FIRST PATTERN ADDRESS
9317 062464 012337 002314          1004:  MOV      (R3)+,DATA      ;GET A TEST PATTERN
9318 062470 042737 177400          BIC      #1C<377>,DATA      ;DATA IS BYTE
9319 062476 010337 002320          MOV      R3,TSTPTR      ;SETUP CURRENT TSTBLK POINTER
9320          ; Do a WRITE NPR to set loopback and tape direction OUT and
9321          ; and SET Write Wrong Parity.
9322 062502 012700 000100          MOV      #NP.OUT,R0      ;SET TAPE DIRECTION OUT
9323 062506 052700 000040          BIS      #NP.LOOP,R0      ;SET LOOPBACK
9324 062512 042700 000020          BIC      #NP.WRP,R0      ;SET WRITE WRONG PARITY (INVERTED)
9325 062516 004737 065556          JSR      PC,T20WNPR      ;SETUP T20PK2 FOR WRITE NPR
9326 062522 012704 066140          MOV      #T20PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
9327 062526 010465 000000          MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
9328 062532 004737 016376          JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
9329 062536          FORCERROR 1024          ;ADDFORCE ERROR IF FORCER=1
9330 062552 103407          BCS      1054          ;BR IF CARRY SET (GOOD RETURN)
9331 062554 010001          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
9332 062556          NEXT.ERRNO
9333 062556          1024:  ERRDF  ERRNO,T204SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      062556 104455                                     TRAP      C4ERDF
      062560 001610                                     .WORD    904
      062562 064767                                     .WORD    T204SSR
      062564 012106                                     .WORD    PKTSSR
9334 062566 005237 002222          INC      FATFLG      ;SET FATAL ERROR FLAG
9335 062572          1054:  CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
      062572 104406                                     TRAP      C4CLP1
9336          ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
9337 062574 012700 000001          MOV      #WF.I4RES,R0      ;IRESV4==>IRSTR = 1
9338 062600 004737 065652          JSR      PC,T20WFMT      ;SETUP T20PK2 FOR WRITE FORMAT
9339 062604 012704 066140          MOV      #T20PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
9340 062610 010465 000000          MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS TO EXECUTE
9341 062614 004737 016376          JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
9342 062620          FORCERROR 1124          ;ADDFORCE ERROR IF FORCER=1
9343 062634 103407          BCS      1204          ;BR IF CARRY SET (GOOD RETURN)
9344 062636 010001          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
9345 062640          NEXT.ERRNO
9346 062640          1124:  ERRDF  ERRNO,T208SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      062640 104455                                     TRAP      C4ERDF
      062642 001611                                     .WORD    905
      062644 065141                                     .WORD    T208SSR
      062646 012106                                     .WORD    PKTSSR
9347 062650 005237 002222          INC      FATFLG      ;SET FATAL ERROR FLAG
9348 062654          1204:  CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
      062654 104406                                     TRAP      C4CLP1
9349          ; Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
9350 062656 012700 000001          MOV      #1,R0          ;WRITE 1 BYTE
9351 062662 012701 002314          MOV      #DATA,R1      ;FIFO WRITE DATA ADDRESS

```

J1

TEST 9: READ/WRITE DATA PARITY TEST

```

9352 062666 004737 065616      JSR      PC,T20WFIF      ;SETUP T20PK2 FOR WRITE FIFO
9353 062672 012704 066140      MOV      #T20PK2,R4     ;GET WRITE SUBSYSTEM COMMAND PACKET
9354 062676 010465 000000      MOV      R4,TSDB(R5)   ;SET THE PACKET ADDRESS TO EXECUTE
9355 062702 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
9356 062706                FORCERROR 152#         ;@DFORCE ERROR IF FORCER=1
9357 062722 103407                BCS      160#         ;BR IF CARRY SET (GOOD RETURN)
9358 062724 010001                MOV      R0,R1        ;SAVE CONTENTS OF TSSR
9359 062726                NEXT.ERRNO
9360 062726 152#:  ERRDF  ERRNO,T205SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C#ERDF
                                .WORD    906
                                .WORD    T205SSR
                                .WORD    PKTSSR
                                062726 104455
                                062730 001612
                                062732 065032
                                062734 012106
9361 062736 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
9362 062742 160#:  CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
062742 104406
; Do a READ FIFO with tape direction OUT to load tape out write latch
; (this is when wrong parity (IWP) is set)
9365 062744 012700 000001      MOV      #1,R0         ;SET READ BYTE COUNT
9366 062750 004737 065576      JSR      PC,T20RFIF     ;SETUP T20PK2 FOR READ FIFO
9367 062754 012704 066140      MOV      #T20PK2,R4     ;GET WRITE SUBSYSTEM COMMAND PACKET
9368 062760 010465 000000      MOV      R4,TSDB(R5)   ;SET THE PACKET ADDRESS TO EXECUTE
9369 062764 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
9370 062770                FORCERROR 172#         ;@DFORCE ERROR IF FORCER=1
9371 063004 103407                BCS      180#         ;BR IF CARRY SET (GOOD RETURN)
9372 063006 010001                MOV      R0,R1        ;SAVE CONTENTS OF TSSR
9373 063010                NEXT.ERRNO
9374 063010 172#:  ERRDF  ERRNO,T206SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C#ERDF
                                .WORD    907
                                .WORD    T206SSR
                                .WORD    PKTSSR
                                063010 104455
                                063012 001613
                                063014 065076
                                063016 012106
9375 063020 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
9376 063024 180#:  CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
063024 104406
; Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 (sets read strobe low)
; (Read Strobe sets PAR IN H [Parity Error])
9379 063026 005000                CLR      R0            ;IRESV4==>IRSTR = 0
9380 063030 004737 065652      JSR      PC,T20WFMT     ;SETUP T20PK2 FOR WRITE FORMAT
9381 063034 012704 066140      MOV      #T20PK2,R4     ;GET WRITE SUBSYSTEM COMMAND PACKET
9382 063040 010465 000000      MOV      R4,TSDB(R5)   ;SET THE PACKET ADDRESS TO EXECUTE
9383 063044 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
9384 063050                FORCERROR 192#         ;@DFORCE ERROR IF FORCER=1
9385 063064 103407                BCS      200#         ;BR IF CARRY SET (GOOD RETURN)
9386 063066 010001                MOV      R0,R1        ;SAVE CONTENTS OF TSSR
9387 063070                NEXT.ERRNO
9388 063070 192#:  ERRDF  ERRNO,T208SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C#ERDF
                                .WORD    908
                                .WORD    T208SSR
                                .WORD    PKTSSR
                                063070 104455
                                063072 001614
                                063074 065141
                                063076 012106
9389 063100 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
9390 063104 200#:  CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
063104 104406
; Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
9391 063106 012700 000001      MOV      #WF.I4RES,R0  ;IRESV4==>IRSTR = 1
9393 063112 004737 065652      JSR      PC,T20WFMT     ;SETUP T20PK2 FOR WRITE FORMAT

```

K1

TEST 9: READ/WRITE DATA PARITY TEST

```

9394 063116 012704 066140      MOV      #T20PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
9395 063122 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
9396 063126 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
9397 063132                FORCERROR      212#     ;@@@FORCE ERROR IF FORCER=1
9398 063146 103407                BCS      220#         ;BR IF CARRY SET (GOOD RETURN)
9399 063150 010001                MOV      R0,R1        ;SAVE CONTENTS OF TSSR
9400 063152                NEXT.ERRNO
9401 063152 212#:  ERRDF  ERRNO,T208SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C#ERDF
                                .WORD    909
                                .WORD    T208SSR
                                .WORD    PKTSSR
                                TRAP      C#CLP1
                                TRAP      C#CLP1
                                TRAP      C#CLP1
                                TRAP      C#CLP1
9402 063162 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
9403 063166 220#:  CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
                                TRAP      C#CLP1
9404                ; Do a Write Subsystem READ STATUS
9405 063170 004737 065536      JSR      PC,T20SRD     ;SETUP PACKET FOR READ STATUS
9406 063174 012704 066140      MOV      #T20PK2,R4    ;GET WRITE SUBSYSTEM COMMAND PACKET
9407 063200 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
9408 063204 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
9409 063210                FORCERROR      232#     ;@@@FORCE ERROR IF FORCER=1
9410 063224 103407                BCS      240#         ;BR IF CARRY SET (GOOD RETURN)
9411 063226 010001                MOV      R0,R1        ;SAVE CONTENTS OF TSSR
9412 063230                NEXT.ERRNO
9413 063230 232#:  ERRDF  ERRNO,T203SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C#ERDF
                                .WORD    910
                                .WORD    T203SSR
                                .WORD    PKTSSR
                                TRAP      C#CLP1
                                TRAP      C#CLP1
                                TRAP      C#CLP1
                                TRAP      C#CLP1
9414 063240 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
9415 063244 240#:  CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
                                TRAP      C#CLP1
9416                ; If Read Data parity error NOT=1 Then Print Error
9417 063246 004737 065744      JSR      PC,T20SETEXP  ;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
9418 063252 012701 064472      MOV      #T20EXSTA,R1  ;GET EXPECTED READ STATUS
9419 063256 012702 066032      MOV      #T20BFSTA,R2 ;GET RCV READ STATUS
9420 063262 011211                MOV      (R2),(R1)     ;SET EXPD WORD #8 = RCV TEMP
9421 063264 016261 000002 000002  MOV      2(R2),2(R1)   ;SET EXPD WORD #9 = RCV (NOT TESTED)
9422 063272 052711 100000      BIS      #S1.PARERR,(R1) ;SET EXP PAR ERR =1
9423 063276 005000                CLR      R0           ;HIGH RCV ADDRESS FOR CKMSG2
9424 063300 012701 066012      MOV      #T20BFR,R1   ;LOW RCV ADDRESS FOR CKMSG2
9425 063304 012702 064452      MOV      #T20EXP,R2   ;EXPD ADDRESS
9426 063310 012703 000024      MOV      #20,R3       ;NUMBER OF BYTES TO COMPARE
9427 063314 004737 011540      JSR      PC,CKMSG2     ;EXPD EQUAL RCV?
9428 063320                FORCERROR      252#,.NOTSSR ;@@@
9429 063330 103404                BCS      260#         ;BR IF YES
9430 063332                NEXT.ERRNO
9431 063332 252#:  ERRHRD  ERRNO,T20SWP,MSGSTAT ;REPORT ERROR
                                TRAP      C#ERHRD
                                .WORD    911
                                .WORD    T20SWP
                                .WORD    MSGSTAT
                                TRAP      C#CLP1
                                TRAP      C#CLP1
                                TRAP      C#CLP1
                                TRAP      C#CLP1
9432 063342 260#:  CKLOOP                ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
                                TRAP      C#CLP1
9433                ; Do a Write Misc to RESET FIFO
9434 063344 012700 000020      MOV      #MS.RSFIF,R0 ;SET RESET FIFO COMMAND
9435 063350 004737 065672      JSR      PC,T20WMISC   ;SETUP T20PK2 FOR WRITE MISC

```

L1

TEST 9: READ/WRITE DATA PARITY TEST

```

9436 063354 012704 066140      MOV      #T20PK2,R4      ;GET WRITE SUBSYSTEM COMMAND PACKET
9437 063360 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
9438 063364 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
9439 063370      FORCERROR 282#         ;###FORCE ERROR IF FORCER=1
9440 063404 103407      BCS     290#           ;BR IF CARRY SET (GOOD RETURN)
9441 063406 010001      MOV      R0,R1         ;SAVE CONTENTS OF TSSR
9442 063410      NEXT.ERRNO
9443 063410 282#      ERRDF  ERRNO,T202SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C#ERDF
                                .WORD    912
                                .WORD    T202SSR
                                .WORD    PKTSSR
9444 063420 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
9445 063424 104406 290#      CKLOOP              ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
9446      ; Do a Write Subsystem READ STATUS
9447      ; If Read Data parity error NOT=0 Then Print Error
9448 063426 004737 065744      JSR      PC,T20SETEXP   ;SET WORDS 0-7 EXPD=RCV (NOT TESTING)
9449 063432 012701 064472      MOV      #T20EXSTA,R1  ;GET EXPECTED READ STATUS
9450 063436 012702 066032      MOV      #T20BFSTA,R2  ;GET RCV READ STATUS
9451 063442 011211      MOV      (R2),(R1)     ;SET EXPD WORD #8 = RCV TEMP
9452 063444 016261 000002 000002      MOV      2(R2),2(R1)   ;SET EXPD WORD #9 = RCV (NOT TESTED)
9453 063452 042711 100000      BIC     #S1.PARERR,(R1) ;SET EXP PAR ERR =0
9454 063456 005000      CLR     R0             ;HIGH RCV ADDRESS FOR CKMSG2
9455 063460 012701 066012      MOV      #T20BFR,R1    ;LOW RCV ADDRESS FOR CKMSG2
9456 063464 012702 064452      MOV      #T20EXP,R2    ;EXPD ADDRESS
9457 063470 012703 000024      MOV      #20,R3        ;NUMBER OF BYTES TO COMPARE
9458 063474 004737 011540      JSR      PC,CKMSG2     ;EXPD EQUAL RCV?
9459 063500      FORCERROR 302#,.NOTSSR ;###
9460 063510 103404      BCS     320#           ;BR IF YES
9461 063512      NEXT.ERRNO
9462 063512 302#      ERRHRD ERRNO,T20RSF,MSGSTAT ;REPORT ERROR
                                TRAP      C#ERHRD
                                .WORD    913
                                .WORD    T20RSF
                                .WORD    MSGSTAT
9463 063522 104406 320#      CKLOOP              ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1
9464      ; (* Verify Data can be transferred without a Parity Error *)
9465      ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
9466 063524 012700 000001      MOV      #F.I4RES,R0   ;IRESV4==>IRSTR = 1
9467 063530 004737 065652      JSR      PC,T20WFM     ;SETUP T20PK2 FOR WRITE FORMAT
9468 063534 012704 066140      MOV      #T20PK2,R4    ;GET WRITE SUBSYSTEM COMMAND PACKET
9469 063540 010465 000000      MOV      R4,TSDB(R5)   ;SET THE PACKET ADDRESS TO EXECUTE
9470 063544 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
9471 063550      FORCERROR 332#         ;###FORCE ERROR IF FORCER=1
9472 063564 103407      BCS     340#           ;BR IF CARRY SET (GOOD RETURN)
9473 063566 010001      MOV      R0,R1         ;SAVE CONTENTS OF TSSR
9474 063570      NEXT.ERRNO
9475 063570 332#      ERRDF  ERRNO,T208SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C#ERDF
                                .WORD    914
                                .WORD    T208SSR
                                .WORD    PKTSSR
9476 063600 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
9477 063604 104406 340#      CKLOOP              ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C#CLP1

```

TEST 9: READ/WRITE DATA PARITY TEST

```

9478      ;      Do a WRITE NPR to set loopback and tape direction OUT and
9479      ;      and CLEAR Write Wrong Parity.
9480 063606 012700 000100      MOV      #NP.OUT,R0      ;SET TAPE DIRECTION OUT
9481 063612 052700 000040      BIS      #NP.LOOP,R0     ;SET LOOPBACK
9482 063616 052700 000020      BIS      #NP.WRP,R0     ;CLEAR WRITE WRONG PARITY (INVERTED)
9483 063622 004737 065556      JSR      PC,T20WNPR     ;SETUP T20PK2 FOR WRITE NPR
9484 063626 012704 066140      MOV      #T20PK2,R4     ;GET WRITE SUBSYSTEM COMMAND PACKET
9485 063632 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
9486 063636 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
9487 063642      FORCERROR      352#      ;@DFORCE ERROR IF FORCER=1
9488 063656 103407      BCS      360#      ;BR IF CARRY SET (GOOD RETURN)
9489 063660 010001      MOV      R0,R1        ;SAVE CONTENTS OF TSSR
9490 063662      NEXT.ERRNO
9491 063662 352#      ERRDF      ERRNO,T204SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
          063662 104455      TRAP      C#ERDF
          063664 001623      .WORD      915
          063666 064767      .WORD      T204SSR
          063670 012106      .WORD      PKTSSR
9492 063672 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
9493 063676 360#      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
          063676 104406      TRAP      C#CLP1
9494      ;      Do a WRITE FIFO with byte count equal to 1 and Tape direction OUT
9495 063700 012700 000001      MOV      #1,R0         ;WRITE 1 BYTE
9496 063704 012701 002314      MOV      #DATA,R1      ;FIFO WRITE DATA ADDRESS
9497 063710 004737 065616      JSR      PC,T20WFIF     ;SETUP T20PK2 FOR WRITE FIFO
9498 063714 012704 066140      MOV      #T20PK2,R4     ;GET WRITE SUBSYSTEM COMMAND PACKET
9499 063720 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
9500 063724 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
9501 063730      FORCERROR      372#      ;@DFORCE ERROR IF FORCER=1
9502 063744 103407      BCS      380#      ;BR IF CARRY SET (GOOD RETURN)
9503 063746 010001      MOV      R0,R1        ;SAVE CONTENTS OF TSSR
9504 063750
9505 063750 372#      ERRDF      ERRNO,T205SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
          063750 104455      TRAP      C#ERDF
          063752 001624      .WORD      916
          063754 065032      .WORD      T205SSR
          063756 012106      .WORD      PKTSSR
9506 063760 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
9507 063764 380#      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
          063764 104406      TRAP      C#CLP1
9508      ;      Do a READ FIFO with tape direction OUT to load tape out write latch
9509 063766 012700 000001      MOV      #1,R0         ;SET READ BYTE COUNT
9510 063772 004737 065576      JSR      PC,T20RFIF     ;SETUP T20PK2 FOR READ FIFO
9511 063776 012704 066140      MOV      #T20PK2,R4     ;GET WRITE SUBSYSTEM COMMAND PACKET
9512 064002 010465 000000      MOV      R4,TSDB(R5)    ;SET THE PACKET ADDRESS TO EXECUTE
9513 064006 004737 016376      JSR      PC,CHKTSSR     ;WAIT FOR SSR TO SET
9514 064012      FORCERROR      392#      ;@DFORCE ERROR IF FORCER=1
9515 064026 103407      BCS      400#      ;BR IF CARRY SET (GOOD RETURN)
9516 064030 010001      MOV      R0,R1        ;SAVE CONTENTS OF TSSR
9517 064032
9518 064032 392#      ERRDF      ERRNO,T206SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
          064032 104455      TRAP      C#ERDF
          064034 001625      .WORD      917
          064036 065076      .WORD      T206SSR
          064040 012106      .WORD      PKTSSR
9519 064042 005237 002222      INC      FATFLG        ;SET FATAL ERROR FLAG
9520 064046 400#      CKLOOP      ;LOOP ON ERROR, IF FLAG SET

```

N1

TEST 9: READ/WRITE DATA PARITY TEST

```

9521 064046 104406 ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 0 (sets read strobe low) TRAP C4CLP1
9522 ; (Read Strobe sets PAR IN H [Parity Error])
9523 064050 005000 CLR R0 ; IRESV4==>IRSTR = 0
9524 064052 004737 065652 JSR PC,T20WFMT ; SETUP T20PK2 FOR WRITE FORMAT
9525 064056 012704 066140 MOV #T20PK2,R4 ; GET WRITE SUBSYSTEM COMMAND PACKET
9526 064062 010465 000000 MOV R4,TSDB(R5) ; SET THE PACKET ADDRESS TO EXECUTE
9527 064066 004737 016376 JSR PC,CHKTSSR ; WAIT FOR SSR TO SET
9528 064072 FORCERROR 412# ; @DFORCE ERROR IF FORCER=1
9529 064106 103407 BCS 420# ; BR IF CARRY SET (GOOD RETURN)
9530 064110 010001 MOV R0,R1 ; SAVE CONTENTS OF TSSR
9531 064112 NEXT.ERRNO
9532 064112 412# ERRDF ERRNO,T208SSR,PKTSSR ; DEVICE FATAL SSR FAILED TO SET
064112 104455 TRAP C4ERDF
064114 001626 .WORD 918
064116 065141 .WORD T208SSR
064120 012106 .WORD PKTSSR
9533 064122 005237 002222 INC FATFLG ; SET FATAL ERROR FLAG
9534 064126 420# CKLOOP ; LOOP ON ERROR, IF FLAG SET TRAP C4CLP1
064126 104406
9535 ; Do a WRITE FORMAT to set IRESV4==>IRSTR = 1 (sets read strobe high)
9536 064130 012700 000001 MOV #WF,I4RES,R0 ; IRESV4==>IRSTR = 1
9537 064134 004737 065652 JSR PC,T20WFMT ; SETUP T20PK2 FOR WRITE FORMAT
9538 064140 012704 066140 MOV #T20PK2,R4 ; GET WRITE SUBSYSTEM COMMAND PACKET
9539 064144 010465 000000 MOV R4,TSDB(R5) ; SET THE PACKET ADDRESS TO EXECUTE
9540 064150 004737 016376 JSR PC,CHKTSSR ; WAIT FOR SSR TO SET
9541 064154 FORCERROR 432# ; @DFORCE ERROR IF FORCER=1
9542 064170 103407 BCS 440# ; BR IF CARRY SET (GOOD RETURN)
9543 064172 010001 MOV R0,R1 ; SAVE CONTENTS OF TSSR
9544 064174 NEXT.ERRNO
9545 064174 432# ERRDF ERRNO,T208SSR,PKTSSR ; DEVICE FATAL SSR FAILED TO SET
064174 104455 TRAP C4ERDF
064176 001627 .WORD 919
064200 065141 .WORD T208SSR
064202 012106 .WORD PKTSSR
9546 064204 005237 002222 INC FATFLG ; SET FATAL ERROR FLAG
9547 064210 440# CKLOOP ; LOOP ON ERROR, IF FLAG SET TRAP C4CLP1
064210 104406
9548
9549 ; Do a Write Subsystem READ STATUS
9550 064212 004737 065536 JSR PC,T20SRD ; SETUP PACKET FOR READ STATUS
9551 064216 012704 066140 MOV #T20PK2,R4 ; GET WRITE SUBSYSTEM COMMAND PACKET
9552 064222 010465 000000 MOV R4,TSDB(R5) ; SET THE PACKET ADDRESS TO EXECUTE
9553 064226 004737 016376 JSR PC,CHKTSSR ; WAIT FOR SSR TO SET
9554 064232 FORCERROR 452# ; @DFORCE ERROR IF FORCER=1
9555 064246 103407 BCS 460# ; BR IF CARRY SET (GOOD RETURN)
9556 064250 010001 MOV R0,R1 ; SAVE CONTENTS OF TSSR
9557 064252 NEXT.ERRNO
9558 064252 452# ERRDF ERRNO,T203SSR,PKTSSR ; DEVICE FATAL SSR FAILED TO SET
064252 104455 TRAP C4ERDF
064254 001630 .WORD 920
064256 064722 .WORD T203SSR
064260 012106 .WORD PKTSSR
9559 064262 005237 002222 INC FATFLG ; SET FATAL ERROR FLAG
9560 064266 460# CKLOOP ; LOOP ON ERROR, IF FLAG SET TRAP C4CLP1
064266 104406
9561 ; If Read Data parity error NOT=0 Then Print Error

```

TEST 9: READ/WRITE DATA PARITY TEST

```

9562 064270 004737 065744 JSR PC,T20SETEXP ;SET WORDS 0-7 EXPD=RECV (NOT TESTING)
9563 064274 012701 064472 MOV #T20EXSTA,R1 ;GET EXPECTED READ STATUS
9564 064300 012702 066032 MOV #T20BFSTA,R2 ;GET RECV READ STATUS
9565 064304 011211 MOV (R2),(R1) ;SET EXPD WORD #8 = RECV TEMP
9566 064306 016261 000002 000002 MOV 2(R2),2(R1) ;SET EXPD WORD #9 = RECV (NOT TESTED)
9567 064314 042711 100000 BIC #S1.PARERR,(R1) ;SET EXP PAR ERR =C
9568 064320 005000 CLR R0 ;HIGH RECV ADDRESS FOR CKMSG2
9569 064322 012701 066012 MOV #T20BFR,R1 ;LOW RECV ADDRESS FOR CKMSG2
9570 064326 012702 064452 MOV #T20EXP,R2 ;EXPD ADDRESS
9571 064332 012703 000024 MOV #20.,R3 ;NUMBER OF BYTES TO COMPARE
9572 064336 004737 011540 JSR PC,CKMSG2 ;EXPD EQUAL RECV?
9573 064342 FORCERROR 472#,NOTSSR ;@D
9574 064352 103404 BCS 480# ;BR IF YES
9575 064354 NEXT.ERRNO
9576 064354 472# : ERRHRD ERRNO,T20CWP,MSGSTAT ;REPORT ERROR
                                TRAP C#ERHRD
                                .WORD 921
                                .WORD T20CWP
                                .WORD MSGSTAT
9577 064364 480# : CKLOOP ;LOOP ON ERROR, IF FLAG SET
                                TRAP C#CLP1
9578 064364 104406
9579 064366 FORCEEXIT 555# ;@D
9580 064376 013703 002320 MOV TSTPTR,R3 ;RESTORE CURRENT TSTBLK POINTER
9581 064402 020327 003064 CMP R3,#TBLEND ;END OF TSTBLK?
9582 064406 103002 BHS 555# ;BR IF YES
9583 064410 000137 062464 JMP 100# ;DO ANOTHER TSTBLK PATTERN
9584 064414 555# :
9585 064414 ENDSUB ;////////// END SUBTEST ///////////
9586 064414 104403 L10074: TRAP C#ESUB
9587 064416 005737 002222 ;ANY FATAL ERRORS ?
9588 064422 001402 BEQ 560# ;BRANCH IF NOT
9589 064424 004737 017242 JSR PC,CKDROP ;TRY TO DROP THE UNIT
9590 064430 560# :
9591 064430 004737 016516 JSR PC,TSTLOOP ;DO ITERATIONS?
9592 064434 103002 BCC 565# ;BR IF NO
9593 064436 000137 050116 JMP T18LOOP ;LOOP UNTIL ITERATIONS DONE
9594 064442 565# :
9595 064442 EXIT TST ;////////// EXIT TEST ///////////
9596 064442 104432 TRAP C#EXIT
9597 064444 001610 .WORD L10073-.
9598
9599 ;*
9600 ;LOCAL STORAGE FOR THIS TEST
9601 ;-
9602
9603
9604 064446 T20MSK: ;MASK OF UNTESTED BITS IN READ STATUS
9605 ;UNTESTED BITS ARE SET TO 1
9606 064446 377 .BYTE #C<000> ;BYTE 0 MASK
9607 064447 037 .BYTE #C<340> ;BYTE 1 MASK (PARERR,IRESV2,IRESV1)
9608 064450 360 .BYTE #C<017> ;BYTE 2 (TIMER A,TIMER B,UNDEFINED<1:0>)
9609 064451 000 .BYTE 0 ;MAKE IT EVEN

```

TEST 9: READ/WRITE DATA PARITY TEST

```

9610
9611 064452          T20EXP:          ;BEGIN EXPECTED DATA BUFFER
9612 064452 000000   .WORD          0          ;MESSAGE TYPE
9613 064454 000000   .WORD          0          ;DATA FIELD LENGTH
9614 064456 000000   .WORD          0          ;RBPGR
9615 064460 000000   .WORD          0          ;XST0
9616 064462 000000   .WORD          0          ;XST1
9617 064464 000000   .WORD          0          ;XST2
9618 064466 000000   .WORD          0          ;XST3
9619 064470 000000   .WORD          0          ;XST4 (ALWAYS PRESENT FOR WRITE SUB.)
9620 064472          T20EXSTA: .BLKB 64.   ;EXPECTED READ STATUS AND WRITE FIFO DATA
9621 064572          T20XEND:          ;END EXPECTED DATA BUFFER
9622
9623 ;*
9624 ;LOCAL TEXT MESSAGES FOR TEST
9625 ;-
9626 064572          122          145          141  TST20ID: .ASCIZ 'Read/Write Data Parity'
9627 064621          127          122          111  T20SSR: .ASCIZ 'WRITE CHARACTERISTICS Failed'
9628 064656          127          122          111  T202SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Misc) Failed'
9629 064722          127          122          111  T203SSR: .ASCIZ 'WRITE SUBSYSTEM (Read Status) Failed'
9630 064767          127          122          111  T204SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Npr) Failed'
9631 065032          127          122          111  T205SSR: .ASCIZ 'WRITE SUBSYSTEM (Write FIFO) Failed'
9632 065076          127          122          111  T206SSR: .ASCIZ 'WRITE SUBSYSTEM (Read FIFO) Failed'
9633 065141          127          122          111  T208SSR: .ASCIZ 'WRITE SUBSYSTEM (Write Format) Failed'
9634 065207          122          145          141  T20SMP: .ASCIZ 'Read Data Parity Error (PARERR) Failed to Set after Write Wrong Parity'
9635 065316          122          145          141  T20RSF: .ASCIZ 'Read Data Parity Error (PARERR) Failed to Clear after RESET FIFO'
9636 065417          122          145          141  T20CWP: .ASCIZ 'Read Data Parity Error (PARERR) occurred in Data Loopback'
9637
9638
9639 ;*
9640 ; CLEAR MESSAGE BUFFER
9641 ;-
9642 065512          T20CLRBUF:
9643 065512          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
9644 065516 012701 066012  MOV          #T20BFR,R1      ;GET MESSAGE BUFFER ADDRESS
9645 065522 012702 000120  MOV          #T20BEND-T20BFR,R2 ;SIZE OF MESSAGE BUFFER IN BYTES
9646 065526 105021          104:  CLRB          (R1)          ;CLEAR A BYTE
9647 065530 005302          DEC          R2          ;DONE?
9648 065532 003375          BGT          104          ;BR IF NO
9649 065534 000207          RTS          PC          ;RETURN
9650
9651 ;*
9652 ; SETUP T20PK2 PACKET FOR READ STATUS
9653 ;-
9654 065536          T20SRD:
9655 065536 004737 065512  JSR          PC,T20CLRBUF    ;CLEAR MESSAGE BUFFER
9656 065542 012700 066150  MOV          #T20DT2,R0      ;WRITE SUBSYSTEM DATA BUFFER
9657 065546 112720 000005  MOVB         #PW,RDSTATUS,(R0) ;STORE READ STATUS COMMAND IN BSEL0
9658 065552 105010          CLRB          (R0)          ;CLEAR BSEL1
9659 065554 000207          RTS          PC          ;RETURN
9660
9661 ;*
9662 ; SETUP T20PK2 PACKET FOR WRITE NPR
9663 ;
9664 ; INPUT:
9665 ; R0 CONTAINS BSEL1 NPR DATA
9666 ;

```


TEST 9: READ/WRITE DATA PARITY TEST

```

9667
9668
9669 065556
9670 065556 004737 065512
9671 065562 012701 066150
9672 065566 112721 000011
9673 065572 110011
9674 065574 000207
9675
9676
9677
9678
9679
9680
9681
9682 065576
9683 065576 004737 065512
9684 065602 012701 066150
9685 065606 112721 000003
9686 065612 110021
9687 065614 000207
9688
9689
9690
9691
9692
9693
9694
9695 065616
9696 065616
9697 065622 004737 065512
9698 065626 012702 066150
9699 065632 112722 000004
9700 065636 110022
9701 065640 005022
9702 065642 112122
9703 065644 005300
9704 065646 003375
9705 065650 000207
9706
9707
9708
9709
9710
9711
9712
9713 065652
9714 065652 004737 065512
9715 065656 012701 066150
9716 065662 112721 000007
9717 065666 110021
9718 065670 000207
9719
9720
9721
9722
9723

;
;--
T20LNPR:
      JSR    PC,T20CLRBUF      ;CLEAR MESSAGE BUFFER
      MOV    @T20DT2,R1       ;WRITE SUBSYSTEM DATA BUFFER
      MOVB   @PW,LNPR,(R1)    ;STORE WRITE NPR IN BSELO
      MOVB   R0,(R1)         ;STORE NPR DATA IN BSEL1
      RTS    PC               ;RETURN

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

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

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

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

```

TEST 9: READ/WRITE DATA PARITY TEST

```

9724 065672
9725 065672 012701 066150
9726 065676 112721 000010
9727 065702 110011
9728 065704 000207
9729
9730
9731
9732 065706
9733 065706 012700 066150
9734 065712 112720 000010
9735 065716 112710 000200
9736 065722 000207
9737
9738
9739
9740 065724
9741 065724 012701 064452
9742 065730 012700 000120
9743 065734 105021
9744 065736 005300
9745 065740 003375
9746 065742 000207
9747
9748
9749
9750
9751 065744
9752 065744 012702 064452
9753 065750 012703 066012
9754 065754 012700 000010
9755 065760 012322
9756 065762 005300
9757 065764 003375
9758 065766 000207
9759
9760
9761
9765
9766
9767
9768 065770
9769 065770 100004
9770 065772 066000
9771 065774 000000
9772 065776 000012
9773
9774 066000
9775 066000 066012
9776 066002 000000
9777 066004 000024
9778 066006 000000
9779 066010 000007
9780
9781
9782
9783

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

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

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

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

;
;WRITE CHARACTERISTICS COMMAND PACKET
;
T20PACKET:
;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND, WITH ACK
;ADDRESS OF CHARACTERISTICS BLOCK
;MINIMUM MESSAGE PACKET SIZE
;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER
;LENGTH OF MESSAGE BUFFER
;ESS,ENB,EAI,ERI
;EXTENDED FEATURES UNIT NO.
.WORD 100004
.WORD T20DATA
.WORD 0
.WORD 10.

T20DATA:
.WORD T20BFR
.WORD 0
.WORD 20.
.WORD 0
.WORD 7

;MESSAGE BUFFER FOR ALL TEST 17 COMMANDS

```

TEST 9: READ/WRITE DATA PARITY TEST

9784 066012
 9785 066012 000000
 9786 066014 000000
 9787 066016 000000
 9788 066020 000000
 9789 066022 000000
 9790 066024 000000
 9791 066026 000000
 9792 066030 000000
 9793 066032
 9794 066132
 9795
 9796
 9797
 9799 066140 066140
 9801 066140
 9802 066140 100006
 9803 066142 066150
 9804 066144 000000
 9805 066146 000012
 9806
 9807 066150
 9808 066150 000
 9809 066151 000
 9810 066152 000000
 9811 066154
 9812
 9813
 9814 066254
 066254
 066254 104401
 9815
 9816
 9817
 9818
 9819
 9820
 9821
 9822
 9823
 9824
 9825
 9826
 9827
 9828
 9829
 9830
 9831
 9832
 9833
 9834
 9835
 9836
 9837
 9838
 9839
 9840

```

T20BFR:                                ;BEGIN MESSAGE BUFFER
        .WORD 0                          ;MESSAGE TYPE
        .WORD 0                          ;DATA FIELD LENGTH
        .WORD 0                          ;RBPCR
        .WORD 0                          ;XST0
        .WORD 0                          ;XST1
        .WORD 0                          ;XST2
        .WORD 0                          ;XST3
        .WORD 0                          ;XST4 (ALWAYS PRESENT FOR WRITE SUBSYSTEM
T20BFSTA: .BLKB 64.                    ;READ STATUS AND WRITE FIFO BUFFER
T20BEND:                                ;END OF MESSAGE BUFFER
;
;WRITE SUBSYSTEM READ STATUS COMMAND PACKET
;
        .=<.10>&177770
T20PK2:                                ;WRITE SUBSYSTEM WITH ACK
        .WORD P.WRTSUB:P.ACK            ;LOW ADDRESS OF DATA BLOCK
        .WORD T20DT2                    ;HIGH ADDRESS OF DATA BLOCK
        .WORD 0                          ;MINIMUM MESSAGE PACKET SIZE
        .WORD 10.
T20DT2:                                ;DATA BLOCK
        .BYTE 0                          ;BSEL0
        .BYTE 0                          ;BSEL1
        .WORD 0                          ;SEL2
        .BLKB 64.                       ;WRITE FIFO DATA OUTPUT BUFFER
    
```

ENDTST

L10073: TRAP C#ETST

.SBTTL TEST 10: MANUAL INTERVENTION

```

;THE MANUAL INTERVENTION TEST IS A STANDALONE ROUTINE (NOT REALLY A "TEST")
;THAT ALLOWS THE OPERATOR TO CHECK OUT VARIOUS ELEMENTS AND FUNCTIONS OF
;THE SUBSYSTEM THAT CANNOT BE MANIPULATED BY THE PROGRAM ALONE. WHEN
;THIS ROUTINE IS STARTED, IT FIRST PRINTS OUT A MENU OF SELECTABLE
;SUBTESTS AND THEN WAITS FOR THE OPERATOR TO TYPE IN A SELECTION CODE.
;THE ONLY WAYS TO EXIT THIS ROUTINE AND RETURN TO THE DIAGNOSTIC SUPERVISOR
;ARE BY TYPING <CTRL-C> OR SELECTING CODE 7.
;SELECTION CODES AND SUBROUTINES ARE:
    
```

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

```

;EACH MENU ITEM CORRESPONDS TO A SUBTEST, AS FOLLOWS:
;
    
```

TEST 10: MANUAL INTERVENTION

```
9841      ;
9842      ;
9843      ;PRINTS OUT THE MENU ON THE CONSOLE TERMINAL.
9844      ;
9845      ;
9846      ;CAUSES ALL THREE LED INDICATORS ON THE M7196 MODULE
9847      ;TO BE ILLUMINATED. AFTER INITIATING THIS ROUTINE, THE OPERATOR
9848      ;SHOULD OBSERVE THE LED'S AND VERIFY THAT THEY ARE INDEED ALL LIT.
9849      ;THIS ROUTINE FIRST USES THE WRITE SUBSYSTEM MEMORY COMMAND TO
9850      ;SET THE FORCE WRONG PARITY FLIP-FLOP, WHICH SERVES TO DRIVE THE
9851      ;"PROCESSOR NOT OK" LED. THEN IT ENTERS A LOOP THAT CONTINUALLY
9852      ;WRITES THE LOW BYTE OF TSDB AND READS THE TSSR. THESE LATTER TWO
9853      ;OPERATIONS WILL CAUSE THE "NOT SSR" AND "DRIVING BUS" LED'S TO
9854      ;GLOW -- THEY ARE NOT REALLY LIT AT ALL TIMES BUT SHOULD APPEAR
9855      ;REASONABLY VISIBLE.
9856      ;
9857      ;
9858      ;INITIALIZES THE CONTROLLER TO CAUSE ALL LED'S TO
9859      ;EXTINGUISH.
9860      ;
9861      ;
9862      ;
9863      ;
9864      ;THIS ROUTINE INITIALIZES THE CONTROLLER, ISSUES A
9865      ;WRITE CHARACTERISTICS COMMAND TO ENABLE ATTENTION INTERRUPTS,
9866      ;ISSUES A MESSAGE BUFFER RELEASE COMMAND, PRINTS A MESSAGE ON THE
9867      ;CONSOLE TERMINAL INSTRUCTING THE OPERATOR TO TOGGLE THE ON-LINE
9868      ;SWITCH ON THE TRANSPORT, THEN WAITS FOR AN ATTENTION INTERRUPT.
9869      ;EACH TIME THE TRANSPORT TRANSITIONS FROM ON-LINE TO OFF-LINE OR
9870      ;VICE-VERSA, AN ATTENTION INTERRUPT SHOULD BE GENERATED. THE PROGRAM
9871      ;WILL REPORT THE INTERRUPT AND THE CURRENT STATE OF THE TRANSPORT.
9872      ;THE OPERATOR SHOULD VERIFY THAT THE REPORTED STATE MATCHES THE
9873      ;STATE INDICATED BY THE LED ON THE FRONT PANEL OF THE TRANSPORT.
9874      ;IN ADDITION, WHEN THE TRANSPORT IS PLACED OFF-LINE, THE PROGRAM
9875      ;ISSUES A SEQUENCE OF TAPE-MOTION COMMANDS (READ, WRITE, POSITION, ETC.)
9876      ;AND VERIFIES THAT, FOR EACH COMMAND, FUNCTION REJECT TERMINATION
9877      ;RESULTS, ALONG WITH THE NON-EXECUTABLE FUNCTION (NEF) ERROR BIT BEING
9878      ;SET.
9879      ;
9880      ;THIS ROUTINE INSTRUCTS THE OPERATOR TO MOUNT A SCRATCH
9881      ;TAPE REEL THAT DOES NOT HAVE A WRITE-ENABLE RING INSTALLED, THEN
9882      ;WAITS FOR THE OPERATOR TO RESPOND THAT THIS HAS BEEN ACCOMPLISHED.
9883      ;UPON THE RESPONSE, THE PROGRAM VERIFIES THAT THE TRANSPORT SHOWS
9884      ;A WRITE-PROTECTED STATUS, THEN ATTEMPTS TO WRITE DATA ON THE
9885      ;TAPE AND EXPECTS THE APPROPRIATE ERROR TERMINATION INDICATING THAT
9886      ;THE WRITE FUNCTION COULD NOT BE PERFORMED BECAUSE THE REEL IS
9887      ;WRITE-PROTECTED. IF THE APPROPRIATE TERMINATION IS NOT RECEIVED,
9888      ;AN ERROR IS REPORTED.
9889      ;
9890      ;
9891      ;
9892      ;INSTRUCTS THE OPERATOR TO PLACE THE TAPE TRANSPORT(S)
9893      ;ON-LINE (IF ANY ARE OFF-LINE) THEN ATTEMPTS TO PERFORM AN EXTENDED
9894      ;STATUS READOUT. FOR EACH TRANSPORT EQUIPPED WITH THIS FEATURE,
9895      ;THE PROGRAM FORMATS AND PRINTS OUT THE RESULTING STATUS. IF THE
9896      ;TRANSPORT IS NOT EQUIPPED WITH THIS FEATURE, A MESSAGE INDICATING
9897      ;SUCH IS ISSUED.
```

H2

TEST 10: MANUAL INTERVENTION

```

9898
9899
9900
9901
9902 066256          BGNTST
          066256
9907 066256          RFLAGS RO          ;GET OPERATOR FLAGS          T10::
          066256 104421          ;BR, IF OK TO RUN          TRAP C4RFLA
          066260 001403          ;"TEST NOT EXECUTED"
9908 066260 001403          BEQ 21:          ;JUMP IF NOT FIRST TEST
9909 066262 012700 071630          MOV #T38NE,R0
9910 066266 000402          BR 3:
9911 066270          21:
9912 066270 012700 072745          MOV #T38ID,R0
9913 066274 004737 016550          34: JSR PC,TSTSETUP
9914 066300 004737 020022          JSR PC,CHKMAN
9915 066304 103402          BCS 22:
9916 066306 000137 071032          JMP 64:
9917 066312          22:
9921 066312 005037 002222          24: CLR FATFLG
9922 066316 012737 176750 071044          MOV #65000.,T38DLY
9923 066324 004737 016034          54: JSR PC,SOFINIT
9924 066330 103427          BCS 23:
9925 066332 010001          MOV RO,R1
9926 066334 032701 000200          BIT #SSR,R1
9927 066340 001023          BNE 23:
9928 066342          DELAY 250
          066342 012727 000250          ;CALL DELAY ROUTINE
          066346 000000          MOV #250,(PC)+
          066350 013727 002116          .WORD 0
          066354 000000          MOV L4DLY,(PC)+
          066356 005367 177772          .WORD 0
          066362 001375          DEC -6(PC)
          066364 005367 177756          BNE -4
          066370 001367          DEC -22(PC)
9929 066372 005337 071044          BNE -20
9930 066376 001352          DEC T38DLY
9931 066400          BNE 54
          066400 104455          ;BUMP COUNTER DOWN
          066402 001751          ;BR, IF MORE TIME LEFT
          066404 003654          ;REPORT FATAL ERROR
          066406 012074          TRAP C4ERDF
9932 066410 012700 072772          234: MOV #MIMENU,R0
9933 066414 012701 000005          MOV #5,R1
9934 066420 004737 017600          JSR PC,GETSEL
9935 066424 010004          ;MENU OF MANUAL INTERVENTIONS
9936 066426 006304          ;MAXIMUM ALLOWED SELECTION
9937 066430 000174 066434          MOV RO,R4
9938 066434 066312          64: ASL R4
9939 066436 066450          ;GO GET THE OPERATORS SELECTION
9940 066440 066732          ;GET NUMBER FROM ROUTINE
9941 066442 067164          ;CONVERT TO WORD OFFSET
9942 066444 067620          ;JUMP TO PROPER LOOP
9943 066446 070554          JMP 264(R4)
9944 066450          104: .WORD 24
          066450 012746 072641          ;RETYPE THE MENU
          066454 012746 000001          ; 1 TURN ON LED'S
          066460 010600          ; 2 TURN OFF LED'S
          ; 3 ONLINE ATTENTION
          ; 4 WRITE PROTECT
          ; 5 EXTENDED TRANSPORT STATUS
          ;TELL OPERATOR TO CNTRL-C FOR EXIT
          MOV #T38MS2,-(SP)
          MOV #1,-(SP)
          MOV SP,R0

```

TEST 10: MANUAL INTERVENTION

```

066462 104417
066464 062706 000004
9945 066470 004737 073402
9946 066474 004737 016034
9947 066500 103405
9951 066502 010001
9952 066504
066504 104455
066506 001752
066510 003654
066512 012074
9953 066514 013737 002202 071570 1004:
9954
9955 066522 012704 071550
9956 066526 004737 010662
9957 066532 103405
9961 066534 010001
9962 066536
066536 104456
066540 001753
066542 005060
066544 012074
9963 066546
9964 066546 112737 000000 071061
9965 066554 112737 000011 071060
9966 066562 012704 071050
9967
9968
9969
9970 066566 010465 000000
9971 066572 004737 016376
9972 066576 103405
9973 066600 010001
9977 066602
066602 104455
066604 001754
066606 072246
066610 012106
9978 066612
066612 104406
9979 066614
066614 012700 000340
066620 104441
9980 066622 005037 071036
9981 066626 032737 000100 177560
9982 066634 001005
9983 066636 005237 071036
9984 066642 052737 000100 177560
9985 066650 012701 000060
9986 066654 011137 071040
9987 066660 012721 070336
9988 066664 011137 071042
9989 066670 012711 000340
9990 066674
066674 012700 000000
066700 104441
9991 066702 012701 177777

```

```

                                TRAP      C4PNTF
                                ADD        #4,SP
JSR      PC,T38REST             ;SET PACKET TO INITIAL VALUES
JSR      PC,SOFINIT            ;DO SOFT INIT OF CONTROLLER
BCS      1004                  ;BR IF SOFT INIT = OK
MOV      R0,R1                 ;SAVE CONTENTS OF TSSR
ERRDF    ERRNO,SFIERR,SFIMSG   ;DEVICE FATAL ERROR DURING INIT
                                TRAP      C4ERDF
                                .WORD    1002
                                .WORD    SFIERR
                                .WORD    SFIMSG
MOV      UNITN,T38DSW          ;SET UNIT NUMBER
MOV      #T38PK2,R4            ;SUBROUTINE NEEDS PACKET ADDRESS
JSR      PC,WRTCHR             ;ISSUE WRITE CHARACTERISTICS
BCS      1104                  ;BR, IF COMMAND ISSUED OK
MOV      R0,R1                 ;SAVE CONTENTS OF TSSR
ERRHRD   ERRNO,WRTMSG,SFIMSG  ;WRITE CHARACTERISTIC FAILED
                                TRAP      C4ERHRD
                                .WORD    1003
                                .WORD    WRTMSG
                                .WORD    SFIMSG
1104:
MOVB     #0,T38BS1             ;CLEAR BIT #4
MOVB     #11,T38BS0           ;WRITE MISC COMMAND
MOV      #T38PACKET,R4        ;SET UP NEW WRT. SUBSYS MEM. COMMAND
;NOTE: THIS COMMAND TURNS ON THE PROCESSOR FAIL LED
;
MOV      R4,TSDB(R5)          ;SET THE PACKET ADDRESS
JSR      PC,CHKTSSR          ;WAIT FOR SSR TO SET
BCS      1504                  ;BR IF CARRY SET (GOOD RETURN)
MOV      R0,R1                 ;SAVE CONTENTS OF TSSR
ERRDF    ERRNO,T38SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C4ERDF
                                .WORD    1004
                                .WORD    T38SSR
                                .WORD    PKTSSR
1504:
CKLOOP
;LOOP ON ERROR, IF FLAG SET
                                TRAP      C4CLP1
SETPRI   #PRI07              ;RAISE THE PRIORITY
                                MOV      #PRI07,R0
                                TRAP      C4SPRI
CLR      TTION2               ;ASSUME INTERRUPTS ARE ENABLED
BIT      #100,@TTICSR        ;ARE TTI INTERRUPTS ON ?
BNE      7014                 ;BRANCH IF YES
INC      TTION2               ;FLAG SET IF INTERRUPTS OFF
BIS      #100,@TTICSR        ;ENABLE INTERRUPTS
7014:
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      C4SPRI
MOV      #-1,R1              ;DATA TO WRITE TO TSDB

```

TEST 10: MANUAL INTERVENTION

```

9992 066706 000240          124:  NOP                    ;ALLOW OPERATOR TO TYPE ^C
9993 066710 012702 001750    MOV      #1000.,R2          ;SET-UP INNER LOOP
9994 066714 110165 000000    144:  MOVB     R1,TSD8(R5)      ;WRITE DATA TO TSD8
9995 066720 016500 000002    MOV      TSSR(R5),R0      ;READ TSSR
9996 066724 005302          DEC      R2                ;REDUCE INNER COUNT
9997 066726 001372          BNE     144                ;LOOP TILL EXPIRES
9998 066730 000766          BR      124                ;LOOP UNTIL HALTED
9999
10000 066732          154:  PRINTF  #T38MS2          ;TYPE CNTL C TO EXIT
      066732 012746 072641    MOV      #T38MS2,-(SP)
      066736 012746 000001    MOV      #1,-(SP)
      066742 010600          MOV      SP,R0
      066744 104417          TRAP    C4PNTF
      066746 062706 000004    ADD     #4,SP
10001 066752 004737 016034    JSR     PC,SOFINIT        ;DO SOFT INIT OF CONTROLLER
10002 066756 103405          BCS    2004                ;BR IF SOFT INIT = OK
10006 066760 010001          MOV     R0,R1              ;SAVE CONTENTS OF TSSR
10007 066762          ERRDF  ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
      066762 104455          TRAP    C4ERDF
      066764 001755          .WORD  1005
      066766 003654          .WORD  SFIERR
      066770 012074          .WORD  SFIMSG
10008 066772
10009 066772 013737 002202 071570 2004:  MOV     UNITN,T38DSW      ;SET UNIT NUMBER
10010 067000 012704 071550    MOV     #T38PK2,R4        ;SUBROUTINE NEEDS PACKET ADDRESS
10011 067004 004737 010662    JSR     PC,WRTCHR         ;ISSUE WRITE CHARACTERISTICS
10012 067010 103405          BCS    2104                ;BR, IF COMMAND ISSUED OK
10016 067012 010001          MOV     R0,R1              ;SAVE CONTENTS OF TSSR
10017 067014          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      067014 104456          TRAP    C4ERHRD
      067016 001756          .WORD  1006
      067020 005060          .WORD  WRTMSG
      067022 012074          .WORD  SFIMSG
10018
10019
10020
10021
10022
10023 067024          ;*****
10024 067024 112737 000000 071061    ; THIS WRITE SUB-SYSTEM MEMORY COMMAND JUST HOLDS THE LEDS OFF
10025 067032 112737 000025 071060    ;*****
10026 067040 012704 071050    2104:  MOVB    #0,T38BS1        ;CLEAR BIT #4
10027 067044 010465 000000    MOVB    #25,T38BS0       ;STOP DRIVE TEST 22
10028 067050 004737 016376    MOV     #T38PACKET,R4    ;SET UP NEW WRT. SUBSYS MEM. COMMAND
10029 067054 103405          MOV     R4,TSD8(R5)      ;SET THE PACKET ADDRESS
10030 067056 010001          JSR     PC,CHKTSSR       ;WAIT FOR SSR TO SET
10034 067060          BCS    2504                ;BR IF CARRY SET (GOOD RETURN)
      067060 104455          MOV     R0,R1              ;SAVE CONTENTS OF TSSR
      067062 001757          ERRDF  ERRNO,T38SSR,FKTSSR ;DEVICE FATAL SSR FAILED TO SET
      067064 072246          TRAP    C4ERDF
      067066 012106          .WORD  1007
10035 067070          .WORD  T38SSR
      067070 104406          .WORD  PKTSSR
      067072          2504:  CKLOOP                    ;LOOP ON ERROR, IF FLAG SET
10036 067072          SETPRI #PRI07            ;RAISE THE PRIORITY
      067072 012700 000340    TRAP    C4CLP1
      067076 104441          MOV     #PRI07,R0
10037 067100 005037 071036    CLR     TTION2           ;ASSUME INTERRUPTS ARE ENABLED
      067076 104441          TRAP    C4SPRI

```

TEST 10: MANUAL INTERVENTION

```

10038 067104 032737 000100 177560      BIT      @100,@TTICSR      ;ARE TTI INTERRUPTS ON ?
10039 067112 001005                BNE      7104             ;BRANCH IF YES
10040 067114 005237 071036                INC      TTION2          ;FLAG SET IF INTERRUPTS OFF
10041 067120 052737 000100 177560      BIS      @100,@TTICSR      ;ENABLE INTERRUPTS
10042 067126 012701 000060      7104:   MOV      @TIVEC,R1        ;START OF TTI VECTORS
10043 067132 011137 071040                MOV      (R1),TVSAV2      ;SAVE THE CURRENT TTI VECTOR
10044 067136 012721 070336                MOV      @5904,(R1)-      ;SET NEW INTERRUPT ROUTINE
10045 067142 011137 071042                MOV      (R1),TPSAV2      ;SAVE THE VECTOR PRIORITY
10046 067146 012711 000340                MOV      @PRI07,(R1)      ;USE PRIORITY SEVEN
10047 067152                SETPRI   @PRI00          ;LOWER INTERRUPT BR LEVEL
        067152 012700 000000                MOV      @PRI00,R0
        067156 104441                TRAP     C$SPRI
10048 067160 000240      2604:   NOP
10049 067162 000776                BR       2604             ;ALLOW CNTL C
                                ;LOOP UNTIL STOPPED
10050
10051
10052 067164      204:   PRINTF  @T38MS2          ;TELL'EM WHAT TO TYPE
        067164 012746 072641                MOV      @T38MS2,-(SP)
        067170 012746 000001                MOV      @1,-(SP)
        067174 010600                MOV      SP,R0
        067176 104417                TRAP     C$PNTF
        067200 062706 000004                ADD      @4,SP
10053 067204                SETPRI   @PRI00          ;LOWER PRIORITY TO ALLOW INTERRUPTS
        067204 012700 000000                MOV      @PRI00,R0
        067210 104441                TRAP     C$SPRI
10054 067212 005037 002224                CLR      INTRECV          ;CLEAR INTERRUPT RECEIVED FLAG
10055 067216 004737 016034                JSR      PC,SOFINIT       ;DO SOFT INIT OF CONTROLLER
10056 067222 103405                BCS      3004             ;BR IF SOFT INIT = OK
10060 067224 010001                MOV      R0,R1           ;SAVE CONTENTS OF TSSR
10061 067226                ERRDF   ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
        067226 104455                TRAP     C$ERDF
        067230 001760                .WORD   1008
        067232 003654                .WORD   SFIERR
        067234 012074                .WORD   SFIMSG
10062 067236      3004:   MOV      UNITN,T38DSW     ;SET UNIT NUMBER IN PACKET
10063 067236 013737 002202 071570      MOV      @BIT5,T38EAI     ;ENABLE ATTENTION INTERRUPTS
10064 067244 012737 000040 071566      MOV      @T38PK2,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
10065 067252 012704 071550                JSR      PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
10066 067256 004737 010662                BCS      3104             ;BR, IF COMMAND ISSUED OK
10071 067264 010001                MOV      R0,R1           ;SAVE CONTENTS OF TSSR
10072 067266                ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
        067266 104456                TRAP     C$ERHRD
        067270 001761                .WORD   1009
        067272 005060                .WORD   WRTMSG
        067274 012074                .WORD   SFIMSG
10073 067276      3104:   MOV      @T38PK3,R4      ;SET UP NEW PACKET FOR MESS BUF REL
10074 067276 012704 071600                MOV      R4,TSDB(R5)     ;MESSAGE BUFFER RELEASE,ACK,CVC=1 CMD
10075 067302 010465 000000                JSR      PC,WAITF        ;WAIT FOR SSR TO SET
10076 067306 004737 016310                CLR      R2              ;MAKE SURE ALL IS CLEAR
10077 067312 005002                MOV      TSSR(R5),R1     ;GET TSSR STATUS
10078 067314 016501 000002                BIT      @OFL,R1         ;IS OFL SET
10079 067320 032701 000100                BEQ      3204             ;BR, IF OFL IS NOT SET
10080 067324 001402                BIS      @OFL,R1         ;SET OFL IN EXPECTED
10081 067326 052702 000100      3204:   BIS      @SSR,R2         ;SET UP EXPECTED
10082 067332 052702 000200                CMP      R2,R1           ;IS EVERYTHING OK
10083 067336 020201

```


L2

TEST 10: MANUAL INTERVENTION

```

10084 067340 001404          BEQ      3504          ;BR, IF ALL IS WELL
10088 067342          ERRHRD  ERRNO,T38SST,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP      C4ERHRD
                                .WORD    1010
                                .WORD    T38SST
                                .WORD    PKTSSR
10089 067352          3504:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C4CLP1
10090 067354          PRINTF  #T38MS1        ;TELL OPERATOR TO TOGGLE SWITCH
                                MOV       #T38MS1,-(SP)
                                MOV       #1,-(SP)
                                MOV       SP,R0
                                TRAP      C4PNTF
                                ADD      #4,SP
10091 067374          PRINTF  #T38MS2        ;TELL OPERATOR TO DO ↑C TO EXIT
                                MOV       #T38MS2,-(SP)
                                MOV       #1,-(SP)
                                MOV       SP,R0
                                TRAP      C4PNTF
                                ADD      #4,SP
10092 067414          SETPRI  #PRI07          ;RAISE THE PRIORITY
                                MOV       #PRI07,R0
                                TRAP      C4SPRI
10093 067422          CLR      TTION2          ;ASSUME INTERRUPTS ARE ENABLED
10094 067426          BIT      #100,#TTICSR ;ARE TTI INTERRUPTS ON ?
10095 067434          BNE     7204          ;BRANCH IF YES
10096 067436          INC     TTION2          ;FLAG SET IF INTERRUPTS OFF
10097 067442          BIS     #100,#TTICSR ;ENABLE INTERRUPTS
10098 067450          MOV     #TTIVEC,R1    ;START OF TTI VECTORS
10099 067454          MOV     (R1),TVSAV2    ;SAVE THE CURRENT TTI VECTOR
10100 067460          MOV     #5904,(R1)-  ;SET NEW INTERRUPT ROUTINE
10101 067464          MOV     (R1),TPSAV2    ;SAVE THE VECTOR PRIORITY
10102 067470          MOV     #PRI07,(R1)  ;USE PRIORITY SEVEN
10103 067474          SETPRI  #PRI00          ;LOWER INTERRUPT BR LEVEL
                                MOV       #PRI00,R0
                                TRAP      C4SPRI
10104 067502          3604:  NOP              ;ALLOW CONTROL C
10105 067504          TST     INTRECV        ;DID AN INTERRUPT OCCUR ?
10106 067510          BNE     3704          ;BRANCH IF YES
10107 067512          BR     3604          ;WAIT SOME MORE FOR INTERRUPT
10108 067514          3704:  PRINTF  #T38INT        ;"INTERRUPT RECEIVED"
                                MOV       #T38INT,-(SP)
                                MOV       #1,-(SP)
                                MOV       SP,R0
                                TRAP      C4PNTF
                                ADD      #4,SP
10109 067534          MOV     TSSR(R5),R1    ;READ TSSR STATUS
10110 067540          BIT     #OFL,R1      ;CHECK THE OFF-LINE BIT
10111 067544          BNE     3804          ;BR, IF DRIVE IS OFF-LINE
10112 067546          PRINTF  #T38ONL        ;"DRIVE IS NOW ON-LINE"
                                MOV       #T38ONL,-(SP)
                                MOV       #1,-(SP)
                                MOV       SP,R0
                                TRAP      C4PNTF
                                ADD      #4,SP
10113 067566          3804:  BR     3904          ;ALMOST DONE
10114 067570          PRINTF  #T38OFL        ;"DRIVE IS NOW OFF-LINE"

```

TEST 10: MANUAL INTERVENTION

067570	012746	072422					MOV	#T380FL,-(SP)
067574	012746	000001					MOV	#1,-(SP)
067600	010600						MOV	SP,R0
067602	104417						TRAP	C#PNTF
067604	062706	000004					ADD	#4,SP
10115	067610	005037	002224	390#:	CLR	INTRECV		;CLEAR INTERRUPT FLAG
10116	067614	000137	067236		JMP	300#		;TRY AGAIN
10117	067620			25#:	GMANIL	T38MSG,T38DAT,-1,NO		;WAIT FOR OPERATOR TO MOUNT TAPE
	067620	104443					TRAP	C#GMAN
	067622	000404					BR	10000#
	067624	073400					.WORD	T38DAT
	067626	000120					.WORD	T#CODE
	067630	072705					.WORD	T38MSG
	067632	177777					.WORD	-1
	067634							
10118	067634				BNCOMPLETE	25#		;RETRY IF ERROR 10000#:
	067634	103371						
10119	067636	005737	073400		TST	T38DAT		BCC 25#
10120	067642	001002			BNE	27#		;DID OPERATOR SAY 'YES' ?
10121	067644	000137	066312		JMP	2#		;BRANCH IF YES
10122	067650			27#:				;RETURN TO MAIN MENU
10123	067650	004737	016034		JSR	PC,SOFINIT		;DO SOFT INIT OF CONTROLLER
10124	067654	103405			BCS	400#		;BR IF SOFT INIT = OK
10128	067656	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR
10129	067660				ERRDF	ERRNO,SFIERR,SFIMSG		;DEVICE FATAL ERROR DURING INIT
	067660	104455					TRAP	C#ERDF
	067662	001763					.WORD	1011
	067664	003654					.WORD	SFIERR
	067666	012074					.WORD	SFIMSG
10130	067670			400#:	CKLOOP			;LOOP IF SELECTED
	067670	104406					TRAP	C#CLP1
10131	067672	013737	002202	071570	MOV	UNITN,T38DSW		;SET UNIT NUMBER
10132	067700	012704	071550		MOV	#T38PK2,R4		;SUBROUTINE NEEDS PACKET ADDRESS
10133	067704	004737	010662		JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS
10134	067710	103405			BCS	410#		;BR, IF COMMAND ISSUED OK
10138	067712	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR
10139	067714				ERRHRD	ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTIC FAILED
	067714	104456					TRAP	C#ERHRD
	067716	001764					.WORD	1012
	067720	005060					.WORD	WRTMSG
	067722	012074					.WORD	SFIMSG
10140	067724			410#:	CKLOOP			;LOOP IF SELECTED
	067724	104406					TRAP	C#CLP1
10141	067726	013701	071074		MOV	T38BFR-6,R1		;PICK UP XSTO CONTENTS
10142	067732	010102			MOV	R1,R2		;SET UP EXPECTED
10143	067734	052702	000004		BIS	#BIT2,R2		;SET UP THE WRITE LOCKED BIT
10144	067740	020102			CMF	R1,R2		;ARE THEY CORRECT
10145	067742	001406			BEQ	430#		;BR, IF ALL IS WELL (OK)
10149	067744				ERRHRD	ERRNO,T38WRL,EXPREC		;WRITE LOCKED BIT IS NOT SET ETC."
	067744	104456					TRAP	C#ERHRD
	067746	001765					.WORD	1013
	067750	072064					.WORD	T38WRL
	067752	015534					.WORD	EXPREC
10150	067754	005237	002222		INC	FATFLG		;SET FATAL FLAG
10151	067760			430#:	CKLOOP			;LOOP IF SELECTED
	067760	104406					TRAP	C#CLP1
10152	067762	005737	002222		TST	FATFLG		;WAS THE DRIVE NOT WRITE LOCKED

TEST 10: MANUAL INTERVENTION

```

10153 067766 001402          BEQ      435$          ;BR, IF FLAG NOT SET
10154 067770 000137 066312    JMP      2$          ;RE-WRITE MENU
10155 067774 017737 113126 071622 435$:  MOV     @FREE,T38WR  ;SET UP WRITE BUFFER ADDRESS
10156 070002 012704 071620    MOV     @T38PK4,R4   ;GET PACKET ADDRESS
10157 070006 010465 000000    MOV     R4,TSD8(R5)  ;SET THE PACKET ADDRESS
10158 070012 004737 016310    JSR     PC,WAITF     ;WAIT FOR SSR TO SET
10159 070016 016501 000002    MOV     TSSR(R5),R1  ;GET TSSR
10160 070022 012702 100206    MOV     @SC!SSR!BIT1!BIT2,R2 ;SET UP EXPECTED
10161 070026 020102          CMP     R1,R2        ;ARE THEY EQUAL (CORRECT)
10162 070030 001404          BEQ     440$         ;BR, IF CORRECT STATUS
10166 070032          ERRHRD  ERRNO,T38WRT,PKTSSR ;"TSSR INCORRECT AFTER WRITE COMMAND
                                TRAP      C$ERHRD
                                .WORD    1014
                                .WORD    T38WRT
                                .WORD    PKTSSR
                                TRAP      C$CLP1
                                .WORD    104456
                                .WORD    001766
                                .WORD    072000
                                .WORD    012106
10167 070042          440$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
                                TRAP      C$CLP1
                                .WORD    104406
                                .WORD    013701 071074
10168 070044          MOV     T38BFR+6,R1  ;READ XSTO CONTENTS
10169 070050          MOV     R1,R2        ;SET UPR EXPECTED
10170 070052 052702 004000    BIS     @BIT11,R2    ;SET THE WRITE LOCK ERROR BIT (XSTO)
10171 070056 020102          CMP     R1,R2        ;WAS THE BIT SET
10172 070060 001404          BEQ     450$         ;BR, IF IT WAS (GOOD)
10176 070062          ERRHRD  ERRNO,T38MLE,EXPREC ;"WRITE LOCK ERROR BIT NOT SET"
                                TRAP      C$ERHRD
                                .WORD    1015
                                .WORD    T38MLE
                                .WORD    EXPREC
                                TRAP      C$CLP1
                                .WORD    104456
                                .WORD    001767
                                .WORD    072125
                                .WORD    015534
10177 070072          450$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    104406
                                .WORD    070072 104406
10178 070074 000137 066312    JMP      2$          ;GO BACK TO MENU
10179
10180 ;*****
10181 ;   SERVO EXERCISER NO LONGER USED
10182 ;*****
10183 070100          30$:  PRINTB  @T38MS3    ;"EXE ANY OTHER MENU SELECTION TO STOP
10184 070100          MOV     @T38MS3,-(SP)
                                MOV     #1,-(SP)
                                MOV     SP,R0
                                TRAP      C$PNTB
                                ADD     #4,SP
                                .WORD    012746 071705
                                .WORD    012746 000001
                                .WORD    010600
                                .WORD    104414
                                .WORD    062706 000004
10185 070120 004737 073402    JSR     PC,T38REST   ;SET PACKET TO INITIAL VALUES
10186 070124 004737 016034    JSR     PC,SOFINIT   ;DO SOFT INIT OF CONTROLLER
10187 070130 103405          BCS     500$         ;BR IF SOFT INIT = OK
10191 070132 010001          MOV     R0,R1        ;SAVE CONTENTS OF TSSR
10192 070134          ERRDF  ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
                                TRAP      C$ERDF
                                .WORD    1016
                                .WORD    SFIERR
                                .WORD    SFIMSG
                                .WORD    002202 071570 500$:
10193 070144          MOV     UNITN,T38DSW ;SET UNIT NUMBER
10194 070152          MOV     @T38PK2,R4  ;SUBROUTINE NEEDS PACKET ADDRESS
10195 070156 004737 010662    JSR     PC,WRTCHR    ;ISSUE WRITE CHARACTERISTICS
10196 070162 103405          BCS     510$         ;BR, IF COMMAND ISSUED OK
10200 070164 010001          MOV     R0,R1        ;SAVE CONTENTS OF TSSR
10201 070166          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP      C$ERHRD
                                .WORD    104456

```

TEST 10: MANUAL INTERVENTION

```

070170 001771
070172 005060
070174 012074
10202 070176
10203 070176 112737 000000 071061 5104:
10204 070204 112737 000020 071060
10205 070212 012704 071050
10206 070216 010465 000000
10207 070222 004737 016376
10208 070226 103405
10209 070230 010001
10213 070232
070232 104455
070234 001772
070236 072246
070240 012106
10214 070242 5504: CKLOOP
070242 104406
10215 070244 SETPRI #PRI07
070244 012700 000340
070250 104441
10216 070252 005037 071036
10217 070256 032737 000100 177560
10218 070264 001005
10219 070266 005237 071036
10220 070272 052737 000100 177560
10221 070300 012701 000060 5554:
10222 070304 011137 071040
10223 070310 012721 070336
10224 070314 011137 071042
10225 070320 012711 000340
10226 070324
070324 012700 000000
070330 104441
10227 070332 000240
10228 070334 000776
10229
10230
10231
10232
10233 070336 010046
10234 070340 113700 177562
10235 070344 042700 000200
10236 070350 122700 000015
10237 070354 001075
10238 070356 012766 066312 000002
10239 070364 005066 000004
10240 070370 013737 071040 000060
10241 070376 013737 071042 000062
10242 070404 112737 000025 071060
10243 070412 112737 000000 071061
10244 070420 012704 071050
10245 070424 010465 000000
10246 070430 012737 176750 071044
10247 070436 004737 016310
10248 070442 016501 000002
10249 070446 032701 000200

MOV #0,T38BS1 ;CLEAR BIT #4
MOV #20,T38BS0 ;EXECUTE DRIVE TEST 22
MOV #T38PACKET,R4 ;SET UP NEW WRT. SUBSYS MEM. COMMAND
MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS
JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
BCS 5504 ;BR IF CARRY SET (GOOD RETURN)
MOV RO,R1 ;SAVE CONTENTS OF TSSR
ERRDF ERRNO,T38SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET

TRAP C#ERDF
.WORD 1017
.WORD WRTMSG
.WORD SFMSG

TRAP C#ERDF
.WORD 1018
.WORD T38SSR
.WORD PKTSSR

SET TRAP C#CLP1
TRAP C#CLP1

MOV #PRI07,RO
TRAP C#SPRI

CLR TTION2 ;ASSUME INTERRUPTS ARE ENABLED
BIT #100,@TTICSR ;ARE TTI INTERRUPTS ON ?
BNE 5554 ;BRANCH IF YES
INC TTION2 ;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 #5904,(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,RO
TRAP C#SPRI

NOP ;LOOP AWHILE
BR 5604 ;STAY IN "TIGHT" LOOP

;+
;PROCESS CONSOLE INTERRUPTS
;-

MOV RO,-(SP) ;SAVE WORK REGISTER
MOVB #@TTIBFR,RO ;GET THE OPERATOR INPUT
BIC #200,RO ;STRIP OFF PARITY BIT
CMPB #15,RO ;IS IT A CARRIAGE RETURN ?
BNE 5914 ;JUST EXIT IF NOT
MOV #24,2(SP) ;RETURN TO MASTER MENU
CLR 4(SP) ;FORCE PRIORITY 0
MOV TVSAV2,@TTIVEC ;RESTORE VECTOR
MOV TPSAV2,@TTIVEC-2 ;RESTORE SUPER PRIORITY
MOVB #25,T38BS0 ;STOP DRIVE TEST 22
MOVB #0,T38BS1 ;CLEAR BS1
MOV #T38PACKET,R4 ;SET UP NEW WRT. SUBSYS MEM. COMMAND
MOV R4,TSDB(R5) ;SET THE PACKET ADDRESS
MOV #65000,T38DLY ;SET UP DELAY COUNTER
JSR PC,WAITF ;DO A WAIT FOR SSR
MOV TSSR(R5),R1 ;CONTENTS OF TSSR REGISTER
BIT #SSR,R1 ;CHECK FOR TSSR SET

```


TEST 10: MANUAL INTERVENTION

```

10294 070656 104406                                TRAP    C4CLP1
10294 070660 112737 000000 071061             MOVB   #0,T38BS1           ;CLEAR BIT #4
10295 070666 112737 000024 071060             MOVB   #24,T38BS0        ;READ EXTENDED DRIVE STATUS
10296 070674 012704 071050             MOV    #T38PACKET,R4    ;SET UP NEW WRT. SUBSYS MEM. COMMAND
10297 070700 010465 000000             MOV    R4,TSD8(R5)      ;SET THE PACKET ADDRESS
10298 070704 012737 000144 071044             MOV    #100.,T38DLY     ;SET UP DELAY ROUTINE
10299 070712 004737 016310 6204:         JSR    PC,WAITF         ;WAIT AWHILE FOR SSR TO SET
10300 070716 016501 000002             MOV    TSSR(R5),R1     ;SEE IF IT REALLY DID
10301 070722 032701 000200             BIT    #SSR,R1         ;JUST CHECK THAT BIT
10302 070726 001017 6304:         BNE   6304             ;BR, IF SSR IS SET
10303 070730 012727 000250             DELAY  250             ;DELAY ABOUT .25 SEC
070730 012727 000250
070734 000000
070736 013727 002116
070742 000000
070744 005367 177772
070750 001375
070752 005367 177756
070756 001367
10304 070760 005337 071044             DEC    T38DLY          ;START DELAY COUNT DOWN
10305 070764 001352 6204:         BNE   6204             ;BR, IF COUNTER IS NOT AT DONE
10306 070766 004737 016376             JSR    PC,CHKTSSR      ;WAIT FOR SSR TO SET
10307 070772 103405 6504:         BCS   6504             ;BR IF CARRY SET (GOOD RETURN)
10308 070774 010001             MOV    R0,R1           ;SAVE CONTENTS OF TSSR
10312 070776 104455             ERRDF  ERRNO,T38SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
070776 104455
071000 001776
071002 072246
071004 012106
10313 071006 6504:         CKLOOP          ;LOOP ON ERROR, IF FLAG SET
071006 104406
10314 071010 012700 071106             MOV    #T38BFR-20,R0   ;MESSAGE BUFFER ADDRESS
10315 071014 005001 R1           CLR    R1              ;NO HIGH ORDER ADDRESS BITS
10316 071016 005037 003136             CLR    KTENABLE        ;NO KT11 STUFF EITHER
10317 071022 004737 073440             JSR    PC,T38MBP       ;GO PRINT MESSAGE BUFFER CONTENTS
10318 071026 000137 066312             JMP    24              ;GO BACK TO MENU
10319
10320
10321 634:         JMP    200             ;TYPE +C TO RETURN TO THE SUPERVISOR
10322 071032 644:         EXIT   TST          ;LEAVE TEST
071032 104432
071034 003056                                TRAP    C4EXIT
071034 003056                                .WORD  L10075-.

10323
10324
10325
10326
10327
10328
10329
10330
10331
10332
10333
10334 071036 000000             TTION2: .WORD 0        ;WORD SET IF SUPERVISOR TTI INTER OFF
10335 071040 000000             TVSAV2: .WORD 0        ;SAVE TTI VECTOR
10336 071042 000000             TPSAV2: .WORD 0        ;SAVE TTI PRIORITY
10337

```

E3

TEST 10: MANUAL INTERVENTION

```

10338 071044 000000 T38DLY: .WORD 0 ;DELAY COUNTER FOR TEST
10340 071050 071050 .=< .10>E177770
10342 071050 T38PACKET: ;COMMAND PACKET FOR TEST
10343 071050 140006 .WORD 140006 ;WRITE SUBSYSTEM MEM. CMD, ACK,CVC=1
10344 071052 071060 .WORD T38TAD ;ADDRESS OF CHARACTERISTICS BLOCK
10345 071054 000000 .WORD 0
10346 071056 000012 .WORD 10. ;STARTING VALUE OF BLOCK SIZE
10347 071060 T38TAD: ;CHARACTERISTICS DATA BLOCK
10348 071060 000 T38BS0: .BYTE 0 ;BSEL0 BYTE
10349 071061 000 T38BS1: .BYTE 0 ;BSEL1 BYTE
10350 071062 000000 T38BS2: .WORD 0 ;BSEL1 WORD
10351 071064 000000 .WORD 0 ;DATA
10352 071066 T38BFR: .BLKW 150. ;MESSAGE BUFFER
10353 071542 000000 T38EB: .WORD ;END OF BUFFER ADDRESS
10354
10355
10357 071550 071550 .=< .10>E177770
10359 071550 T38PK2: ;COMMAND PACKET FOR TEST
10360 071550 140004 .WORD 140004 ;WRITE CHARA. MEM. CMD., ACK,CVC=1
10361 071552 071560 .WORD T38DTA ;ADDRESS OF SELECT DATA BLOCK
10362 071554 000000 .WORD 0
10363 071556 000012 .WORD 10. ;STARTING VALUE OF BLOCK SIZE
10364
10365
10366 071560 T38DTA: ;SELECT DATA BLOCK
10367 071560 071066 .WORD T38BFR ;ADDRESS OF MESSAGE BUFFER
10368 071562 000000 .WORD 0
10369 071564 000400 .WORD 256. ;LENGTH OF MESSAGE BUFFER
10370 071566 000000 T38EAI: .WORD 0 ;EAI BIT WORD
10371 071570 000000 T38DSW: .WORD 0 ;DRIVE SELECT WORD ETC
10373 071600 071600 .=< .10>E177770
10375 071600 140212 T38PK3: .WORD 140212 ;MESSAGE BUFFER RELEASE COMMAND
10376 071602 000000 .WORD 0 ;NOT USED
10377 071604 000000 .WORD 0 ;NOT USED
10378 071606 000000 .WORD 0 ;NOT USED
10379 071610 000000 .WORD 0 ;NOT USED
10380
10381 ;WRITE TAPE PACKET
10382
10384 071620 071620 .=< .10>E177770
10386 071620 140005 T38PK4: .WORD 140005 ;WRITE, ACK, CVC=1 COMMAND
10387 071622 000000 T38WR: .WORD 0 ;ADDRESS OF WRITE BUFFER
10388 071624 000000 .WORD 0 ;MORE ADDRESS OF WRITE BUFFER
10389 071626 000400 T38SIZ: .WORD 256. ;SIZE OF RECORD
10390
10391
10392
10393
10394
10395 ;*
10396 ;LOCAL TEXT MESSAGES FOR TEST
10397 ;-
10398
10399
10400
10401
10402 071630 123 164 141 T38NE: .ASCIZ 'Stand-alone Manual Intervention Not Executed'
```

TEST 10: MANUAL INTERVENTION

```

10403 071705      045      116      045  T38MS3: .ASCIZ  'MMA Type <RETURN> To Stop Servo Exerciser, Return To Menu'
10404 072000      124      123      123  T38WRT: .ASCIZ  'TSSR Not Correct After WRITE, With WRITE PROTECT On'
10405 072064      127      122      111  T38WRL: .ASCIZ  'WRITE LOCKED Bit Not Set In XST0'
10406 072125      127      122      111  T38WLE: .ASCIZ  'WRITE LOCK ERROR Bit Not Set In XST0'
10407 072172      127      122      111  T38NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
10408 072246      103      157      156  T38SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
10409 072336      045      116      045  T38INT: .ASCIZ  'MMA Interrupt Received'
10410 072366      045      116      045  T38ONL: .ASCIZ  'MMA Drive Is Now ON-LINE'
10411 072422      045      116      045  T38OFL: .ASCIZ  'MMA Drive Is Now OFF-LINE'
10412 072456      103      157      156  T38SST: .ASCIZ  'Contents Of TSSR Incorrect After MESSAGE BUFFER RELEASE'
10413 072546      045      116      045  T38MS1: .ASCIZ  'MMA Toggle ON-LINE Switch to Generate ATTENTION Interrupts'
10414 072641      045      116      045  T38MS2: .ASCIZ  'MMA Type RETURN To Return To Menu'
10415 072705      111      163      040  T38MSG: .ASCIZ  'Is Write-Protected Tape Mounted'
10416 072745      115      141      156  T38ID:  .ASCIZ  'Manual Intervention'
10417           .EVEN
10418 072772      073016  073070  073116  MIMENU: .WORD   14,24,34,44,54,64
10419 073006      073265  073330  073377  .WORD   84,94,104,0
10420
10421 073016      012      123      105  14:    .ASCIZ  '<12>' SELECT OPERATION FROM FOLLOWING OPTIONS:'
10422 073070      012      011      060  24:    .ASCIZ  '<12>' 0 Display This Menu'
10423 073116      011      061      011  34:    .ASCIZ  ' 1 Turn On All M7196 LED's'
10424 073150      011      062      011  44:    .ASCIZ  ' 2 Turn Off All M7196 LED's'
10425 073203      011      063      011  54:    .ASCIZ  ' 3 Offline/Online Attention'
10426 073237      011      064      011  64:    .ASCIZ  ' 4 Write Protect Test'
10427 073265      011      065      011  84:    .ASCIZ  ' 5 Print Extended Transport Status'
10428 073330      011      136      103  94:    .ASCIZ  ' 9C To Return to Diagnostic Supervisor'
10429 073377      000
10430           .EVEN
10431
10432           ;*
10433           ;LOCAL STORAGE FOR THIS TEST
10434           ;-
10435
10436 073400      000000  T38DAT: .WORD   0 ;LOGICAL RESPONSE TO QUESTION
10437 073402  T38REST:
10438 073402  SAVREG
10439 073406  012701  071050  MOV     #T38PACKET,R1 ;SAVE THE REGISTERS
10440 073412  012721  140206  MOV     #140206,(R1)- ;START OF THE PACKET
10441 073416  012721  071060  MOV     #T38TAD,(R1)- ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
10442 073422  005021  CLR     (R1)- ;ADDRESS OF DATA BLOCK
10443 073424  012721  000006  MOV     #6,(R1)- ;EXTENDED ADDRESS
10444 073430  005021  CLR     (R1)- ;SIZE OF DATA BLOCK IN BYTES
10445 073432  005021  CLR     (R1)- ;CLEAR BSELO AND BSEL1
10446 073434  005011  CLR     (R1)- ;CLEAR SEL2
10447 073436  000207  RTS     PC ;CLEAR DATA AREA
10448           ;RETURN
10449
10450           ;*
10451           ;
10452           ;THIS ROUTINE PRINTS THE CONTENTS OF
10453           ;THE 256 BYTE MESSAGE BUFFER RETURNED BY THE
10454           ;TSV-05.
10455           ;
10456           ;INPUT:
10457           ;
10458           ; R0 LOW ORDER ADDRESS OF MESSAGE BUFFER
10459           ; R1 HIGH ORDER ADDRESS OF MESSAGE BUFFER

```


TEST 10: MANUAL INTERVENTION

```

10460      ; NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
10461      ;
10462      ;
10463      ;
10464      ;
10465 073440      T38MBP: SAVREG
10466 073440      MOV R0,R5 ;SAVE THE REGISTERS
10467 073444 010005 003136 TST KTENABLE ;SAVE LOW ORDER ADDRESS
10468 073446 005737 003136 BNE 9104 ;ADDRESS ABOVE 28K?
10469 073452 001001 ;BR IF YES
10470 073454 005001 CLR R1 ;SET HIGH ORDER ADDRESS TO 0
10471 073456 010103 9104: MOV R1,R3 ;SAVE HIGH ORDER ADDRESS
10472 073460 006100 ROL R0 ;SHIFT BIT15 TO C BIT
10473 073462 006101 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
10474 073464 PRINTX #T38AS0,R1,R5 ;PRINT MESSAGE BUFFER ADDRESS
      MOV R5,-(SP)
      MOV R1,-(SP)
      MOV #T38AS0,-(SP)
      MOV #3,-(SP)
      MOV SP,R0
      TRAP C#PNTX
      ADD #10,SP
      MOV #T38AS1,-(SP)
      MOV #1,-(SP)
      MOV SP,R0
      TRAP C#PNTX
      ADD #4,SP
10475 073510 PRINTX #T38AS1 ;PRINT HEADER FOR CONTENTS
      MOV #T38AS1,-(SP)
      MOV #1,-(SP)
      MOV SP,R0
      TRAP C#PNTX
      ADD #4,SP
10476 073530 MOV R5,R1 ;COPY LOW ORDER ADDRESS
10477 073532 MOV R3,R0 ;COPY HIGH ORDER ADDRESS
10478 073534 BEQ 9134 ;BR IF NOT ABOVE 28K
10479 073536 004737 017356 JSR PC,SETMAP ;SETUP PAR ADDRESS IN R0
10480 073542 010005 MOV R0,R5 ;GET PAR FORMAT ADDRESS ABOVE 28K
10481 073544 010537 074110 9134: MOV R5,T38CNT ;HOLD ADDRESS
10482 073550 011504 9114: MOV (R5),R4 ;GET BUFFER ENTRY
10483 073552 022704 125252 CMP #125252,R4 ;CHECK FOR NO LOAD CONDITION
10484 073556 001417 BEQ 9124 ;BR, IF BUFFER WASN'T LOADED
10485 073560 010403 MOV R4,R3 ;MAKE COPY
10486 073562 042704 170377 BIC #170377,R4 ;ONLY BITS 11,10,9 AND 8 ARE SAVED
10487 073566 000241 CLC ;CLEAR CARRY
10488 073570 006004 ROR R4 ;11 TO 10 BIT POSITION
10489 073572 006004 ROR R4 ;10 TO 9 BIT POSITION
10490 073574 006004 ROR R4 ;9 TO 8 BIT POSITION
10491 073576 006004 ROR R4 ;8 TO 7 BIT POSITION
10492 073600 042703 177760 BIC #177760,R3 ;ONLY BITS 3,2,1 AND 0 ARE SAVED
10493 073604 060403 ADD R4,R3 ;"OR'EM TOGETHER
10494 073606 010325 MOV R3,(R5) ;PUT BACK IN BUFFER
10495 073610 020527 071542 CMP R5,#T38EB ;END OF BUFFER YET
10496 073614 001355 BNE 9114 ;BR, IF NOT AT END YET
10497 073616 013705 074110 9124: MOV T38CNT,R5 ;PUT ADDRESS BACK
10498 073622 012704 000001 MOV #1,R4 ;START BYTE NUMBER AT ONE
10499 073626 9154: PRINTX #T38ASN,R4,(R5) ;PRT MEM BUFFER W/NEWLINE
      MOV (R5),-(SP)
      MOV R4,-(SP)
      MOV #T38ASN,-(SP)
      MOV #3,-(SP)
      MOV SP,R0
073464 010546
073466 010146
073470 012746 073742
073474 012746 000003
073500 010600
073502 104415
073504 062706 000010
073510 012746 074007
073514 012746 000001
073520 010600
073522 104415
073524 062706 000004
073530 010501
073532 010300
073534 001403
073536 004737 017356
073542 010005
073544 010537 074110
073550 011504
073552 022704 125252
073556 001417
073560 010403
073562 042704 170377
073566 000241
073570 006004
073572 006004
073574 006004
073576 006004
073600 042703 177760
073604 060403
073606 010325
073610 020527 071542
073614 001355
073616 013705 074110
073622 012704 000001
073626 012546
073630 010446
073632 012746 074064
073636 012746 000003
073642 010600

```

H3

TEST 10: MANUAL INTERVENTION

```

073644 104415
073646 062706 000010
10500 073652 005037 074110
10501 073656 000412
10502 073660
          9204: PRINTX @T38ASC,R4,(R5)
073660 012546
073662 010446
073664 012746 074045
073670 012746 000003
073674 010600
073676 104415
073700 062706 000010
10503 073704 005237 074110
10504 073710 005204
10505 073712 020427 000200
10506 073716 003010
10507 073720 023727 074110 000004
10508 073726 001401
10509 073730 000753
10510 073732 005037 074110
10511 073736 000733
10512 073740 000207
          9214: INC T38CNT ;BUMP COUNTER
          INC R4 ;NUMBER OF THE NEXT
          CMP R4,#128. ;DONE ALL YET ?
          BGT 504 ;BRANCH IF ALL DONE
          CMP T38CNT,#4 ;DONE FOUR YET
          BEQ 9254 ;BR, IF THREE DONE
          BR 9204 ;KEEP GOING
          9254: CLR T38CNT ;CLEAR COUNTER
          BR 9154 ;PRINT WITH NEW LINE
          504: RTS PC ;RETURN
10514 073742 045 116 045 T38AS0: .ASCIZ '#N#A Message Buffer Address = #01#05'
10515 074007 045 116 045 T38AS1: .ASCIZ '#N#A Message Buffer Contents:'
10516 074045 045 101 040 T38ASC: .ASCIZ '#A #D4#A: #03'
10517 074064 045 116 045 T38ASN: .ASCIZ '#N#A Bytes#D4#A: #03'
10518
10519 074110 000000 T38CNT: .WORD ;COUNTER FOR PRINT
10520 074112
          074112
          074112 104401
          L10075: TRAP C#ETST

```

```

10521 .SBTTL TEST 11: CONFIGURATION TYPEOUT
10522
10523 ;THIS IS A STANDALONE ROUTINE THAT PRINTS OUT ON THE CONSOLE TERMINAL
10524 ;THE CONFIGURATION OF THE M7196 MODULE AND TSV05 SUBSYSTEM. SPECIFICALLY,
10525 ;THE FOLLOWING INFORMATION IS PRESENTED:
10526 ;
10527 ;
10528 ; 1.0 STATE OF THE EXTENDED FEATURES SWITCH ON THE M7196: ON (EXTENDED
10529 ; FEATURES ENABLED) OR OFF (EXTENDED FEATURES DISABLED),
10530 ;
10531 ; 2.0 STATE OF THE BUFFERING ENABLE SWITCH ON THE M7196: ON
10532 ; (BUFFERING ENABLED) OR OFF (BUFFERING DISABLED),
10533 ;
10534 ; 3.0 MICROCODE REVISION LEVEL OF THE M7196,
10535 ;
10536 ; 4.0 NUMBER OF TAPE TRANSPORTS CONNECTED TO THE CONTROLLER,
10537 ;
10538 ; 5.0 UNIT SELECT CODE AND STATE (ONLINE/OFFLINE, WRITE ENABLED/PROTECTED)
10539 ; OF EACH CONNECTED TRANSPORT. IN ADDITION, THE PROGRAM WILL INDICATE,
10540 ; FOR EACH ON-LINE TRANSPORT, WHETHER OR NOT IT IS EQUIPPED WITH THE
10541 ; EXTENDED TAPE STATUS READOUT FEATURE.
10542 ;
10543 ;
10544 ;THE OPERATOR IS EXPECTED TO READ THE PRINTOUT AND VERIFY THAT IT MATCHES
10545 ;THE ACTUAL CONFIGURATION AT HAND. IF, FOR EXAMPLE, THE PROGRAM INDICATES

```

TEST 11: CONFIGURATION TYPEOUT

```

10546 ;THAT IT "SEES" TWO TRANSPORTS CONNECTED WHEN IN FACT ONLY ONE IS PRESENT.
10547 ;THE OPERATOR MUST INTERPRET THIS AS AN ERROR AND ATTEMPT TO FIND THE
10548 ;CAUSE (BAD CABLE, FAULTY UNIT-SELECT DECODING IN THE TRANSPORT, ETC.).
10549 ;[SINCE THE CONTROLLER CAN ONLY ACCESS UNIT 0 IF IT IS IN "STANDARD"
10550 ;MODE, THE PROGRAM WILL FORCE THE MODULE INTO EXTENDED MODE VIA THE
10551 ;WRITE SUBSYSTEM MEMORY COMMAND IN ORDER TO SCAN FOR CONNECTED TRANSPORTS.]
10552 ;
10553 ;
10554 ;THIS ROUTINE, WHEN ITS ACTIONS ARE COMPLETED, WILL EXIT BACK TO THE
10555 ;DIAGNOSTIC SUPERVISOR SO THAT IF ADDITIONAL UNITS (CONTROLLERS) ARE
10556 ;SELECTED (E.G., FROM THE INITIAL STARTUP DIALOG), THE ROUTINE WILL BE
10557 ;REENTERED SO THAT THEIR CONFIGURATIONS CAN BE PRINTED.
10558 ;
10559 074114          BGNTST
074114          T11::
10564 074114          6324:  RFLAGS  RO          ;GET OPERATOR FLAGS          TRAP  C#RFLA
074114          104421
10565 074116          001403          BEQ    104          ;BR, IF OK TO RUN
10566 074120          012700  076373          MOV    #T39NE,R0          ;"TEST NOT EXECUTED"
10567 074124          000402          BR     114          ;JUMP OUT OF TEST IF NOT
10568 074126          012700  077574          104:  MOV    #TST39ID,R0          ;TEST ID MESSAGE
10569 074132          004737  016550          114:  JSR    PC,TSTSETUP          ;DO THE COMMON SETUP
10570 074136          004737  020022          JSR    PC,CHKMAN          ;IS MANUAL INTERVENTION ALLOWED?
10571 074142          103402          BCS    204          ;BR, IF MANUAL INTERVENTION ALLOWED
10572 074144          000137  075376          JMP    644          ;JUMP TO OUT IF NOT
10573 074150          204:
10574 074150          004737  016034          JSR    PC,SOFINIT          ;DO SOFT INIT OF CONTROLLER
10575 074154          103405          BCS    254          ;BR IF SOFT INIT = OK
10579 074156          010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
10580 074160          ERRDF  ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
074160          104455          TRAP  C#ERDF
074162          002115          .WORD 1101
074164          003654          .WORD SFIERR
074166          012074          .WORD SFIMSG
10581 074170          254:  CKLOOP          ;LOOP IF SELECTED          TRAP  C#CLP1
074170          104406
10582 074172          013737  002202  076340          MOV    UNITN,T39DSW          ;SET UNIT NUMBER
10583 074200          012704  076320          MOV    #T39PK2,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
10584 074204          004737  010662          JSR    PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
10585 074210          103405          BCS    504          ;BR, IF COMMAND ISSUED OK
10589 074212          010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
10590 074214          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
074214          104456          TRAP  C#ERHRD
074216          002116          .WORD 1102
074220          005060          .WORD WRTMSG
074222          012074          .WORD SFIMSG
10591 074224          504:  CKLOOP          ;LOOP IF SELECTED          TRAP  C#CLP1
074224          104406
10592 074226          013701  075650          MOV    T398FR-12,R1          ;GET XST2 STATUS FROM MESSAGE BUFFER
10593 074232          PRINTX  #T39SFS          ;"STATE OF EXTENDED FEATURES SW ="
074232          012746  077312          MOV    #T39SFS,-(SP)
074236          012746  000001          MOV    #1,-(SP)
074242          010600          MOV    SP,R0
074244          104415          TRAP  C#PNTX
074246          062706  000004          ADD    #4,SP
10594 074252          032701  000200          BIT    #BIT7,R1          ;CHECK STATE OF E.F.S.
10595 074256          001011          BNE    1004          ;BR, IF EXT. FEA. SW. IS ON

```

TEST 11: CONFIGURATION TYPEOUT

```

10596 074260          PRINTX  #T390FF          ;" OFF"
      074260 012746 077436          MOV      #T390FF,-(SP)
      074264 012746 000001          MOV      #1,-(SP)
      074270 010600          MOV      SP,R0
      074272 104415          TRAP     C#PNTX
      074274 062706 000004          ADD      #4,SP
10597 074300          BR        110#          ;SKIP OTHER PRINT STATEMENT
10598 074302          PRINTX  #T390N          ;" ON "
      074302 012746 077445          MOV      #T390N,-(SP)
      074306 012746 000001          MOV      #1,-(SP)
      074312 010600          MOV      SP,R0
      074314 104415          TRAP     C#PNTX
      074316 062706 000004          ADD      #4,SP
10599 074322          PRINTX  #T39SBS          ;"STATE OF BUFFERING SWITCH ="
      074322 012746 077364          MOV      #T39SBS,-(SP)
      074326 012746 000001          MOV      #1,-(SP)
      074332 010600          MOV      SP,R0
      074334 104415          TRAP     C#PNTX
      074336 062706 000004          ADD      #4,SP
10600 074342          BIT        #BIT6,R1          ;CHECK STATE OF BUFFERING SW
10601 074346          BNE       120#          ;BR, IF BUFFERING IS ON
10602 074350          PRINTX  #T390FF          ;" OFF"
      074350 012746 077436          MOV      #T390FF,-(SP)
      074354 012746 000001          MOV      #1,-(SP)
      074360 010600          MOV      SP,R0
      074362 104415          TRAP     C#PNTX
      074364 062706 000004          ADD      #4,SP
10603 074370          BR        130#          ;SKIP OTHER PRINT STATEMENT
10604 074372          PRINTX  #T390N          ;" ON "
      074372 012746 077445          MOV      #T390N,-(SP)
      074376 012746 000001          MOV      #1,-(SP)
      074402 010600          MOV      SP,R0
      074404 104415          TRAP     C#PNTX
      074406 062706 000004          ADD      #4,SP
10605 074412          BIC       #177700,R1          ;ONLY LEAVE MICROCODE REV LEVEL
10606 074416          MOV      R1,T39RL          ;LOAD UP REV LEVEL
10607 074422          PRINTX  #T39MCL,T39RL          ;"MICROCODE REVISION LEVEL =000XXX"
      074422 013746 077532          MOV      T39RL,-(SP)
      074426 012746 077454          MOV      #T39MCL,-(SP)
      074432 012746 000002          MOV      #2,-(SP)
      074436 010600          MOV      SP,R0
      074440 104415          TRAP     C#PNTX
      074442 062706 000006          ADD      #6,SP
10608 074446          JSR      PC,SOFINIT          ;DO SOFT INIT OF CONTROLLER
10609 074452          BCS      140#          ;BR IF SOFT INIT = OK
10613 074454          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
10614 074456          ERDF    ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
      074456 104455          TRAP     C#ERDF
      074460 002117          .WORD   1103
      074462 003654          .WORD   SFIERR
      074464 012074          .WORD   SFIMSG
10615 074466          CKLOOP          ;LOOP IF SELECTED
      074466 104406          TRAP     C#CLP1
10616 074470          MOV      UNITN,T39DSW          ;SET UNIT NUMBER
10617 074476          MOV      #T39PK2,R4          ;SUBROUTINE NEEDS PACKET ADDRESS
10618 074502          JSR      PC,WRTCHR          ;ISSUE WRITE CHARACTERISTICS
10619 074506          BCS      150#          ;BR, IF COMMAND ISSUED OK

```

K3

TEST 11: CONFIGURATION TYPEOUT

```

10623 074510 010001          MOV      RO,R1          ;SAVE CONTENTS OF TSSR
10624 074512          ERRHRD   ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED
                                TRAP      C4ERHRD
                                .WORD    1104
                                .WORD    WRTMSG
                                .WORD    SFIMSG
10625 074522          1504:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C4CLP1
10626 074524 005737 002226    TST      EXTFEA          ;CHECK FOR EXTENDED FEATURES SW SWITCH
10627 074530 001036          BNE     1744             ;BR IF SWITCH IS ON
10628 074532 112737 000200 075631  MOVB    #200,T39BS1     ;WRITE MISCELLANEOUS CONT/READ STATUS
10629 074540 112737 000010 075630  MOVB    #10,T39BS0      ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
10630 074546 012704 075620    MOV     #T39PACKET,R4   ;WRITE SUBSYS MEM PACKET
10631 074552 010465 000000    MOV     R4,TSDB(R5)     ;ISSUE COMMAND
10632 074556 004737 016376    JSR     PC,CHKTSSR      ;WAIT FOR SSR
10633 074562 103405          BCS    1604             ;BR, IF NO ERROR
10634 074564 010001          MOV     RO,R1           ;ERROR, SAVE TSSR
10638 074566          ERRHRD   ERRNO,T39NBA,PKTSSR ;TSSR NOT CORRECT AFTER WRT, MISCELLANEOUS
                                TRAP      C4ERHRD
                                .WORD    1105
                                .WORD    T39NBA
                                .WORD    PKTSSR
10639 074576          1604:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C4CLP1
10640 074600 012704 076320    MOV     #T39PK2,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
10641          ;*****
10642          ;
10643          ;WRITE CHARACTERISTICS COMMAND (CALL TO WRTCHR)
10644          ;
10645          ;*****
10646          ;
10647 074604 004737 010662    JSR     PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
10648 074610 103405          BCS    1704             ;BR, IF COMMAND ISSUED OK
10652 074612 010001          MOV     RO,R1           ;SAVE CONTENTS OF TSSR
10653 074614          ERRHRD   ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED
                                TRAP      C4ERHRD
                                .WORD    1106
                                .WORD    WRTMSG
                                .WORD    SFIMSG
10654 074624          1704:   CKLOOP          ;SCOPE LOOP
                                TRAP      C4CLP1
10655 074626 005037 002202 1744:   CLR     UNITN           ;SET TO DRIVE 0
10656 074632 013737 002202 076340 1754:   MOV     UNITN,T39DSW    ;SET UNIT NUMBER
10657 074640 012704 076320    MOV     #T39PK2,R4     ;SUBROUTINE NEEDS PACKET ADDRESS
10658 074644 004737 010662    JSR     PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
10659 074650 103405          BCS    1804             ;BR, IF COMMAND ISSUED OK
10663 074652 010001          MOV     RO,R1           ;SAVE CONTENTS OF TSSR
10664 074654          ERRHRD   ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED
                                TRAP      C4ERHRD
                                .WORD    1107
                                .WORD    WRTMSG
                                .WORD    SFIMSG
10665 074664          1804:   CKLOOP          ;LOOP IF SELECTED
                                TRAP      C4CLP1
10666          ;
10667 074666 016501 000002 1904:   MOV     TSSR(R5),R1     ;GET TSSR STATUS
10668 074672 032701 000100    BIT     #OFL,R1        ;CHECK FOR OFF-LINE

```

TEST 11: CONFIGURATION TYPEOUT

10669	074676	001414		BEG	200‡				;	BR, IF DRIVE IS ON-LINE
10670	074700			PRINTX	#T390F2,UNITN				;	"DRIVE NUMBER XX IS OFF-LINE"
	074700	013746	002202							MOV UNITN,-(SP)
	074704	012746	076634							MOV #T390F2,-(SP)
	074710	012746	000002							MOV #2,-(SP)
	074714	010600								MOV SP,R0
	074716	104415								TRAP C‡PNTX
	074720	062706	000006							ADD #6,SP
10671	074724	000137	075260							;
10672	074730			200‡:	JMP	250‡			;	DO NOT TRY TO GET ANYMORE INFO.
	074730	013746	002202		PRINTX	#T390N2,UNITN			;	"DRIVE NUMBER XX IS ON-LINE"
	074734	012746	076700							MOV UNITN,-(SP)
	074740	012746	000002							MOV #T390N2,-(SP)
	074744	010600								MOV #2,-(SP)
	074746	104415								MOV SP,R0
	074750	062706	000006							TRAP C‡PNTX
10673	074754	013701	075644							ADD #6,SP
10674	074760	032701	000004		MOV	T39BFR-6,R1				;
10675	074764	001013			BIT	#BIT2,R1				;
10676	074766				BNE	210‡				;
	074766	013746	002202		PRINTX	#T39WPN,UNITN				;
	074772	012746	077016							;
	074776	012746	000002							;
	075002	010600								;
	075004	104415								;
	075006	062706	000006							;
10677	075012	000412								;
10678	075014			210‡:	BR	220‡				;
	075014	013746	002202		PRINTX	#T39WRT,UNITN				;
	075020	012746	076743							;
	075024	012746	000002							;
	075030	010600								;
	075032	104415								;
	075034	062706	000006							;
10679	075040	012737	125252	075736	220‡:	MOV	#125252,T39BFR-100			;
10680	075046	112737	000000	075631		MOVB	#0,T39BS1			;
10681	075054	112737	000024	075630		MOVB	#24,T39BS0			;
10682	075062	012704	075620			MOV	#T39PACKET,R4			;
10683	075066	010465	000000			MOV	R4,TSDB(R5)			;
10684	075072	012737	000144	075612		MOV	#100,T39DLY			;
10685	075100	004737	016310		222‡:	JSR	PC,WAITF			;
10686	075104	016501	000002			MOV	TSSR(R5),R1			;
10687	075110	032701	000200			BIT	#SSR,R1			;
10688	075114	001017				BNE	225‡			;
10689	075116					DELAY	250			;
	075116	012727	000250							;
	075122	000000								;
	075124	013727	002116							;
	075130	000000								;
	075132	005367	177772							;
	075136	001375								;
	075140	005367	177756							;
	075144	001367								;
10690	075146	005337	075612		DEC	T39DLY				;
10691	075152	001352			BNE	222‡				;
10692	075154	004737	016376	225‡:	JSR	PC,CHKTSSR				;
10693	075160	103405			BCS	230‡				;

TEST 11: CONFIGURATION TYPEOUT

```

10694 075162 010001          MOV    RO,R1          ;ERROR, SAVE TSSR
10698 075164          ERRHRD  ERRNO,T39NBA,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
      075164 104456          TRAP   C4ERRHRD
      075166 002124          .WORD  1108
      075170 077075          .WORD  T39NBA
      075172 012106          .WORD  PKTSSR
10699 075174          230$: CKLOOP          ;LOOP IF SELECTED
      075174 104406          TRAP   C4CLP1
10700 075176 023727 075736 125252  CMP    T398FR*100,#125252 ;DID LOC GET OVER WRITTEN
10701 075204 001013          BNE    240$          ;BR, IF IT DIDN'T GET ETC.
10702 075206          PRINTX #T39ETN,UNITN ;"DRIVE DOESN'T HAVE EXT TAPE STATUS
      075206 013746 002202          MOV    UNITN,-(SP)
      075212 012746 076541          MOV    #T39ETN,-(SP)
      075216 012746 000002          MOV    #2,-(SP)
      075222 010600          MOV    SP,RO
      075224 104415          TRAP   C4PNTX
      075226 062706 000006          ADD    #6,SP
10703 075232 000412          BR     250$          ;SKIP OVER
10704 075234          240$: PRINTX #T39ETS,UNITN ;"DRIVE HAS EXT TAPE STATUS"
      075234 013746 002202          MOV    UNITN,-(SP)
      075240 012746 076452          MOV    #T39ETS,-(SP)
      075244 012746 000002          MOV    #2,-(SP)
      075250 010600          MOV    SP,RO
      075252 104415          TRAP   C4PNTX
      075254 062706 000006          ADD    #6,SP
10705 075260 005237 002202          INC    UNITN          ;BUMP DRIVE NUMBER
10706 075264 023727 002202 000003  CMP    UNITN,#3      ;AT END OF DRIVES YET
10707 075272 001402          BEQ    631$          ;BR, IF NO MORE DRIVES
10708 075274 000137 074632          JMP    175$          ;DO NEXT DRIVE
10709 075300 012700 075402          MOV    #NUMENU,RO    ;MENU OF MANUAL INTERVENTIONS
10710 075304 012701 000001          MOV    #1,R1         ;MAXIMUM ALLOWED SELECTION
10711 075310 004737 017600          JSR    PC,GETSEL     ;GO GET THE OPERATORS SELECTION
10712 075314 010004          MOV    RO,R4         ;GET NUMBER FROM ROUTINE
10713 075316 006304          ASL    R4            ;CONVERT TO WORD OFFSET
10714 075320 000174 075324          JMP    #6,(R4)       ;JUMP TO PROPER LOOP
10715 075324 075300          6$:  .WORD  631$      ;RETYPE THE MENU
10716 075326 074114          .WORD  632$
10717 075330          63$: SETPRI #PRI00    ;RERUN THE TEST
      075330 012700 000000          MOV    #PRI00,RO
      075334 104441          TRAP   C4SPRI
10718 075336          printf #T38MS2
      075336 012746 072641          MOV    #T38MS2,-(SP)
      075342 012746 000001          MOV    #1,-(SP)
      075346 010600          MOV    SP,RO
      075350 104417          TRAP   C4PNTF
      075352 062706 000004          ADD    #4,SP
10719 075356          PRINTX #T39NFL      ;NEW LINE
      075356 012746 076370          MOV    #T39NFL,-(SP)
      075362 012746 000001          MOV    #1,-(SP)
      075366 010600          MOV    SP,RO
      075370 104415          TRAP   C4PNTX
      075372 062706 000004          ADD    #4,SP
10720 075376          64$: EXIT  TST      ;EXIT THIS SECTION
      075376 104432          TRAP   C4EXIT
      075400 002222          .WORD  L10076-.
10721
10722          ;*
          ;LOCAL TEXT MESSAGES FOR TEST

```

TEST 11: CONFIGURATION TYPEOUT

```

10723
10724
10725
10726
10727 075402 075416 075470 075516
10728 075416 012 123 105
10729 075470 012 011 060
10730 075516 011 061 011
10731 075541 011 136 103
10732 075610 000
10733
10734
10735 075612 000000
10737 075620
10739 075620
10740 075620 140006
10741 075622 075630
10742 075624 000000
10743 075626 000012
10744 075630
10745 075630 000
10746 075631 000
10747 075632 000000
10748 075634 000000
10749 075636
10750
10751
10753 076320 076320
10755 076320 140004
10757 076322 076330
10758 076324 000000
10759 076326 000012
10760
10761
10762 076330
10763 076330 075636
10764 076332 000000
10765 076334 000400
10766 076336 000000
10767 076340 000000
10769 076350 076350
10771 076350 140012
10772 076352 000000
10773
10774
10775
10777 076360 076360
10779 076360 140005
10780 076362 000000
10781 076364 000000
10782 076366 000400
10783
10784
10785
10786
10787

```

```

;-
;LOCAL STORAGE FOR THIS TEST
;-
NUMENU: .WORD 14,24,34,44,54,0
1#: .ASCIZ '<12>' SELECT OPERATION FROM FOLLOWING OPTIONS:'
2#: .ASCIZ '<12>' 0 Display This Menu'
3#: .ASCIZ ' 1 Rerun this test'
4#: .ASCIZ ' 1C To Return to Diagnostic Supervisor'
5#: .ASCIZ ''
.EVEN

T39DLY: .WORD 0 ;DELAY COUNTER FOR TEST
;=<..10>E177770

T39PACKET: ;COMMAND PACKET FOR TEST
;WRITE SUBSYSTEM MEM. CMD. ACK,CVC=1
;ADDRESS OF CHARACTERISTICS BLOCK
.WORD 140006
.WORD T39TAD
.WORD 0
.WORD 10. ;STARTING VALUE OF BLOCK SIZE
;CHARACTERISTICS DATA BLOCK
;BSEL0 BYTE
;BSEL1 BYTE
;BSEL1 WORD
;DATA
;MESSAGE BUFFER

T39TAD:
T39BS0: .BYTE 0
T39BS1: .BYTE 0
T39BS2: .WORD 0
;WORD 0
T39BFR: .BLKW 150.

T39PK2: ;=<..10>E177770
;COMMAND PACKET FOR TEST
;WRITE CHARA. MEM. CMD. ACK,CVC=1
;ADDRESS OF SELECT DATA BLOCK
.WORD 140004
.WORD T39DTA
.WORD 0
.WORD 10. ;STARTING VALUE OF BLOCK SIZE

T39DTA: ;SELECT DATA BLOCK
;ADDRESS OF MESSAGE BUFFER
.WORD T39BFR
.WORD 0
.WORD 256. ;LENGTH OF MESSAGE BUFFER
T39EAI: .WORD 0 ;EAI BIT WORD
T39DSW: .WORD 0 ;DRIVE SELECT WORD ETC
;=<..10>E177770
T39PK3: .WORD 140012
;MESSAGE BUFFER RELEASE COMMAND
.WORD 0 ;NOT USED

;WRITE TAPE PACKET
;
;=<..10>E177770
T39PK4: .WORD 140005
;WRITE, ACK, CVC=1 COMMAND
T39WR: .WORD 0 ;ADDRESS OF WRITE BUFFER
.WORD 0 ;MORE ADDRESS OF WRITE BUFFER
T39SIZ: .WORD 256. ;SIZE OF RECORD

```


TEST 11: CONFIGURATION TYPEOUT

```

10788 ;LOCAL TEXT MESSAGES FOR TEST
10789 ;-
10790
10791
10792
10793
10794 076370 045 116 000 T39NFL: .ASCIZ '#N'
10795 076373 123 164 141 T39NE: .ASCIZ 'Stand-alone Configuration Typeout Not Executed'
10796 076452 045 116 045 T39ETS: .ASCIZ '#N#A Extended Tape Status Available, Drive Number #D2'
10797 076541 045 116 045 T39ETN: .ASCIZ '#N#A Extended Tape Status NOT Available, Drive Number #D2'
10798 076634 045 116 045 T39OF2: .ASCIZ '#N#A Drive Number #D2#A Is Off-Line'
10799 076700 045 116 045 T39ON2: .ASCIZ '#N#A Drive Number #D2#A Is On-Line'
10800 076743 045 116 045 T39WRT: .ASCIZ '#N#A Drive Number #D2#A Is Write Protected'
10801 077016 045 116 045 T39WPN: .ASCIZ '#N#A Drive Number #D2#A Is NOT Write Protected'
10802 077075 127 122 111 T39NBA: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted'
10803 077151 103 157 156 T39SSR: .ASCIZ 'Contents of TSSR Incorrect After WRITE SUBSYSTEM MEMORY'
10804 077241 045 116 045 T39OUT: .ASCIZ '#N#A Type #C to Return to the Supervisor'
10805 077312 045 116 045 T39SFS: .ASCIZ '#N#A State Of Extended Features Switch ='
10806 077364 045 116 045 T39SBS: .ASCIZ '#N#A State Of Buffering Switch ='
10807 077436 045 101 040 T39OFF: .ASCIZ '#A OFF'
10808 077445 045 101 040 T39ON: .ASCIZ '#A ON'
10809 077454 045 116 045 T39MCL: .ASCIZ '#N#A M7196 Microcode Revision Level =#02'
10810
10811 077532 000000 T39RL: .WORD 0
10812 .EVEN
10813 .EVEN
10814
10815 ;+
10816 ;LOCAL STORAGE FOR THIS TEST
10817 ;-
10818
10819 077534 000000 T39DAT: .WORD 0 ;LOGICAL RESPONSE TO QUESTION
10820 077536 T39REST:
10821 077536 SAVREG ;SAVE THE REGISTERS
10822 077542 012701 075620 MOV #T39PACKET,R1 ;START OF THE PACKET
10823 077546 012721 140006 MOV #140006,(R1)- ;WRITE SUBSYSTEM MEM. WITH ACK,CVC=1
10824 077552 012721 075630 MOV #T39TAD,(R1)- ;ADDRESS OF DATA BLOCK
10825 077556 005021 CLR (R1)- ;EXTENDED ADDRESS
10826 077560 012721 000006 MOV #6,(R1)- ;SIZE OF DATA BLOCK IN BYTES
10827 077564 005021 CLR (R1)- ;CLEAR BSELO AND BSEL1
10828 077566 005021 CLR (R1)- ;CLEAR SEL2
10829 077570 005011 CLR (R1) ;CLEAR DATA AREA
10830 077572 000207 RTS PC ;RETURN
10831
10832 ;+
10833 ;LOCAL TEXT MESSAGES FOR TEST
10834 ;-
10835
10836 077574 103 157 156 TST39ID: .ASCIZ 'Configuration Typeout'
10837 .EVEN
10838 077622 .ENDTST
10839 077622 L10076: TRAP C#ETST
10840 077622 104401
10841
10842 .SBTTL TEST 12: SCOPE LOOPS

```

TEST 12: SCOPE LOOPS

```

10843
10844
10845
10846
10847
10848
10849
10850
10851
10852
10853
10854
10855
10856
10857
10858
10859
10860
10861
10862
10863
10864
10865
10866
10867
10868
10869
10870
10871
10872
10873
10874
10875
10876
10877
10878
10879

```

```

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

```

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

```

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

```

```

10884 077624 104421 101215 14: 1004: BGNTST
                                RFLAGS RO ;GET OPERATOR FLAGS T12:: TRAP C4RFLA
10885 077626 001403 BEQ 14 ;BR, IF OK TO RUN
10886 077630 012700 MOV #T4ONE,RO ;"TEST NOT EXECUTED"
10887 077634 000402 BR 1004 ;JUST EXIT IF NOT
10888 077636 012700 MOV #TST40ID,RO ;TEST ID MESSAGE
10889 077642 004737 JSR PC,TSTSETUP ;DO THE COMMON SETUP
10890 077646 004737 JSR PC,CHKMAN ;SEE IF MANUAL INTERVENTION ALLOWED
10891 077652 103402 BCS 24 ;CARRY SET IF INTERVENTION ALLOWED
10892 077654 000137 JMP 644 ;EXIT IF NO MANUAL INTERVENTION
10893 077660 004737 JSR PC,SOFINIT ;DO A SOFT INIT
10894 077664 103405 BCS 54 ;BRANCH IF OK
10895 077666 010001 MOV RO,R1 ;CONTENTS OF TSSR REGISTER
10899 077670 104455 ERDF ERRNO,SFIERR,SFIMSG ;REPORT FATAL ERROR TRAP C4ERDF
                                .WORD 1201
                                .WORD SFIERR
                                .WORD SFIMSG
10900 077700 012700 100350 54: MOV #SCMENU,RO ;MENU OF SCOPE LOOP SELECTIONS

```

D4

TEST 12: SCOPE LOOPS

10901	077704	012701	000007		MOV	#7,R1		;MAXIMUM ALLOWED SELECTION
10902	077710	004737	017600		JSR	PC,GETSEL		;GO GET THE OPERATORS SELECTION
10903	077714	005700			TST	RO		;WAS ZERO SPECIFIED ?
10904	077716	001760			BEG	24		;REPEAT MENU IF YES.
10905	077720	020027	000007		CMP	RO,#7		;EXTENDED TSSR ?
10906	077724	001015			BNE	34		;BRANCH IF NOT
10907	077726	005737	002226		TST	EXTFEA		;CHECK FOR EXTENDED FEATURES SET
10908	077732	001012			BNE	34		;BR, IF IT IS ON
10909	077734				PRINTF	#EXFMSG		;WARN OPERATOR EXTENDED FEATURES CLEAR
	077734	012746	101137					MOV #EXFMSG,-(SP)
	077740	012746	000001					MOV #1,-(SP)
	077744	010600						MOV SP,RO
	077746	104417						TRAP C4PNTF
	077750	062706	000004					ADD #4,SP
10910	077754	000137	077660		JMP	24		;GO BACK TO BASIC MENU
10911	077760	010004		34:	MOV	RO,R4		;SAVE THE MENU SELECTION
10912	077762				SETPRI	#PRI07		;RAISE THE PRIORITY
	077762	012700	000340					MOV #PRI07,RO
	077766	104441						TRAP C4SPRI
10913	077770	005037	100342		CLR	TTION		;ASSUME INTERRUPTS ARE ENABLED
10914	077774	032737	000100	177560	BIT	#100,@TTICSR		;ARE TTI INTERRUPTS ON ?
10915	100002	001005			BNE	44		;BRANCH IF YES
10916	100004	005237	100342		INC	TTION		;FLAG SET IF INTERRUPTS OFF
10917	100010	052737	000100	177560	BIS	#100,@TTICSR		;ENABLE INTERRUPTS
10918	100016	012701	000060		MOV	@TTIVEC,R1	44:	;START OF TTI VECTORS
10919	100022	011137	100344		MOV	(R1),TVECSAV		;SAVE THE CURRENT TTI VECTOR
10920	100026	012721	100250		MOV	#604,(R1)-		;SET NEW INTERRUPT ROUTINE
10921	100032	011137	100346		MOV	(R1),TPRISAV		;SAVE THE VECTOR PRIORITY
10922	100036	012711	000340		MOV	#PRI07,(R1)		;USE PRIORITY SEVEN
10923	100042				SETPRI	#PRI00		;LOWER INTERRUPT BR LEVEL
	100042	012700	000000					MOV #PRI00,RO
	100046	104441						TRAP C4SPRI
10924	100050	006304			ASL	R4		;CONVERT TO WORD OFFSET
10925	100052	000174	100056		JMP	264(R4)		;JUMP TO PROPER LOOP
10926	100056	077660		64:	.WORD	24		;RETYPE THE MENU
10927	100060	100076			.WORD	104		;TSBA READ ACCESS
10928	100062	100106			.WORD	154		;TSSR READ ACCESS
10929	100064	100120			.WORD	204		;TSSR WRITE ACCESS
10930	100066	100140			.WORD	254		;TSDB HIGH BYTE WRITE ACCESS
10931	100070	100164			.WORD	304		;TSDB LOW BYTE WRITE ACCESS
10932	100072	100210			.WORD	354		;TSDB MAINTENANCE MODE
10933	100074	100230			.WORD	404		;TSDBX WRITE ACCESS
10934								;LEAVE THE TEST
10935								
10936								
10937	100076	105065	000000	104:	CLRB	TSDB(R5)		;ENTER MAINTENANCE MODE
10938	100102	011500		124:	MOV	(R5),RO		;READ TSBA REGISTER
10939	100104	000776			BR	124		;LOOP UNTIL HALTED
10940								
10941								
10942	100106	012703	000002	154:	MOV	#TSSR,R3		;ADDRESS OF TSSR REGISTER
10943	100112	060503			ADD	R5,R3		;POINT TO TSV05'S REGISTERS
10944	100114	011300		184:	MOV	(R3),RO		;READ TSSR REGISTER
10945	100116	000776			BR	184		;LOOP UNTIL STOPPED
10946								
10947	100120	004737	017516	204:	JSR	PC,GETPAT		;READ THE DATA PATTERN
10948	100124	010001			MOV	RO,R1		;DATA PATTERN FOR LOOP

E4

TEST 12: SCOPE LOOPS

```

10949 100126 012703 000002          MOV    #TSSR,R3          ;ADDRESS OF TSSR
10950 100132 060503                ADD    R5,R3            ;POINT TO TSV05'S REGISTERS
10951 100134 010113                224:  MOV    R1,(R3)    ;WRITE DATA TO TSSR
10952 100136 000776                BR     224              ;LOOP
10953
10954
10955 100140 105065 000000          254:  CLRB   TSDB(R5)    ;ENTER MAINTENANCE MODE
10956 100144 004737 017516          JSR   PC,GETPAT        ;READ THE DATA PATTERN
10957 100150 010001                MOV    R0,R1           ;DATA PATTERN FOR LOOP
10958 100152 012703 000001          MOV    #TSDBH,R3      ;ADDRESS OF HIGH BYTE OF TSDB
10959 100156 060503                ADD    R5,R3           ;POINT TO TSV05'S REGISTERS
10960 100160 110113                274:  MOVB  R1,(R3)    ;WRITE THE DATA TO TSDB, HIGH BYTE
10961 100162 000776                BR     274              ;LOOP UNTIL STOPPED
10962
10963
10964 100164 105065 000000          304:  CLRB   TSDB(R5)    ;ENTER MAINTENANCE MODE
10965 100170 004737 017516          JSR   PC,GETPAT        ;READ THE DATA PATTERN
10966 100174 010001                MOV    R0,R1           ;DATA PATTERN FOR LOOP
10967 100176 012703 000000          MOV    #TSDB,R3       ;ADDRESS OF TSSR
10968 100202 060503                ADD    R5,R3           ;POINT TO TSV05'S REGISTERS
10969 100204 110113                324:  MOVB  R1,(R3)    ;WRITE DATA TO TSSR, LOW BYTE
10970 100206 000776                BR     324              ;LOOP UNTIL HALTED BY OPERATOR
10971
10972 100210 004737 017516          354:  JSR   PC,GETPAT        ;READ THE DATA PATTERN
10973 100214 010001                MOV    R0,R1           ;DATA PATTERN FOR LOOP
10974 100216 012703 000000          MOV    #TSDB,R3       ;SELECT TSDB
10975 100222 060503                ADD    R5,R3           ;POINT TO TSV05'S REGISTERS
10976 100224 010113                374:  MOV    R1,(R3)    ;WRITE THE DATA PATTERN
10977
10978 100226 000776                BR     374              ;LOOP UNTIL HALTED
10979
10980 100230 004737 017516          404:  JSR   PC,GETPAT        ;READ THE DATA PATTERN
10981 100234 010001                MOV    R0,R1           ;SAVE THE DATA PATTERN
10982 100236 012703 000003          MOV    #TSSRH,R3      ;BYTE ADDRESS OF TSSR, HIGH BYTE
10983 100242 060503                ADD    R5,R3           ;POINT TO TSV05'S REGISTERS
10984 100244 110113                424:  MOVB  R1,(R3)    ;WRITE THE DATA TO REGISTER
10985 100246 000776                BR     424              ;LOOP UNTIL HALTED
10986
10987
10988
10989
10990
10991
10992 100250 010046                604:  MOV    R0,-(SP)        ;SAVE WORK REGISTER
10993 100252 113700 177562          MOVB  @TTIBFR,R0      ;GET THE OPERATOR INPUT
10994 100256 042700 000200          BIC   #200,R0         ;STRIP OFF PARITY BIT
10995 100262 122700 000015          CMPB  #15,R0          ;IS IT A CARRIAGE RETURN ?
10996 100266 001021                BNE   614              ;JUST EXIT IF NOT
10997 100270 012766 077660 000002          MOV    #24,2(SP)     ;RETURN TO MASTER MENU
10998 100276 005066 000004          CLR   4(SP)           ;FORCE PRIORITY ZERO
10999 100302 013737 100344 000060          MOV    TVECSAV,@TTIVEC ;RESTORE SUPERVISOR VECTOR
11000 100310 013737 100346 000062          MOV    TPRISAV,@TTIVEC.2 ;RESTORE SUPERVISOR PRIORITY
11001 100316 005737 100342                TST   TTION           ;ARE SUPERVISOR INTERRUPTS ENABLED ?
11002 100322 001403                BEQ   614              ;BRANCH IF YES
11003 100324 042737 000100 177560          614:  BIC   #100,@TTICSR   ;TURN OFF TTI INTERRUPTS
11004 100332 012600                MOV    (SP)-,R0       ;RESTORE REGISTER
11005 100334 000002                RTI                    ;RETURN FROM INTERRUPT

```

F4

TEST 12: SCOPE LOOPS

```

11006
11007 100336
11008 100336 104432
      100336 000736
      64$: EXIT TST ;EXIT THE TEST
      63$: TRAP C#EXIT L10077-.
      .WORD
11009 ;65$: JMP 200 ;RETURN TO SUPERVISOR
11010 ;*****JUMP TO 200 IS NOT SUPPORTED ON XXDP- VERSION 2.X.
11011 ;*
11012 ;LOCAL STORAGE FOR THIS TEST
11013 ;-
11014
11015 100342 000000 TTION: .WORD 0 ;WORD SET IF SUPERVISOR TTI INTER OFF
11016 100344 000000 TVECSAV: .WORD 0 ;SAVE TTI VECTOR
11017 100346 000000 TPRISAV: .WORD 0 ;SAVE TTI PRIORITY
11018
11019
11020 ;*
11021 ;MENU FOR OPERATOR INPUT FOR SCOPE LOOPS
11022 ;-
11023
11024 .EVEN
11025 100350 100402 100455 100503 SCMENU: .WORD 1#,2#,3#,4#,5#,6#
11026 100364 100654 100712 100760 .WORD 7#,8#,9#,10#,11#,12#,0
11027
11028
11029 100402 012 123 105 1#: .ASCIZ <12>'SELECT SCOPE LOOP FROM FOLLOWING OPTIONS:'
11030 100455 012 011 060 2#: .ASCIZ <12>' 0 Display This Menu'
11031 100503 011 061 011 3#: .ASCIZ ' 1 TSBA Read Access'
11032 100527 011 062 011 4#: .ASCIZ ' 2 TSSR Read Access'
11033 100553 011 063 011 5#: .ASCIZ ' 3 Initialize (TSSR Write Access)'
11034 100615 011 064 011 6#: .ASCIZ ' 4 TSDB High Byte Write Access'
11035 100654 011 065 011 7#: .ASCIZ ' 5 TSDB Low Byte Write Access'
11036 100712 011 066 011 8#: .ASCIZ ' 6 TSDB Maintenance Mode Write Access'
11037 100760 011 067 011 9#: .ASCIZ ' 7 TSDBX (TSSR High Byte) Write Access'
11038 101027 011 136 103 10#: .ASCIZ ' ?C To Return to Diagnostic Supervisor'
11039 101076 000 11#: .ASCIZ ''
11040 101077 124 171 160 12#: .ASCIZ 'Type RETURN To Stop Scope Loops'
11041 101137 045 116 045 EXFMMSG: .ASCIZ 'MSA *** Extended Features Switch Not On *** '
11042 101215 123 164 141 T4ONE: .ASCIZ 'Stand-alone Scope Loops Not Executed'
11043 101262 123 143 157 TST40ID: .ASCIZ 'Scope Loops'
11044
11045 101276 .EVEN
      101276 ENDTST
      101276 L10077: TRAP C#ETST
11046 101300 104401
      101300 ENDMOD
11048 .TITLE TSV6 - PARAMETER CODING
11054
11059
11065
11066 101300 BGNMOD TSV6
      101300 TSV6::
11067 .SBTTL HARDWARE PARAMETER CODING SECTION
11068
11069
11070 ;*
11071 ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
11072 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE

```

HARDWARE PARAMETER CODING SECTION

```

11073      ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
11074      ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
11075      ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
11076      ; WITH THE OPERATOR.
11077      ; --
11078      101300      BGNHRD
11079      101300      000010      .WORD L10100-L$HARD/2
11080      101302      L$HARD::
11080      101302      GPRMA      HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
11080      101302      .WORD      T$CODE
11080      101304      101322      .WORD      HPM1
11080      101306      160010      .WORD      T$LQIM
11080      101310      177776      .WORD      T$HILIM
11081      101312      GPRMA      HPM2,2,0,0,776,YES      ;GET VECTOR ADDRESS.
11081      101312      .WORD      T$CODE
11081      101314      101356      .WORD      HPM2
11081      101316      000000      .WORD      T$LQIM
11081      101320      000776      .WORD      T$HILIM
11082      ;GPRMD      HPM3,4,0,340,0,7,YES      ;GET INTERRUPT PRIORITY.
11083      101322      ENDHRD
11083      .EVEN
11084      101322      L10100:
11084      101322      104      105      126      HPM1:      .ASCIZ      'DEVICE ADDRESS (TSBA/TSDB) '
11085      101356      111      116      124      HPM2:      .ASCIZ      'INTERRUPT VECTOR '
11086      101402      111      116      124      HPM3:      .ASCIZ      'INTERRUPT PRIORITY '
11087      .EVEN

```

SOFTWARE PARAMETER CODING SECTION

```

11089                                     .SBTTL SOFTWARE PARAMETER CODING SECTION
11090
11091                                     ;**
11092                                     ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
11093                                     ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
11094                                     ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
11095                                     ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
11096                                     ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
11097                                     ; WITH THE OPERATOR.
11098                                     ;**
11099                                     ;--
11099 101432                                     BGNSFT
11099 101432 000003                             .WORD L10101-L$SOFT/2
11099 101434
11100                                     L$SOFT::
11101 101434                                     ; GPRM1 SPM1,0,-1,YES ; GET TRANSPORT TEST FLAG.
11101 101434 001130                             GPRM1 SPM4,2,-1,YES ; GET ITERATION CONTROL.
11101 101436 101472                             .WORD T$CODE
11101 101440 177777                             .WORD SPM4
11102                                     ; .WORD -1
11103                                     ; GPRM6 SPM6,4,D,7777,0,7777,YES ; GET LOCAL ERROR LIMIT
11104 101442                                     ; GPRM7 SPM7,6,D,7777,0,7777,YES ; GET GLOBAL ERROR LIMIT
11104                                     ENDSFT
11104                                     .EVEN
11105                                     L10101:
11106 101442 105 116 101 SPM1: .ASCIZ 'ENABLE TRANSPORT TESTS '
11107 101472 111 116 110 SPM4: .ASCIZ 'INHIBIT ITERATIONS '
11108 101522 120 105 122 SPM6: .ASCIZ 'PER TEST ERROR LIMIT '
11109 101552 120 105 122 SPM7: .ASCIZ 'PER UNIT ERROR LIMIT '
11110                                     .SBTTL PATCH AREA
11111
11112                                     ;
11113                                     ; FINALLY A GENEROUS PATCH AREA.
11114                                     ;
11115                                     ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
11116                                     ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
11117                                     ;
11118                                     ;
11119 101602                                     PATCH::
11120                                     .BLKW 32.
11121 101602
11122                                     .B, !377-1
11124 102000 102000                             LASTAD ;SET LAST USED ADDRESS.
11126 102000                                     .EVEN
11126 102000 000000                             .WORD 0
11126 102002 000000                             .WORD 0
11126 102004
11127 102004                                     L$LAST::
11128 000001                                     ENDMOD
11128                                     .END

```

Symbol table

ADDSSR	012166	G	C#AUTO=	000061	DEVNRD	022705	FUSI	004121	INTMAS	016204	
ADR	= 000020	G	C#BRK =	000022	DEVNXR	022623	F#AU =	000015	INTR	016256	G
AMBTSS	006637		C#BSEG=	000004	DEVNL	022553	F#AUTO=	000020	INTREC	002224	G
ASSEMB=	000010		C#BSUB=	000002	DEVSUM	022516	F#BGN=	000040	INTVEC	016206	
A1716	= 000003		C#CEFG=	000045	DFPTBL	002156	F#CLEA=	000007	INTX	004302	
BADDAT	003160	G	C#CLCK=	000062	DIAGMC=	000000	F#DU =	000016	INVERT	020504	G
BADSSR	015740	G	C#CLEA=	000012	DICEB =	000001	F#END =	000041	IOKCKI=	000200	
BDVPCR=	177520	G	C#CLOS=	000035	DSBINT	016244	F#HARD=	000004	IOKSTP=	000001	
BENBSW	002232	G	C#CLP1=	000006	DUAD12	004645	F#HW =	000013	IPRI	002212	G
BIE	= 040000		C#CVEC=	000036	DUFLG	003114	F#INIT=	000006	ISR	= 000100	G
BIT0	= 000001	G	C#DCLN=	000044	DUMMY	003064	F#JMP =	000050	IVEC	002210	G
BIT00	= 000001	G	C#DODU=	000051	EF.CON=	000036	F#MOD =	000000	IXE	= 004000	G
BIT01	= 000002	G	C#DRPT=	000024	EF.NEW=	000035	F#MSG =	000011	I#AU =	000041	
BIT02	= 000004	G	C#DU =	000053	EF.PWR=	000034	F#PROT=	000021	I#AUTO=	000041	
BIT03	= 000010	G	C#EDIT=	000003	EF.RES=	000037	F#PWR =	000017	I#CLN =	000041	
BIT04	= 000020	G	C#ERDF=	000055	EF.STA=	000040	F#RPT =	000012	I#DU =	000041	
BIT05	= 000040	G	C#ERHR=	000056	EMAXDU	017037	F#SEG =	000003	I#HRD =	000041	
BIT06	= 000100	G	C#ERRO=	000060	EN	= 000000	F#SOFT=	000005	I#INIT=	000041	
BIT07	= 000200	G	C#ERSF=	000054	ENAINI	016212	F#SRV =	000010	I#MOD =	000041	
BIT08	= 000400	G	C#ERSO=	000057	ENVIRN	020152	F#SUB =	000002	I#MSG =	000041	
BIT09	= 001000	G	C#ESCA=	000010	EPRTSW	002200	F#SW =	000014	I#PROT=	000040	
BIT1	= 000002	G	C#ESEG=	000005	EPRT1	006362	F#TEST=	000001	I#PTAB=	000041	
BIT10	= 002000	G	C#ESUB=	000003	EPRT2	006362	GDDAT	003162	I#PWR =	000041	G
BIT11	= 004000	G	C#ETST=	000001	ERCM	011773	GERRMA	002174	I#RPT =	000041	G
BIT12	= 010000	G	C#EXIT=	000032	ERRHI	002240	GETPAT	017516	I#SEG =	000041	G
BIT13	= 020000	G	C#GETB=	000026	ERRK	017016	GETSEL	017600	I#SETU=	000041	G
BIT14	= 040000	G	C#GETW=	000027	ERRLO	002242	G#CNT0=	000200	I#SFT =	000041	G
BIT15	= 100000	G	C#GMAN=	000043	ERRNO =	002261	G#DELM=	000372	I#SRV =	000041	G
BIT2	= 000004	G	C#GPHR=	000042	ERRVEC=	000004	G#DISP=	000003	I#SUB =	000041	G
BIT3	= 000010	G	C#GPLO=	000030	ERTABE	003400	G#EXCP=	000400	I#TST =	000041	G
BIT4	= 000020	G	C#GPRI=	000040	ERTABL	003200	G#HILI=	000002	J#JMP	= 000167	
BIT5	= 000040	G	C#INIT=	000011	ESUM	017020	G#LOLI=	000001	KIPAR0=	172340	
BIT6	= 000100	G	C#INLP=	000020	EVL	= 000004	G#ND =	000000	KIPAR1=	172342	
BIT7	= 000200	G	C#MANI=	000050	EXBCNT=	000010	G#OFFS=	000400	KIPAR2=	172344	
BIT8	= 000400	G	C#MEM =	000031	EXFMSG	101137	G#OFSI=	000376	KIPAR3=	172346	
BIT9	= 001000	G	C#MSG =	000023	EXPBRE	015542	G#PRMA=	000001	KIPAR4=	172350	
BOE	= 000400	G	C#OPEN=	000034	EXPD	002234	G#PRMD=	000002	KIPAR5=	172352	
BRINIT	004461		C#PNTB=	000014	EXPGOT	004535	G#PRML=	000000	KIPAR6=	172354	
BSELO	= 000000		C#PNTF=	000017	EXPGT2	004571	G#RADA=	000140	KIPAR7=	172356	
BSEL1	= 000001		C#PNTS=	000016	EXPMSG	002324	G#RADB=	000000	KIPDR0=	172300	
CHKAMB	016104		C#PNTX=	000015	EXPREC	015534	G#RADD=	000040	KIPDR1=	172302	
CHKMAN	020022	G	C#QIO =	000377	EXTA	005774	G#RADL=	000120	KIPDR2=	172304	
CHKTSS	016376		C#RDBU=	000007	EXTEND	005772	G#RAD0=	000020	KIPDR3=	172306	
CKDROP	017242		C#REFG=	000047	EXTFEA	002226	G#XFER=	000004	KIPDR4=	172310	
CKEMAX	017142		C#RESE=	000033	E#END =	002100	G#YES =	000010	KIPDR5=	172312	
CKMSG	011420	G	C#REVI=	000003	E#LOAD=	000035	HIADDR=	001400	KIPDR6=	172314	
CKMSG2	011540	G	C#RFLA=	000021	FATERR=	000060	HOE	= 100000	KIPDR7=	172316	G
CKRAM	011154	G	C#RPT =	000025	FATFLG	002222	HPM1	101322	KTENAB	003136	G
CKRAM2	011264	G	C#SEFG=	000046	FERCH	011762	HPM2	101356	KTFLG	003134	G
CHDPKT	020650	G	C#SPRI=	000041	FIFEXP	012230	HPM3	101402	KTINIT	020332	
CONFIG	017310		C#SVEC=	000037	FIF1MS	012302	IBE	= 010000	KTOFF	017334	
COUNT	002312	G	C#TPRI=	000013	FIF2MS	012351	IDU	= 000040	KTON	017316	
CSRADD	002206	G	DATA	002314	FNOINT	004217	IER	= 020000	LERRMA	002172	G
CTAB	003166	G	DATASC	017554	FORCER	002176	IFAU	= 004260	LERRND=	000000	
CTABE	003200	G	DEBUGM	011672	FREE	003126	INCERK	017104	LISTAL=	000001	
CTABM	003166	G	DEVCNT	002220	FREEM	003132	INTCPC	016210	LOE	= 040000	G
C#AU =	000052		DEVDR0	022766	FRESIZ	003130	INTFLA	016205	LOOPCN	002216	G

Symbol table

LOOPCO	013166	L10001	002176	L10073	066254	ONEFIL	= 000000	PRMSGE	014612	G
LOOPFL	003164	L10002	005770	L10074	064414	O4APTS	= 000000	PRMSGO	014772	
LOT	= 000010	L10003	012104	L10075	074112	O4AU	= 000001	PRMSG1	015037	
L\$ACP	002110	L10004	012122	L10076	077622	O4BGNR	= 000001	PRMSG2	015075	
L\$APT	002036	L10005	012140	L10077	101276	O4BGNS	= 000001	PROASC	014460	
L\$AU	021742	L10006	012146	L10100	101322	O4DU	= 000001	PR1ASC	014525	
L\$AUT	002070	L10007	012164	L10101	101442	O4ERRT	= 000000	PST32W	003154	G
L\$AUTO	022146	L10010	012202	MEMADD	014014	O4GNSW	= 000001	PUNIT	021674	
L\$CCP	002106	L10011	012226	MEMCK	020666	O4POIN	= 000001	PW.D11	= 000021	
L\$CLEA	022226	L10012	012300	MENASC	017771	O4SETU	= 000000	PW.D13	= 000022	
L\$CO	002032	L10013	012450	MENERR	017716	PASRPT	021444	PW.D22	= 000020	
L\$DEPO	002011	L10014	013164	MENRES	020020	PATCH	101602	PW.NOP	= 000000	
L\$DESC	003412	L10015	014012	MIMENU	072772	PATDAT	017552	PW.N01	= 000023	
L\$DESP	002076	L10016	014034	MMVEC	= 000250	PC.ERA	= 002400	PW.RDE	= 000024	
L\$DEVP	002060	L10017	015540	MSA.FR	= 000006	PC.IER	= 002000	PW.RDR	= 000001	
L\$DISP	002124	L10020	015546	MSA.NO	= 000000	PC.N00	= 001000	PW.RDS	= 000005	
L\$DLY	002116	L10021	015554	MSA.NR	= 000004	PC.REL	= 000000	PW.RFI	= 000003	
L\$DTP	002040	L10022	015566	MSA.V0	= 000002	PC.REW	= 000400	PW.WCT	= 000006	
L\$DTYP	002034	L10023	015610	MSGEXP	012204	PKBCNT	= 000006	PW.WFI	= 000004	
L\$DU	022040	L10024	015636	MSGLOO	013124	PKHI	= 000004	PW.WFM	= 000007	
L\$DUT	002072	L10025	015776	MSGSTA	012410	PKLOW	= 000002	PW.WPI	= 000010	
L\$DVTY	003404	L10026	016306	MSGSUB	014002	PKTADD	007556	PW.WNP	= 000011	
L\$EF	002052	L10030	021672	MS.ATT	= 000006	PKTFRM	007520	PW.WTR	= 000002	
L\$ENVI	002044	L10031	022036	MS.EXT	= 000200	PKTGET	012124	P.ACK	= 100000	
L\$ETP	002102	L10032	022144	MS.RSD	= 000001	PKTMES	012150	P.CMD	= 000037	
L\$EXP1	002046	L10033	022224	MS.RSF	= 000020	PKTRAM	004747	P.CONT	= 000012	
L\$EXP4	002064	L10034	022252	MS.RST	= 000010	PKTSSR	012106	P.CVC	= 040000	
L\$EXP5	002066	L10035	022514	M8186	005556	PNT	= 001000	P.FMT	= 000140	
L\$HARD	101302	L10036	024014	M8189	005647	PRAMPK	014036	P.FORM	= 000011	
L\$HIME	002120	L10037	026006	NBA	= 002000	PRASC	014563	P.GETS	= 000017	
L\$HPCP	002016	L10040	024270	NEWPAS	021400	PRBEXP	015530	P.IE	= 000200	
L\$HPTP	002022	L10041	024534	NODEV	003116	PRBMSG	015376	P.INIT	= 000013	
L\$HW	002156	L10042	031402	NOEXTF	027576	PRBREC	015532	P.MODE	= 007400	
L\$ICP	002104	L10043	026362	NOINIT	004337	PRBTOT	015463	P.OPP	= 020000	
L\$INIT	021146	L10044	026652	NOINTR	004223	PRBYTE	015162	P.POSI	= 000010	
L\$LADP	002026	L10045	027150	NOITS	002170	PRI	= 002000	P.READ	= 000001	
L\$LAST	102004	L10046	027602	NOMAN	020056	PRIADD	010162	P.SWB	= 010000	
L\$LOAD	002100	L10047	034172	NOMEM	005462	PRIAO	010232	P.WRIT	= 000005	
L\$LUN	002074	L10050	031736	NP.IR	= 000200	PRIBX0	007614	P.WRTC	= 000004	
L\$MREV	002050	L10051	032330	NP.L00	= 000040	PRIEQU	010062	P.WRTS	= 000006	
L\$NAME	002000	L10052	032736	NP.OUT	= 000100	PRIPKT	007372	GVP	002204	G
L\$PRIO	002042	L10053	037764	NP.WRP	= 000020	PRIRAM	010070	RAMASC	014216	
L\$PROT	021136	L10054	035256	NSI	004154	PRITAD	010276	RAMDAT	002244	G
L\$PRT	002112	L10055	036210	NSINIT	004411	PRITSS	006026	RAMERR	015550	G
L\$REPP	002062	L10056	050076	NUL	004531	PRITO	010360	RAMEXP	015570	G
L\$REV	002010	L10057	040270	NULCR	004532	PRIT1	010423	RAMFOR	010120	
L\$RPT	022254	L10060	041500	NUMENU	075402	PRIXOR	007744	RAMSIZ	002304	G
L\$SOFT	101434	L10061	043040	NXM	= 004000	PRI00	= 000000	RAMTAD	015556	G
L\$SPC	002056	L10062	043416	NXMFLG	003140	PRI01	= 000040	RCVHIA	002306	G
L\$SPCP	002020	L10063	044670	NXMI	003144	PRI02	= 000100	RCVLOA	002310	G
L\$SPTP	002024	L10064	045734	NXMLO	003142	PRI03	= 000140	RDERR	005210	
L\$STA	002030	L10065	051356	NXMTST	021042	PRI04	= 000200	RECMG	002470	G
L\$SW	002166	L10066	062204	NXR	003742	PRI05	= 000240	RECV	002236	G
L\$TEST	002114	L10067	053362	NXRERR	005740	PRI06	= 000300	REGSAV	017462	
L\$TIPL	002014	L10070	054650	NXRX	004001	PRI07	= 000340	RETERR	005374	
L\$UNIT	002012	L10071	056120	NXTU	021412	PRMESS	014302	REV	002230	G
L10000	002164	L10072	057364	OFL	= 000100	PRMNO	002322	REWIND	011054	G

Symbol table

RMCHBE=	000167	S1.IID=	004000	TST40I	101262	T10	066256	G	T15BFR	033002
RMCHEN=	000200	S1.I1R=	020000	TSV2	002000	T11	074114	G	T15BF2	033470
RMMSGB=	000215	S1.I2R=	040000	TSV3	002176	T12	077624	G	T15BS0	033470
RMMSGC=	000234	S1.PAR=	100000	TSV4	021136	T12BFR	027672		T15BS1	033471
RMPKTB=	000201	S2.ATI=	000010	TSV5	023036	T12BKG	030347		T15DAT	032770
RMPKTE=	000210	S2.BTI=	000004	TSV6	101300	T12BLK	027724		T15L00	031434
RMR	= 010000	S2.DIM=	000200	TTIBFR=	177562	T12CHA	031340		T15PAC	032760
RMPACK	011150	S2.ILW=	000100	TTICSR=	177560	T12CKR	031120	G	T15PK2	033460
SC	= 100000	S2.INR=	000020	TTION	100342	T12CON	030726		T15RES	034062
SCE	= 020000	S2.OUT=	000040	TTION2	071036	T12DAT	027660		T15RT2	034134
SCHERR	005302	S2.UND=	000003	TTIVC=	000060	T12DPR	030526		T15SSR	033476
SCME	005015	TBLEND=	003064	TVECSA	100344	T12GET	030106		T15S2	033472
SCMENU	100350	TCOASC	006500	TVSAV2	071040	T12HIA	027712		T15S3	033474
SDELAY	010660	TCOCOD	006700	T\$ARGC=	000001	T12KT	027720		T16BEN	037650
SELASC	017764	TEMP1	003120	T\$CODE=	001130	T12LOA	027714		T16BFR	037622
SELDAT=	000004	TEMP2	003122	T\$ERRN=	002261	T12L00	026036		T16BFS	037642
SEL2	= 000002	TERCLS=	000016	T\$EXCP=	000000	T12MSG	030251		T16CLR	037466
SETMAP	017356	TESTNO=	000014	T\$FLAG=	000040	T12NIN	030435		T16DAT	037610
SETU	021476	TEXASC	006437	T\$GMAN=	000000	T12NXM	030617		T16DT2	037660
SFFMSG	012142	TFCASC	006541	T\$HILI=	000776	T12PAC	027650		T16D01	036242
SFHERR	003707	TIMEXP	015612	T\$LAST=	000001	T12PAR	027716		T16D28	036244
SFIERR	003654	TIMSGO	015640	T\$LOLI=	000000	T12SET	031276		T16D53	036246
SFIMSG	012074	TINERR	012061	T\$LSYM=	010000	T12SMR	031230		T16D78	036250
SFPTBL	002166	TMPBFR	002634	T\$LTNO=	000014	T12TBE	030056		T16LEN	036572
SIFLAG	003156	TNAM	016744	T\$NEST=	177777	T12WRT	030162		T16L00	034212
SIMSG	012026	TPRISA	100346	T\$NS0 =	000000	T121L0	026150		T16PAC	037600
SKIPT	003402	TPSAV2	071042	T\$NS1 =	000005	T122L0	026504		T16PK2	037650
SOFINI	016034	TRANST	002166	T\$NS2 =	000002	T123L0	026724		T16REJ	036704
SPACE	010470	TSBA	= 000000	T\$NS3 =	000003	T124L0	027346		T16RES	037420
SPM1	101442	TSBAH =	000001	T\$PTNU=	000000	T124TS	027722		T16SEX	037560
SPM4	101472	TSDB =	000000	T\$SAVL =	177777	T13BFR	023452		T16SRD	037512
SPM6	101522	TSDBH =	000001	T\$SEGL =	177777	T13DAT	023440		T16SSR	036322
SPM7	101552	TSECOD	007240	T\$SEKO=	010000	T13L00	023054		T16TSB	036470
SR0	= 177572	TSREJ =	000006	T\$SUBN=	000000	T13MEM	023545		T16T01	037021
SR1	= 177574	TSSDEF	006610	T\$TAGL =	177777	T13NBA	023604		T16T28	037120
SR2	= 177576	TSSR =	000002	T\$TAGN=	010102	T13PAC	023430		T16T53	037220
SR3	= 172516	TSSRBI	003504	T\$TEMP=	000000	T13RES	023746		T16T78	037320
SSR	= 000200	TSSRFO	006417	T\$TEST=	000014	T13SSR	023657		T16MMI	037532
STATCO	012452	TSSRM =	000003	T\$TSTM=	177777	T14BFR	024566		T16SS	036357
SVCGBL=	000000	TSSX	004022	T\$TSTS=	000001	T14BS0	024560		T163SS	036423
SVCINS=	000000	TSTBLK	002754	T\$AU =	010031	T14BS1	024561		T17BEN	047752
SVCSUB=	000001	TSTCNT	002214	T\$AUT =	010033	T14BS2	024562		T17BFR	047632
SVCTAG=	000000	TSTEND	016760	T\$CLE =	010034	T14DAT	024560		T17BFS	047652
SVCTST=	000001	TSTFLA	002316	T\$DU =	010032	T14DTA	025200		T17CLE	047536
S\$LSYM=	010000	TSTL00	016516	T\$HAR=	010100	T14L00	024034		T17CLR	047350
SO.IDB=	000010	TSTPTR	002320	T\$HMJ =	010000	T14NBA	025212		T17DAT	047620
SO.IFB=	000002	TSTSET	016550	T\$INI =	010030	T14NIN	025453		T17DT2	047770
SO.IFP=	000001	TST12I	030060	T\$MSG=	010025	T14PAC	024550		T17EXE	046114
SO.ILD=	000020	TST13I	023472	T\$PRO=	010027	T14PK2	025170		T17EXP	045772
SO.ION=	000040	TST14I	025631	T\$RPT=	010035	T14RES	025676		T17EXS	046012
SO.IRD=	000100	TST15I	034043	T\$SEG=	010000	T14RST	025734		T17L00	040014
SO.IRW=	000004	TST16I	036252	T\$SOF=	010101	T14SSR	025363		T17MSK	045766
SO.ISP=	000200	TST17I	046216	T\$SRV=	010026	T14TSB	025545		T17PAC	047610
S1.ICE=	002000	TST18I	050606	T\$SUB=	010074	T142RE	025266		T17PK2	047760
S1.IE0=	010000	TST19I	057572	T\$SW =	010001	T15AM2	033552		T17RFI	047516
S1.IFM=	001000	TST20I	064572	T\$TES=	010077	T15AM3	033653		T17RSF	047414
S1.IHE=	000400	TST39I	077574	T1	023036	T15AM4	033755		T17SET	047556

Symbol table

T17SNP	047436	T19SSR	057633	T3	026010	G	T39NBA	077075	WC.ISR=	000020	
T17SRD	047374	T19WCT	061532	T38FLG	003152	G	T39NE	076373	WF.IED=	000010	
T17SSR	046235	T19WFI	061476	T3.1	026036		T39NFL	076370	WF.IER=	000004	
T17MFD	046114	T19WFM	061552	T3.2	026376		T39OFF	077436	WF.IHI=	000200	
T17WFI	047462	T19WST	061133	T3.3	026666		T39OF2	076634	WF.IRE=	000040	
T171CM	046555	T191CM	060270	T3.4	027164		T39ON	077445	WF.IMF=	000020	
T172CM	046637	T192CM	060352	T38ASC	074045		T39ON2	076700	WF.IWR=	000100	
T172SS	046272	T192SS	057670	T38ASN	074064		T39OUT	077241	WF.I3R=	000002	
T173CM	046733	T193CM	060446	T38ASO	073742		T39PAC	075620	WF.I4R=	000001	
T173SS	046336	T193SS	057734	T38AS1	074007		T39PK2	076320	WRTCHR	010662	G
T174CM	047017	T194SS	060001	T38BFR	071066		T39PK3	076350	WRTERR	005115	
T174SS	046403	T195CM	060534	T38BSO	071060		T39PK4	076360	WRTMSG	005060	
T175CM	047102	T195SS	060044	T38BS1	071061		T39RES	077536	WSMBK	020660	G
T175SS	046446	T196CM	060610	T38BS2	071062		T39RL	077532	XFERAS	016000	
T176CM	047156	T196SS	060110	T38CNT	074110		T39SBS	077364	XNKM	016436	
T176SS	046512	T197CM	060673	T38DAT	073400		T39SFS	077312	XORBF0	007676	
T177CM	047264	T197SS	060153	T38DLY	071044		T39SIZ	076366	XORFOR	010014	
T18BFR	051312	T198CM	060761	T38DSW	071570		T39SSR	077151	XST0	= 000006	G
T18CMP	051013	T198SS	060222	T38DTA	071560		T39TAD	075630	XST1	= 000010	G
T18DAT	051300	T199CM	061050	T38EAI	071566		T39MPN	077016	XST2	= 000012	G
T18DT2	051350	T2	024016	T38EB	071542		T39MR	076362	XST3	= 000014	G
T18EXP	050556	T2.1	024034	T38ID	072745		T39MRT	076743	XST4	= 000016	G
T18L00	050116	T2.2	024304	T38INT	072336		T4	031404	XSOBOT=	000002	
T18MSK	050552	T20BEN	066132	T38MBP	073440		T4.1	031434	XSOEOT=	000001	
T18PAC	051270	T20BFR	066012	T38MSG	072705		T4.2	031740	XSOIE =	000040	
T18PK2	051340	T20BFS	066032	T38MS1	072546		T4.3	032332	XSOILA=	000400	
T18SET	051160	T20CLE	065724	T38MS2	072641		T4ONE	101215	XSOILC=	001000	
T18SMI	051136	T20CLR	065512	T38MS3	071705		T5	034174	XSOLET=	020000	
T18SRD	051116	T20CMP	065417	T38NBA	072172		T5.1	034212	XSONOT=	000200	
T18SSR	050645	T20DAT	066000	T38NE	071630		T5.2	035272	XSONEF=	002000	
T18XS	050602	T20DT2	066150	T38OFL	072422		T6	037766	XSOONL=	000100	
T182SS	050702	T20EXE	064572	T38ONL	072366		T6.1	040014	XSOPED=	000010	
T183SS	050746	T20EXP	064452	T38PAC	071050		T6.2	040304	XSORLL=	010000	
T19BEN	062062	T20EXS	064472	T38PK2	071550		T6.3	041514	XSORLS=	040000	
T19BFC	057406	T20L00	062224	T38PK3	071600		T6.4	043054	XSOTMK=	100000	
T19BFR	061742	T20MSK	064446	T38PK4	071620		T6.5	043432	XSOVCK=	000020	
T19BFS	061762	T20PAC	065770	T38RES	073402		T6.6	044704	XSONLE=	004000	
T19CLE	061610	T20PK2	066140	T38SIZ	071626		T7	050100	XSONLK=	000004	
T19CLR	061344	T20RFI	065576	T38SSR	072246		T8	051360	XXCOMP	003124	G
T19CNV	061654	T20RSF	065316	T38SST	072456		T8.1	051376	XALMA=	000000	
T19DAT	061730	T20SET	065744	T38TAD	071060		T8.2	053376	XIFALS=	000040	
T19DT2	062100	T20SEX	065706	T38MLE	072125		T8.3	054664	XIFFS=	000400	
T19EXE	057572	T20SRD	065536	T38MR	071622		T8.4	056134	XITRUE=	000020	
T19EXP	057452	T20SSR	064621	T38MRL	072064		T9	062206	X1.COR=	020000	
T19XS	057472	T20SMP	065207	T38MRT	072000		T9.1	062224	X1.DLT=	100000	
T19L00	051376	T20WFI	065616	T39BFR	075636		UAM	= 000200	X1.MBZ=	017375	
T19MSK	057446	T20WFM	065652	T39BSO	075630		UNITN	= 002202	X1.RBP=	000400	
T19PAC	061720	T20WMI	065672	T39BS1	075631		UNREC	= 000006	X1.SPA=	040000	
T19PK2	062070	T20WNP	065556	T39BS2	075632		USI	= 004125	X1.UNC=	000002	
T19PRE	057404	T202SS	064656	T39DAT	077534		WAITF	= 016310	X2.BUF=	000100	
T19RFI	061456	T203SS	064722	T39DLY	075612		WC.IFA=	000200	X2.EXT=	000200	
T19RSF	061410	T204SS	064767	T39DSW	076340		WC.IFE=	000002	X2.OPM=	100000	
T19RST	061240	T205SS	065032	T39DTA	076330		WC.IGO=	000001	X2.RCE=	040000	
T19SET	061630	T206SS	065076	T39EAI	076336		WC.IRE=	000010	X2.REV=	000077	
T19SEX	061572	T208SS	065141	T39ETN	076541		WC.IRW=	000004	X2.SPA=	035400	
T19SNP	061432	T23A	003146	T39ETS	076452		WC.IOT=	000100	X2.UNI=	000007	
T19SRD	061370	T23B	003150	T39MCL	077454		WC.IIT=	000040	X2.WCF=	002000	

M4

Symbol table

X3.DCK= 000010	X3.OPI= 000100	X3.SPA= 000200	X4.MBZ= 017400	X4.TSM= 020000
X3.MBZ= 000006	X3.REV= 000040	X3.TRF= 000020	X4.RCE= 040000	X4.WRC= 000377
X3.MDE= 177400	X3.RIB= 000001	X4.HSP= 100000		

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

Errors detected: 0

*** Assembler statistics

Work file reads: 398
Work file writes: 363
Size of work file: 32288 Words (127 Pages)
Size of core pool: 19684 Words (75 Pages)
Operating system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:10:06.20
CVTSBE,CVTSBE/-SP=SVC/ML,CVTSBE