

TSV05

TSV05 CTRL LT1
CVTSABO

COPYRIGHT (c) 1982-84
AH-T094E-MC
FICHE 01 OF 01

APR 1985
digital
Made In USA

The main body of the page consists of a 10x10 grid of small, dark, rectangular elements. Each element appears to be a small diagram or data table, but the text and graphics within them are too small and faded to be legible. The grid covers most of the page area below the header and above the footer.

001 100 100
002 100 100
003 100 100
004 100 100

.REM_
IDENTIFICATION

PRODUCT ID: AC-T093B-MC
PRODUCT TITLE: CVTSABO TSV05 CTRL LT1
DECO/DEPO: 1.0
DEPARTMENT: COMPUTER SPECIAL SYSTEMS/PPG
DATE: JUNE 4, 1984

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

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

COPYRIGHT (C) 1982, 1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

TABLE OF CONTENTS

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

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS IS A LSI-11 RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF A TSV05 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A LSI-11/23 SYSTEM (QBUS). THE PROGRAM PROVIDES ERROR MESSAGES WHICH IDENTIFY FAILING FUNCTIONS THAT AID IN THE REPAIR OF THE DEVICE. THIS DIAGNOSTIC CONSIST OF ELEVEN TEST WHICH ARE EXECUTED IN SEQUENCE.

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

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

1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

1. CHQUS XXDP+ USERS GUIDE; DOCUMENT NUMBER AC-F348E-MC
DATE: 14 JULY 1980.
2. TSV05 TRANSPORT SUBSYSTEM USER'S GUIDE; DOCUMENT NUMBER EK-TSV05-UG-001
DATE: AUGUST 1983
3. TSV05 TRANSPORT SUBSYSTEM TECHNICAL MANUAL; DOCUMENT NUMBER EK-TSV05-TM-001
DATE: AUGUST 1983
4. TSV05 TRANSPORT SUBSYSTEM INSTALLATION MANUAL; DOCUMENT NUMBER EK-TSV05-IN-001
DATE: AUGUST 1983

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

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

1.5 ASSUMPTIONS

ALL HARDWARE EXCEPT THE HARDWARE UNDER TEST IS ASSUMED TO WORK PROPERLY OR FALSE ERRORS CAN BE REPORTED. THE TAPE BEING USED ON THE TSO5 TRANSPORT IS A KNOWN GOOD REEL OF TAPE.

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

BOOT THE DIAGNOSTIC MEDIA

```
.R VTSA??
DIAG. RUN-TIME SERVICES REV D. APR 79
CVTSA-A-0
****TSV05 LOGIC DIAGNOSTIC****
UNIT IS TSV05
>DR
```

2.2 SWITCHES

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

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10. THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO BE RUN. ALL OTHER TESTS WILL NOT BE RUN.
/PASS:DDDD	EXECUTE DDDD 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 DDDD 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 LSI-11 AND UP TO 2 DRIVES PER CONTROLLER

2.5 SOFTWARE QUESTIONS

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

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

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

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

2.6 EXTENDED P-TABLE DIALOGUE

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

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

UNITS (0) ? 8<CR>

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

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

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

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

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

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

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

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

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

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

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

```
# UNITS (D) ? 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 (D) ? 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. TYPE "R NAME", WHERE NAME IS THE NAME OF THE BIN OR BIC FILE FOR THIS PROGRAM.
3. TYPE "START"
4. ANSWER THE "CHANGE HW" QUESTION WITH "Y"
5. ANSWER ALL THE HARDWARE QUESTIONS
6. 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 "IXE" 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
CVTSA WRD 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> I/RY<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 AN UNRECOVERABLE ERROR OCCURED WHICH INDICATES THAT THE CONTROLLER MAY BE DEFECTIVE.

CVTSA WRD 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.

CVTS WPD ERR 00121 ON UNIT 00 TST 001 SUB 002 P: 023306
MOT BIT (XST0) NOT SET DURING REWIND (EXTENDED FEATURES MODE)
EXPD: 060312 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 (LSI-11)

```
DR>STA/FLA:PNT:MOE
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 TWO SWITCHES ON WHICH ARE "PRINT EACH TEST NBR AS EXECUTED" AND "HALT ON ERROR".

```
TST: 001 INITIALIZE #1
TST: 002 WRAP DATA HIGH BYTE TEST
TST: 003 WRAP DATA LOW BYTE TEST
TST: 004 RAM TEST
TST: 005 INITIALIZE 2 TEST
TST: 006 COMMAND REJECT TEST
TST: 007 WRITE CHARACTERISTICS TEST
TST: 008 VOLUME CHECK
TST: 009 COMPLETION INTERRUPT TEST
TST: 010 BASIC PACKET PROTOCOL TEST
TST: 011 NON-TAPE-MOTION COMMANDS TEST
```

0 ERRORS

NOTE: THE DIAGNOSTIC WILL RUN CONTINUOUSLY UNLESS A PASS NUMBER 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 LSI-11 PROCESSOR WITH A LA34 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	1	30	29
2	1	10	9
3	1	8	7
4	25	120	95
5	5	140	135
6	25	475	450
7	20	20	0
8	1	10	9
9	20	20	0
10	1	2	1
11	8	11	3

THE TIMES REQUIRED TO RUN TESTS 1 THROUGH 12 IN ONE COMMAND:

Q.V. 1 MIN 57 SECONDS
DEFAULT 12 MINS

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) ?

INHIBIT ITERATIONS (L) N ?

6.0 TEST SUMMARIES

TEST 1: BUS RESET TEST

THIS TEST VERIFIES THAT THE M7196 MODULE'S DEVICE REGISTERS ARE ACCESSIBLE ON THE BUS (SUBTEST 1) AND THEN CHECKS THAT THE BUILT-IN INITIALIZATION SELF-TEST MICRODIAGNOSTIC DID NOT FIND ANY BASIC PROBLEMS IN THE MODULE. AREAS OF LOGIC TESTED BY THE SELF-TEST SEQUENCE ARE AS FOLLOWS: ROM AND PIPELINE REGISTER, SEQUENCER, INTERNAL BUSES, 2901 MICROPROCESSOR, AND, RAM. THIS TEST INITIALIZES THE CONTROLLER BY ISSUING THE BUS INIT SIGNAL VIA A RESET INSTRUCTION, OR BY WRITING INTO THE TSSR REGISTER, WAITS A PERIOD OF TIME (TO ALLOW THE CONTROLLER'S INITIALIZATION MICRODIAGNOSTIC SEQUENCE TO BE COMPLETED), AND THEN CHECKS THE CONTENTS OF THE TSSR REGISTER. SUCCESSFUL INITIALIZATION IS INDICATED BY SUBSYSTEM READY (SSR) AND NEED BUFFER ADDRESS (NBA) BITS BEING SET (1) AND ALL OTHER BITS (EXCEPT A17 AND A16 AND OFL, WHICH ARE IGNORED FOR THIS TEST) BEING CLEAR (0). IF THE CONTENTS OF TSSR ARE NOT AS EXPECTED, AN ERROR REPORT IS ISSUED LISTING THE EXPECTED DATA, ACTUAL DATA, AND THE DISCREPANCIES. THE ERROR REPORT ANALYZES THE TSSR CONTENTS AND DISCERNS AND REPORTS ONE OF THREE POSSIBILITIES:

1. TSSR CONTENTS ARE AMBIGUOUS (ANY OF BITS 11-14 ARE SET, OR STATES OF SSR AND SC BITS DO NOT CORRESPOND TO THE APPARENT ERROR CODE IN BITS 0-5): INDICATES THAT THE TSSR CONTENT CANNOT BE TRUSTED. INDICATES A CATASTROPHIC CONTROLLER MALFUNCTION. THIS IS A FATAL ERROR (EXECUTION IS ABORTED). FIELD ACTION WOULD BE TO REPLACE THE M7196. IF THE M7196 ITSELF IS BEING DEBUGGED, THE PROGRAM SHOULD BE RESTARTED WITH LOOP ON ERROR ENABLED IN ORDER TO PROBE FOR THE PROBLEM.
2. SSR = 0, SC = 0 AND THE ERROR CODE IN BITS 0-5 IS IN THE RANGE 17-13: THIS IS A FATAL ERROR. THE ERROR CODE IS DECODED AND THE APPROPRIATE DESCRIPTION GIVEN. INDICATES THAT A SERIOUS PROBLEM EXISTS.

TEST 2: WRAP DATA - HIGH BYTE

THIS TEST VERIFIES OPERATION OF:

1. PART OF THE LSI-11 BUS INTERFACE SECTION OF THE M7196 MODULE: PART OF THE INPUT FILE (TSDB HIGH BYTE), PART OF THE OUTPUT FILE (TSSR HIGH BYTE AND TSBA, BOTH BYTES), PART OF THE DCO05 TRANSCEIVER CIRCUITS (ADDRESS DECODER, BDAL DRIVERS, HIGH BYTE OF INTERNAL DAL BUS DRIVERS), AND BASIC PROGRAMMED I/O CONTROL SEQUENCES AND LOGIC;
2. PART OF 2901 MICROPROCESSOR ELEMENTS (Q-REGISTER, REGISTER 0, ROTATE AND NEGATE FUNCTIONS
3. Y AND SOURCE BUSES;
4. BASIC MICROPROGRAM SEQUENCES.

THE PROGRAM WRITES A TEST DATA BYTE INTO THE HIGH BYTE OF TSDB, WAITS FOR THE SSR BIT IN TSSR TO SET, THEN CHECKS THE CONTENTS OF BOTH TSBA AND TSSR. THE MODULE IS FUNCTIONING CORRECTLY IF DATA WRITTEN APPEARS IN BOTH BYTES OF TSBA AND THE FINAL CONTENT OF TSSR IS CORRECT (SAME AS AFTER INITIALIZATION EXCEPT FOR BITS 8 AND 9, WHICH SHOULD CONTAIN BITS 8 AND 9 OF THE DATA PATTERN WRITTEN. AN ERROR IS REPORTED AND A DESCRIPTIVE ANALYSIS GIVEN IF A DISCREPANCY IN TSBA OR TSSR IS DETECTED. THE ANALYSIS LISTS LIKELY FAULTY CANDIDATES FROM THE LOGIC ELEMENTS LISTED ABOVE. THE TEST IS REPEATED FOR ALL COMBINATIONS OF TEST DATA BYTES (0-377 OCTAL).

TEST 3: WRAP DATA - LOW BYTE

THIS TEST FURTHER VERIFIES OPERATION OF MANY OF THE SAME ELEMENTS TESTED IN TEST 2, AND ADDITIONALLY VERIFIES:

1. LOW BYTE OF THE TSDB INPUT FILE REGISTER,
2. LOW BYTE OF INTERNAL DAL BUS DRIVERS ON THE DCO05 TRANSCEIVER CIRCUITS,
3. BASIC FUNCTIONING OF PARTS OF THE RAM.

THE PROGRAM WRITES A TEST DATA BYTE INTO THE LOW BYTE OF TSDB, WAITS FOR THE SSR BIT IN TSSR TO SET, THEN CHECKS THE CONTENTS OF BOTH TSBA AND TSSR. THE MODULE IS FUNCTIONING CORRECTLY IF DATA WRITTEN APPEARS IN BOTH BYTES OF TSBA AND THE FINAL CONTENT OF TSSR IS CORRECT (SAME AS AFTER INITIALIZATION EXCEPT FOR BITS 8 AND 9, WHICH SHOULD CONTAIN BITS 8 AND 9 OF THE DATA PATTERN WRITTEN. AN ERROR IS REPORTED AND A DESCRIPTIVE ANALYSIS GIVEN IF A DISCREPANCY IN TSBA OR TSSR IS DETECTED. THE ANALYSIS LISTS LIKELY FAULTY CANDIDATES FROM THE LOGIC ELEMENTS LISTED ABOVE. THE TEST IS REPEATED FOR ALL COMBINATIONS OF TEST DATA BYTES (0-377 OCTAL).

TEST 4: RAM TEST

THIS TEST VERIFIES THAT ALL LOCATIONS OF THE RAM ON THE M7196 CAN PROPERLY STORE AND READ BACK ALL DATA PATTERNS, AND THAT EACH RAM LOCATION IS UNIQUELY ADDRESSED (I.E., THAT ONE AND ONLY ONE LOCATION IS ACCESSED BY ANY PARTICULAR ADDRESS). THE BYPRODUCT OF THESE TESTS IS A VERIFICATION OF TWO REGISTERS IN THE 2901 AND THE CAPABILITY OF THE 2901 TO CORRECTLY PERFORM AN ADD.

TEST 5: SECOND INITIALIZATION TEST

THIS TEST VERIFIES THE SAME ELEMENTS AS DID INITIALIZATION TEST #1 AND ALSO CHECKS THAT CERTAIN PARTS OF RAM IS CLEARED TO ZERO AND THAT 2901 REGISTERS 10 AND 11 ARE ALSO CLEARED TO ZERO. THIS IS A CONFIDENCE CHECK OF A PART OF THE SELF-TEST SEQUENCE (I.E., THAT IT IS REALLY BEING EXECUTED). FOR EACH OF TWO SUBTESTS (ONE FOR INITIALIZING VIA A BUS INIT, THE OTHER FOR INITIALIZING BY WRITING INTO THE TSSR), THE FOLLOWING SEQUENCE IS PERFORMED:

1. EACH RAM LOCATION AND 2901 REGISTERS 10 AND 11 ARE SET TO ALL 1'S BY USING WRITES INTO THE TSDB REGISTER (LOW BYTE AND MAINTENANCE MODE WORD WRITES).
2. THE CONTROLLER IS INITIALIZED AND THE VARIOUS CHECKS ON THE TSSR DESCRIBED IN INITIALIZATION TEST #1 ARE PERFORMED.
3. #1'S (377 OCTAL) ARE WRITTEN INTO THE LOW BYTE OF TSDB, WHICH SHOULD CAUSE RAM LOCATION 0 TO BE WRITTEN TO ALL 1'S SINCE 2901 REGISTERS 10 AND 11, SPECIFYING THE RAM ADDRESS, SHOULD BE 0. RAM LOCATION 0 IS VERIFIED BY WRITING A WORD OF ZEROS INTO THE TSDB. THE RESULTING LOW BYTE OF TSBA SHOULD CONTAIN ALL 1'S.
4. THE ENTIRE RAM IS SCANNED. LOCATION 0 SHOULD CONTAIN ALL 1'S AND THE REMAINING LOCATIONS, EXCEPT FOR THE MESSAGE BUFFER IMAGE AREA, SHOULD CONTAIN 0. DISCREPANCIES ARE REPORTED. AN ERROR AT THIS POINT IS MOST LIKELY DUE TO A ROM, PIPELINE OR SEQUENCER PROBLEM OR A TIMING PROBLEM.

TEST 6: COMMAND REJECT

THIS TEST VERIFIES THAT ALL COMMANDS OTHER THAN WRITE CHARACTERISTICS ARE REJECTED DUE TO THE NEED BUFFER ADDRESS (NBA) BIT BEING SET IN TSSR, AND THAT THE TSBA AND TSSR REGISTERS ARE LEFT IN THE PROPER STATE AFTER EACH COMMAND IS REJECTED. THIS TEST CHECKS MICROPROCESSOR SEQUENCING, BASIC COMMAND DECODING AND DATA DMA HANDLING. THIS TEST CONTAINS TWO SUBTESTS: SUBTEST 1 SEQUENCES THROUGH ALL COMMAND WORDS (OTHER THAN WRITE CHARACTERISTICS) WITH THE INTERRUPT ENABLE (IE) BIT CLEAR AND VERIFIES THAT AN INTERRUPT IS NOT GENERATED BY THE REJECTED COMMAND; SUBTEST 2 PERFORMS SIMILARLY TO SUBTEST 1 BUT SETS THE IE BIT IN EACH COMMAND WORD AND VERIFIES THAT AN INTERRUPT IS GENERATED WHEN THE COMMAND IS REJECTED.

TEST 7: WRITE CHARACTERISTICS

THIS TEST VERIFIES BASIC OPERATION OF THE WRITE CHARACTERISTICS COMMAND. IT VERIFIES THAT THE COMMAND BLOCK AND CHARACTERISTICS DATA BLOCK ARE FETCHED PROPERLY FROM CPU MEMORY, THE NEED BUFFER ADDRESS (NBA) BIT IN TSSR IS HANDLED PROPERLY, AND THAT A PROPER MESSAGE PACKET IS STORED, WHERE APPROPRIATE. THIS TEST DOES NOT CHECK THAT THE VARIOUS FUNCTIONS ENABLED BY CHARACTERISTIC MODE DATA BITS OPERATE PROPERLY; THE FUNCTIONING OF THESE BITS IS VERIFIED IN SUBSEQUENT TESTS. ALL COMMANDS EXECUTED IN THIS TEST HAVE THE INTERRUPT ENABLE (IE) BIT CLEARED TO ZERO, SO NO INTERRUPTS SHOULD BE GENERATED. HOWEVER, THE PROGRAM RUNS AT PROCESSOR PRIORITY 0, WITH THE INTERRUPT SERVICE ROUTINE SET UP TO FLAG UNEXPECTED INTERRUPTS. IF AN INTERRUPT OCCURS, A PROBLEM EXISTS IN EITHER THE LSI-11 BUS INTERFACE SECTION OR IN THE ROM OR PIPELINE.

TEST 8: VOLUME CHECK

THIS TEST VERIFIES THAT THE VOLUME CHECK (VCK) BIT, A FLAG HELD WITHIN THE M7196 AND APPEARING IN XST0, IS SET BY INITIALIZE AND CLEARED BY EXECUTING A WRITE CHARACTERISTICS COMMAND WITH THE CVC BIT SET. IT IS ALSO VERIFIED THAT A WRITE CHARACTERISTICS COMMAND WITH THE CVC BIT CLEAR DOES NOT AFFECT THE STATE OF THE VOLUME CHECK BIT. THE ACTUAL FUNCTION OF VOLUME CHECK, THAT OF PREVENTING OR ALLOWING A TAPE MOTION COMMAND DEPENDING UPON WHETHER VOLUME CHECK IS SET OR CLEAR, IS NOT CHECKED BY THIS TEST; THIS FUNCTIONALITY IS CHECKED IN THE INDIVIDUAL TESTS OF TAPE MOTION COMMANDS.

TEST 9: COMPLETION INTERRUPT

THIS TEST VERIFIES THAT AN INTERRUPT IS GENERATED AT THE COMPLETION OF THE WRITE CHARACTERISTICS COMMAND IF THE INTERRUPT ENABLE (IE) BIT IN THE COMMAND HEADER WORD IS SET. THIS TEST CHECKS THE FUNCTIONING OF THE INTERRUPT LOGIC AND BASIC PROCESSING OF THE IE BIT.

THE SEQUENCES OF TEST 7 ARE REPEATED, EXCEPT THAT THE INTERRUPT SERVICE ROUTINE IS SET UP TO EXPECT INTERRUPTS AND EACH WRITE CHARACTERISTICS COMMAND IS ISSUED WITH THE IE BIT SET (1). IT IS VERIFIED, WHERE APPROPRIATE, THAT THE IE STATUS BIT IN XSTO OF ANY MESSAGE PACKET IS SET AND THAT A COMPLETION INTERRUPT IS GENERATED. FINALLY, A SEQUENCE OF TWO COMMANDS ARE ISSUED, THE FIRST WITH IE=1 AND THE SECOND WITH IE=0. IT IS VERIFIED THAT NO INTERRUPT IS GENERATED AFTER THE SECOND COMMAND AND THAT THE IE BIT IN XSTO IS 0.

TEST 10: BASIC PACKET PROTOCOL

THIS TEST VERIFIES BASIC OPERATION OF THE MESSAGE BUFFER RELEASE COMMAND, THE FUNCTION OF THE ACK BIT IN THE COMMAND HEADER WORD, AND THE REGISTER MODIFICATION REFUSED (RMR) LOGIC.

TEST 11: NON-TAPE MOTION COMMANDS

THIS TEST VERIFIES PROPER OPERATION OF THE INITIALIZE COMMAND. TWO SUBTESTS ARE USED. THE FIRST VERIFIES THAT THE COMMAND RUNS TO COMPLETION AND STORES A VALID MESSAGE PACKET. THE SECOND VERIFIES THAT NON-ZERO VALUES IN THE COMMAND MODE FIELD CAUSES COMMAND REJECT.

7.0 MAINTENANCE HISTORY

REVISION A - MARCH 1982

REVISION B - JUNE 1984

MINOR CHANGES FOR THE ORION CPU (11/??).
ELIMINATED THE MESSAGE DESCRIBING THE CPU TYPE.

```

2          .TITLE  TSV2 - PROGRAM HEADER
3          .SBTTL  PROGRAM HEADER
4 000000   .PSECT  ABS
5
11         .MCALL  SVC
12 000000   SVC          ; INITIALIZE SUPERVISOR MACROS
13         .ENABLE LC
14         .NLIST  BEX,CND
20 000000   .ENABL  ABS,AMA
21         .=2000
22 002000   BGNMOD  TSV2
23         002000

```

TSV2::

```

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

```

```

30 002000   POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
31 002000   HEADER  CVTSA,8,0,655.,0
002000     L$NAME:: ;DIAGNOSTIC NAME
002000     .ASCII  /C/
002001     .ASCII  /V/
002002     .ASCII  /T/
002003     .ASCII  /S/
002004     .ASCII  /A/
002005     .BYTE   0
002006     .BYTE   0
002007     .BYTE   0
002010     L$REV:: ;REVISION LEVEL
002010     .ASCII  /B/
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$CO::
002032     .WORD   0
002034     L$DTYP:: ;DIAGNOSTIC TYPE
002034     .WORD   0
002036     L$APT:: ;APT EXPANSION
002036     .WORD   0
002040     L$DTP:: ;PTR. TO DISPATCH TABLE

```

002040	002124		.WORD	L\$DISPATCH	
002042		L\$PRIO::			;DIAGNOSTIC RUN PRIORITY
002042	000000		.WORD	0	
002044		L\$ENVI::			;FLAGS DESCRIBE HOW IT WAS SETUP
002044	000000		.WORD	0	
002046		L\$EXP1::			;EXPANSION WORD
002046	000C00		.WORD	0	
002050		L\$MREV::			;SVC REV AND EDIT #
002050	003		.BYTE	C\$REVISION	
002051	003		.BYTE	C\$EDIT	
002052		L\$EF::			;DIAG. EVENT FLAGS
002052	000000		.WORD	0	
002054	000000		.WORD	0	
002056		L\$SPC::			
002056	000000		.WORD	0	
002060		L\$DEVP::			; POINTER TO DEVICE TYPE LIST
002060	003400		.WORD	L\$DVTYP	
002062		L\$REPP::			;PTR. TO REPORT CODE
002062	022620		.WORD	L\$RPT	
002064		L\$EXP4::			
002064	000000		.WORD	0	
002066		L\$EXPS5::			
002066	000000		.WORD	0	
002070		L\$AUT::			;PTR. TO ADD UNIT CODE
002070	022306		.WORD	L\$AU	
002072		L\$DUT::			;PTR. TO DROP UNIT CODE
002072	022404		.WORD	L\$DU	
002074		L\$LUN::			;LUN FOR EXERCISERS TO FILL
002074	000000		.WORD	0	
002076		L\$DESP::			;POINTER TO DIAG. DESCRIPTION
002076	003406		.WORD	L\$DESC	
002100		L\$LOAD::			;GENERATE SPECIAL AUTOLOAD EMT
002100	104035		EMT	E\$LOAD	
002102		L\$ETP::			;POINTER TO ERR_TBL
002102	000000		.WORD	0	
002104		L\$ICP::			;PTR. TO UNIT CODE
002104	021512		.WORD	L\$INIT	
002106		L\$CCP::			;PTR. TO CLEAN-UP CODE
002106	022572		.WORD	L\$CLEAN	
002110		L\$ACP::			;PTR. TO AUTO CODE
002110	022512		.WORD	L\$AUTO	
002112		L\$PRT::			;PTR. TO PROTECT TABLE
002112	021502		.WORD	L\$PROT	
002114		L\$TEST::			;TEST NUMBER
002114	000000		.WORD	0	
002116		L\$DLY::			;DELAY COUNT
002116	000000		.WORD	0	
002120		L\$HIME::			;PTR. TO HIGH MEM
002120	000000		.WORD	0	


```

35
36           .SBTTL DISPATCH TABLE
37
38           ;**
39           ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
40           ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
41           ;--
42
43 002122           DISPATCH 11
002122 000013           .WORD 11
002124           L#DISPATCH::
002124 023402           .WORD T1
002126 023622           .WORD T2
002130 024320           .WORD T3
002132 025012           .WORD T4
002134 026346           .WORD T5
002136 027452           .WORD T6
002140 030724           .WORD T7
002142 034312           .WORD T8
002144 035216           .WORD T9
002146 040342           .WORD T10
002150 043454           .WORD T11
44
45
46           .SBTTL DEFAULT HARDWARE P-TABLE
47
48           ;**
49           ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
50           ; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
51           ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
52           ;--
53
54 002152           BGNHW DFPTBL ;DEFAULT HARD-P-TABLE
002152 000003           .WORD L10000-L#HW/2
002154           L#HW::
002154           DFPTBL::
55
56 002154 172520           .WORD 172520 ; 1ST (OF 2) REGISTERS.
57 002156 000224           .WORD 224 ; INTERRUPT VECTOR
58 002160 000200           .WORD PRI04 ; INTERRUPT PRIORITY.
59 002162           ENDPHW
002162           L10000:
60

```

```

62          .SBTTL  SOFTWARE P-TABLE
63
64          ;**
65          ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
66          ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
67          ;--
68 002162          BGNSW  SFPTBL
002162          .WORD  L10001-L$SW/2
002164          L$SW::
002164          SFPTBL::
69
70 002164          TRANSTST::  .WORD  0          ; ENABLE TEST OF TRANSPORT(S) IF =1
71 002166          NOITS::    .WORD  0          ; INHIBIT ITERATION OPTION.
72                                     ; ... 0 = ITERATE.
73                                     ; ...NZ = INHIBIT ITERATE.
74 002170          LERRMAX::  .WORD  15.         ; LOCAL (PER TEST) ERROR LIMIT
75 002172          GERRMAX::  .WORD  200.        ; GLOBAL (PER UNIT) ERROR LIMIT
76 002174          ENDSW
002174          L10001:
77
78 002174          ENDMOD
79
80
83
84

```

7
8
13
19
20 002174
002174
21
22
23
24
25
26
27
28
32 002174

.TITLE TSV3 - GLOBAL AREAS
.SBTTL GLOBAL EQUATES SECTION

BGNMOD TSV3

TSV3::

.SBTTL GLOBAL EQUATES SECTION

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

EQUALS ; GET STANDARD EQUATES.

; BIT DIFINITIONS

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

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

; 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

```

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

```

```

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

```

33
34 002174

```

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

```

;DEFINE MEMORY MANAGEMENT REGISTERS

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

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

```

39          .SBTTL  TSV05 REGISTER AND PACKET DEFINITIONS
40
41          ;
42          ; SOME GENERAL EQUATES.
43          ;
44
45          000004      ERRVEC==      4          ; POINTER TO ERROR VECTOR FOR BUS TIME OUT.
46          000060      TTIVEC==     60          ; INTERRUPT VECTOR FOR CONSOLE INPUT
47          177560      TTICSR==     177560     ; BUS ADDRESS OF CONSOLE INPUT
48          177562      TTIBFR==     177562     ; CONSOLE INPUT DATA BUFFER
49          177520      BDVPCR==     177520     ; BDV11 PAGE CONTROL REGISTER
50
51          ;*
52          ;BIT DEFINITIONS FOR TSSR REGISTER
53          ;-
54
55          100000      SC=      BIT15      ;SPECIAL CONDITION
56          040000      BIE=     BIT14      ;BUS INTERFACE ERROR
57          020000      SCE=     BIT13      ;SANITY CHECK ERROR
58          010000      RMR=     BIT12      ;MODIFICATION REFUSED
59          004000      NXM=     BIT11      ;NONEXISTANT MEMORY ERROR
60          002000      NBA=     BIT10      ;NEED BUFFER ADDRESS
61          001400      HIADDR= BIT9!BIT8   ;EXTENDED ADDRESS BITS
62          000200      SSR=     BIT7       ;SUB SYSTEM READY
63          000100      OFL=     BIT6       ;OFF LINE BIT
64          000060      FATERR= BIT4!BIT5   ;FATAL TERMINATION ERROR CODES
65          000016      TERCLS= BIT3!BIT2!BIT1 ;TERMINATION CODES
66
67          ;*
68          ;
69          ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
70          ;(XSTO)
71          ;
72          ;-
73
74          100000      XSOTMK= BIT15      ;TAPE MARK DETECTED
75          040000      XSORLS= BIT14      ;RECORD LENGTH SHORT
76          020000      XSOLET= BIT13      ;LOGICAL END OF TAPE
77          010000      XSORLL= BIT12      ;RECORD LENGTH LONG
78          004000      XSOMLE= BIT11      ;WRITE LOCK ERROR
79          002000      XSONEF= BIT10      ;NON EXECUTABLE FUNCTION
80          001000      XS0ILC= BIT9       ;ILLEGAL COMMAND
81          000400      XS0ILA= BIT8       ;ILLEGAL ADDRESS
82          000200      XS0MOT= BIT7       ;TAPE IN MOTION
83          000100      XS0ONL= BIT6       ;TRANSPORT ON LINE
84          000040      XS0IE=  BIT5       ;INTERRUPT ENABLE
85          000020      XS0VCK= BIT4       ;VOLUME CHECK BIT
86          000010      XS0PED= BIT3       ;PHASE ENCODED DRIVE
87          000004      XS0MLK= BIT2       ;WRITE LOCKED
88          000002      XS0BOT= BIT1       ;BEGINNING OF TAPE
89          000001      XS0EOT= BIT0       ;END OF TAPE

```

```

91      ;*
92      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
93      ;(XST1)
94      ;-
95      100000 X1.DLT = BIT15      ;DATA LATE
96      040000 X1.SPARE= BIT14      ;NOT USED
97      020000 X1.COR = BIT13      ;CORRECTABLE DATA ERROR
98      017375 X1.MBZ = BIT12*BIT11*BIT10*BIT9*BIT7*BIT6*BIT5*BIT4*BIT3*BIT2*BIT0 ;ALWAYS 0
99      000400 X1.RBP = BIT8      ;READ BUS PARITY ERROR
100     000002 X1.UNC = BIT1      ;UNCORRECTABLE DATA OR HARD ERROR
101
102     ;*
103     ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
104     ;(XST2)
105     ;-
106     100000 X2.OPM = BIT15      ;OPERATION IN PROGRESS (TAPE MOVING)
107     040000 X2.RCE = BIT14      ;RAM CHECKSUM ERROR
108     035400 X2.SPARE= BIT13*BIT12*BIT11*BIT9*BIT8 ;NOT USED BY TSV05 (ALWAYS=0)
109     002000 X2.WCF = BIT10      ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
110     000200 X2.EXTF = BIT7      ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
111     000100 X2.BUFE = BIT6      ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
112     000077 X2.REV = 000077    ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
113     000007 X2.UNIT = BIT2*BIT1*BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.
114
115     ;*
116     ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
117     ;(XST3)
118     ;-
119     177400 X3.MDE = 177400    ;MICRO-DIAGNOSTIC ERROR CODE
120     000200 X3.SPARE= BIT7      ;NOT USED BY TSV05
121     000100 X3.OPI = BIT6      ;OPERATION INCOMPLETE
122     000040 X3.REV = BIT5      ;REVERSE
123     000020 X3.TRF = BIT4      ;TRANSPORT RESPONSE FAILURE
124     000010 X3.DCK = BIT3      ;DENSITY CHECK
125     000006 X3.MBZ =BIT2*BIT1  ;NOT USED ALWAYS 0
126     000001 X3.RIB = BIT0      ;REVERSE INTO BOT
127
128     ;*
129     ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
130     ;(XST4)
131     ;-
132     100000 X4.HSP = BIT15      ;HIGH SPEED
133     040000 X4.RCE = BIT14      ;RETRY COUNT EXCEEDED
134     020000 X4.TSM = BIT13      ;TRANSPORT SPECIAL MODE
135     017400 X4.MBZ = BIT12*BIT11*BIT10*BIT9*BIT8 ;NOT USED ALWAYS 0
136     000377 X4.WRC = 000377    ;WRITE RETRY COUNT FIELD
137
138     ;*
139     ;
140     ;TSSR TERMINATION CODES (BIT 0-2)
141     ;
142     ;-
143
144     000006 TSREJ= 3*2          ;COMMAND REJECTED
145     000006 UNREC= 6           ;UNRECOVERABLE ERROR
    
```



```

147      ;+
148      ;
149      ;DEVICE REGISTER OFFSETS
150      ;
151      ;-
152
153      000000      TSBA== 0
154      000000      TSDB== 0      ;TSDB/TSBA REGISTER
155      000001      TSBAH== 1
156      000001      TSDBH== 1      ;TSDB/TSBA REGISTER HIGH BYTE
157      000002      TSSR== 2      ;TSSR REGISTER
158      000003      TSSRH== 3      ;TSSR REGISTER HIGH BYTE
159
160      ;+
161      ; TSDB ADDRESS BIT DEFINITIONS
162      ;-
163      000003      A1716 = BIT1+BIT0      ;ADDRESS BITS 17:16 ARE IN 1:0
164
165      ;+
166      ; COMMAND DEFINITIONS
167      ;-
168      000017      P.GETSTAT      = 17      ;GET STATUS
169      000013      P.INIT        = 13      ;INITIALIZE
170      000012      P.CONTROL     = 12      ;CONTROL COMMANDS
171      000011      P.FORMAT      = 11      ;FORMAT
172      000010      P.POSITION    = 10      ;POSITION
173      000006      P.WRTSUB      = 6       ;SUBSYSTEM WRITE
174      000005      P.WRITE       = 5       ;WRITE
175      000004      P.WRTCHAR     = 4       ;WRITE CHARACTERISTICS
176      000001      P.READ        = 1       ;READ
177
178      ;+
179      ; COMMAND PACKET HEADER WORD BIT DEFINITIONS
180      ;-
181      100000      P.ACK        = BIT15      ;BUFFER AVAIL FOR CONTROLLER
182      040000      P.CVC        = BIT14      ;CLEAR VOLUME CHECK
183      020000      P.OPP        = BIT13      ;REVERSE SEQUENCE OF DATA BITS
184      010000      P.SWB        = BIT12      ;SWAP BYTES IN MEMORY
185      007400      P.MODE       = BIT11!BIT10!BIT9!BIT8 ;EXTENDED COMMAND MODE FIELD
186      000200      P.IE         = BIT7       ;INTERRUPT ENABLE
187      000140      P.FMT= BIT6!BIT5      ;PACKET HEADER TYPE (ALWAYS=0)
188      000037      P.CMD        = 37       ;MAJOR COMMAND FIELD
189
190      ;+
191      ; CONTROL COMMAND MODE CODES
192      ;-
192      000000      PC.RELEASE    = 0*256.   ;RELEASE BUFFER
193      000400      PC.REWIND     = 1*256.   ;REWIND
194      001000      PC.NOOP       = 2*256.   ;NO-OP
195      002000      PC.IEREW      = 4*256.   ;REWIND IMMEDIATE INTERRUPT
196      002400      PC.ERASE      = 5*256.   ;SECURITY ERASE

```

```

198      ;+
199      ; CONTROLLER RAM DEFINITIONS
200      ;-
201      000167      RMCHBEG = 167      ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
202      000200      RMCHEND = 200      ;CHARACTERISTICS IO DATA END RAM ADDRESS
203      000201      RMPKTBEG= 201      ;COMMAND PACKET BEGIN RAM ADDRESS
204      000210      RMPKTEND= 210      ;COMMAND PACKET END RAM ADDRESS
205      000215      RMMSGBEG= 215      ;MESSAGE BUFFER BEGIN RAM ADDRESS
206      000234      RMMSGEND= 234      ;MESSAGE BUFFER END RAM ADDRESS
207      ;+
208      ;
209      ;REGISTER DEFINITIONS IN THE MESSAGE BUFFER
210      ;
211      ;-
212
213      000006      XST0== 6      ;EXTENDED STATUS REGISTER 0 (WORD 4)
214      000010      XST1== 8.      ;EXTENDED STATUS REGISTER 1 (WORD 5)
215      000012      XST2== 10.      ;EXTENDED STATUS REGISTER 2 (WORD 6)
216      000014      XST3== 12.      ;EXTENDED STATUS REGISTER 3 (WORD 7)
217      000016      XST4== 14.      ;EXTENDED STATUS REGISTER 4 (WORD 8)
218
219      ;+
220      ;
221      ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
222      ;
223      ;-
224
225      000002      PKLOW  = 2      ;LOW ORDER CHARACTERISTIC DATA POINTER
226      000004      PKHI   = 4      ;HIGH ORDER CHARACTERISTIC DATA POINTER
227      000006      PKBCNT = 6      ;NUMBER OF BYTES IN DATA PACKET
228
229      000010      EXBCNT=10      ;NUMBER OF BYTES IN EXTENDED DATA PACKET
230
231      ;+
232      ;DATA PACKET OFFSETS FOR WRITE SUBSYSTEM COMMAND
233      ;-
234      000000      BSEL0  = 0      ;BYTE 0
235      000001      BSEL1  = 1      ;BYTE 1
236      000002      SEL2   = 2      ;WORD 2
237      000004      SELDATA = 4      ;WORD 3

```

```

239      ;*
240      ;BSELO SELECT CODES FOR WRITE SUBSYSTEM COMMAND
241      ;-
242      000000      PW.NOP          = 0          ;NO-OP
243      000001      PW.RDRAM       = 1          ;READ RAM
244      000002      PW.WTRAM       = 2          ;WRITE RAM
245      000003      PW.RFIFO       = 3          ;READ FIFO
246      000004      PW.WFIFO       = 4          ;WRITE FIFO
247      000005      PW.RDSTAT      = 5          ;READ STATUS
248      000006      PW.WCTL        = 6          ;WRITE TAPE CONTROL
249      000007      PW.WFMT        = 7          ;WRITE TAPE FORMAT
250      000010      PW.WMISC       = 10         ;WRITE MISCELLANEOUS
251      000011      PW.WNPR        = 11         ;WRITE NPR CONTROL
252      000020      PW.D22         = 20         ;DO MICROTEST 22
253      000021      PW.D11         = 21         ;DO MICROTEST 11
254      000022      PW.D13         = 22         ;DO MICROTEST 13
255      000023      PW.NO1311     = 23         ;DISABLE MICROTEST 11 AND 13
256      000024      PW.RDXT        = 24         ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS)
257
258      ;*
259      ;BSEL1 CODES FOR WRITE TAPE CONTROL
260      ;-
261      000200      WC.IFAD         = BIT7       ;IFAD - FORMATTER ADDRESS
262      000100      WC.IOTAD        = BIT6       ;ITADO - TRANSPORT ADDRESS BIT 0
263      000040      WC.I1TAD        = BITS5     ;ITAD1 - TRANSPORT ADDRESS BIT 1
264      000020      WC.ISRESV       = BIT4       ;IRESV5 - RESERVED #5
265      000010      WC.IREW         = BIT3       ;IREW - REWIND
266      000004      WC.IRWU         = BIT2       ;IRWU - REWIND AND UNLOAD
267      000002      WC.IFEN         = BIT1       ;IFEN - FORMATTER ENABLE
268      000001      WC.IGO          = BIT0       ;GO
269
270      ;*
271      ;BSEL1 CODES FOR WRITE FORMAT
272      ;-
273      000200      WF.IHISP         = BIT7       ;IHISP - HIGH SPEED
274      000100      WF.IWRT         = BIT6       ;IWRT - WRITE
275      000040      WF.IREV         = BITS5     ;IREV - REVERSE
276      000020      WF.IWFM         = BIT4       ;IWFM - WRITE FILE MARK
277      000010      WF.IEDIT        = BIT3       ;IEDIT - EDIT
278      000004      WF.IERASE       = BIT2       ;IERASE - ERASE
279      000002      WF.I3RESV       = BIT1       ;IRESV3 - RESERVED #3
280      000001      WF.I4RESV       = BIT0       ;IRESV4 - RESERVED #4
281
282      ;*
283      ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
284      ;-
285      000200      MS.EXT           = BIT7       ;INVERT SENSE OF EXTENDED FEATURES SWITCH
286      000020      MS.RSFIFO       = BIT4       ;RESET FIFO AND INPUT PARITY ERRORR
287      000010      MS.RSTAPE       = BIT3       ;RESET TAPE STATUS IN 2 FLIP-FLOPS
288      000006      MS.ATTN         = BIT2:BIT1  ;ATTENTION TRIGGER FIELD
289      000001      MS.RSD           = BIT0       ;RESET TIMER A,B THEN DELAY TIMES IN SEL2

```

```

291      ;+
292      ; MS.ATTN SUBCODES
293      ;-
294      000000      MSA.NOP = 0*2      ;NO-OP (NOTHING TRIGGERED)
295      000002      MSA.VOL = 1*2      ;SIMULATE ON-LINE/OFF-LINE TRANSITION
296      000004      MSA.NRAM= 2*2      ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
297      000006      MSA.FRAME= 3*2     ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
298      ;+
299      ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
300      ;-
301      000200      NP.IR      = BIT7      ;INTERUPT REQUEST (0-1 TRANSITION)
302      000100      NP.OUT     = BIT6      ;TAPE DATA DIRECTION OUT (0= IN)
303      000040      NP.LOOP   = BITS      ;ENABLE TRANSPORT LOOPBACK
304      000020      NP.WRP    = BIT4      ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
305      ;+
306      ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
307      ;-
308
309      000200      S2.DIM      = BIT7      ;WORD #9 BYTE 2 DATA IN MISS
310      000100      S2.ILW     = BIT6      ; ILW H
311      000040      S2.OUTRDY  = BITS      ; OUT RDY H
312      000020      S2.INRDY   = BIT4      ; IN RDY H
313      000010      S2.ATIMR   = BIT3      ; TIMER A FLAG H
314      000004      S2.BTIMR   = BIT2      ; TIMER B FLAG H
315      000003      S2.UNDEF    = BIT1+BIT0 ;(UNDEFINED)
316      100000      S1.PARIN    = BIT15     ;WORD #8 BYTE 1 PARIN H
317      040000      S1.I2RESV   = BIT14     ; IRESV2
318      020000      S1.I1RESV   = BIT13     ; IRESV1
319      010000      S1.IEOT     = BIT12     ; IEOT L
320      004000      S1.IIDENT   = BIT11     ; IIDENT H
321      002000      S1.ICER     = BIT10     ; ICER H
322      001000      S1.IFMK    = BIT9      ; IFMK H
323      000400      S1.IHER     = BIT8      ; IHER H
324      000200      S0.ISPEED   = BIT7      ;WORD #8 BYTE 0 ISPEED H
325      000100      S0.IRDY    = BIT6      ; IRDY L
326      000040      S0.IONL    = BITS      ; IONL L
327      000020      S0.ILDP    = BIT4      ; ILDP L
328      000010      S0.IDBY    = BIT3      ; IDBY L
329      000004      S0.IRWD    = BIT2      ; IRWD L
330      000002      S0.IFBY    = BIT1      ; IFBY L
331      000001      S0.IFPT    = BIT0      ; IFPT L

```

```

333                   .SBTTL SPECIAL MACROS AND OPDEFS.
334
335           ;+
336           ;SAVE GENERAL REGS 1 TO 5
337           ;-
338
339                   .MACRO SAVREG
340                   JSR    R5,REGSAV
341                   .ENDM
342
343           ;+
344           ; MACRO TO FORCE AN ERROR
345           ;-
346           .MACRO FORCERROR       TAG,NOTSSR
347           .NLIST
348           .IIF NDF LISTALL, .NLIST
349           .LIST
350           .IF B NOTSSR
351                   MOV    TSSR(R5),R1       ;READ TSSR
352           .ENDC
353                   MOV    FORCER,FORCER     ;IS FORCER SET? (LEAVE C BIT ALONE)
354                   BNE    TAG               ;BR IF YES
355           .NLIST
356           .IIF NDF LISTALL, .LIST
357           .LIST
358           .ENDM
359
360           ;+
361           ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
362           ;       WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
363           ;       SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
364           ;       FORCER TO 177777
365           ;       TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
366           ;-
367           .MACRO FORCEEXIT        TAG
368           .NLIST
369           .IIF NDF LISTALL, .NLIST
370           .LIST
371                   MOV    FORCER,FORCER     ;IS FORCER NEGATIVE?
372                   BMI    TAG               ;BR IF YES
373           .NLIST
374           .IIF NDF LISTALL, .LIST
375           .LIST
376           .ENDM
377           ;+
378           ; MACRO TO INCREMENT ERROR COUNTS
379           ;-
380           .MACRO NEXT.ERRNO
381           .NLIST
382           ;:::IIF NDF LISTALL, .NLIST
383                   ERRNO=ERRNO+1
384           ;:::IIF NDF LISTALL, .LIST
385           .LIST
386           .ENDM

```


.SBTTL GLOBAL DATA SECTION

```

411
412
413      ;**
414      ;THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
415      ;IN MORE THAN ONE TEST.
416      ;--
417
418      ;
419      ;THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
420      ;SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.
421      ;
422 002176 000000 EPRTSW::      .WORD 0      ;PRINT SWITCH
423 002200 000000 UNITN::      .WORD 0      ;UNIT # UNDER TEST.
424 002202 000000 QVP::        .WORD 0      ;QUICK VERIFY FLAG.
425 002204 000000 CSRADDR::    .WORD 0      ;ADDRESS OF CSR FOR CURRENT DEVICE
426 002206 000224 IVEC::        .WORD 224    ;INTERRUPT VECTOR
427 002210 000200 IPRI::        .WORD PRI04  ;INTERRUPT PRIORITY.
428 002212 000000 TSTCNT::    .WORD 0      ;NUMBER OF TESTS RUN IN THIS PASS
429 002214 000000 LOOPCNT::   .WORD 0      ;REMAINING ITERATION COUNT FOR TEST
430 002216 000000 DEVCNT::    .WORD 0      ;NUMBER OF DEVICE UNDER TEST
431 002220 000000 FATFLG::    .WORD 0      ;SET IF FATAL ERROR IS DETECTED IN TEST
432 002222 000000 INTRECV::   .WORD C      ;SET IF TAPE INTERRUPT WAS RECEIVED
433 002224 000000 EXTFEA::    .WORD 0      ;EXTENDED FEATURES SOFTWARE SW 0-OFF;1-ON
434 002226 000000 BENBSW::    .WORD 0      ;BUFFER ENABLE SWITCH SW 0-OFF;1-ON
435 002230 000000 EXPD::      .WORD 0      ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
436 002232 000000 RECV::      .WORD 0      ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
437 002234 000000 ERRHI::     .WORD 0      ;HIGH ADDRESS MEMORY ERROR
438 002236 000000 ERRLO::     .WORD 0      ;LOW ADDRESS MEMORY ERROR
439 002240 000000 RANDATA::   .BLKW 16.    ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
440 002300 000000 RAMSIZ::   .WORD 0      ;RAM DATA SIZE FOR PRAMPKT ROUTINE
441 002302 000000 RCVHIADD:: .WORD 0      ;RECEIVED BUFFER HIGH ADDRESS
442 002304 000000 RCVLOADD:: .WORD 0      ;RECEIVED BUFFER LOW ADDRESS
443 002306 000000 COUNT::    .WORD 0      ;TEST COUNT PATTERN
444 002310 000000 DATA::    .WORD 0      ;TEST DATA
445 002312 000000 TSTFLAG::  .WORD 0      ;TEST FLAG WORD
446 002314 000000 TSTPTR::   .WORD 0      ;TSTBLK POINTER
447 002316 000000 PRMNO::    .WORD 0      ;PRINT ROUTINE TEMP
448 002320 000000 EXPMSG::   .BLKB 100.   ;EXPECTED MESSAGE BUFFER DATA
449 002464 000000 RECMSG::   .BLKB 100.   ;RECEIVED MESSAGE BUFFER DATA
450 002630 000000 TMPBFR::   .BLKB 80.    ;TEMPORARY STORAGE FOR PRINT

```

452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468 002750
469 002750 000000
470 002752 177777
471 002754 000001
472 002756 000002
473 002760 000004
474 002762 000010
475 002764 000020
476 002766 000040
477 002770 000100
478 002772 000200
479 002774 000400
480 002776 001000
481 003000 002000
482 003002 004000
483 003004 010000
484 003006 020000
485 003010 040000
486 003012 100000
487 003014 177776
488 003016 177775
489 003020 177773
490 003022 177767
491 003024 177757
492 003026 177737
493 003030 177677
494 003032 177577
495 003034 177377
496 003036 176777
497 003040 175777
498 003042 173777
499 003044 167777
500 003046 157777
501 003050 137777
502 003052 077777
503 003054 125252
504 003056 052525
505 003060

.SBTTL TSTBLK - TEST DATA TABLE

```

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

```

```

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==.

```



```

507          .SBTTL GLOBAL ENVIRONMENT STORAGE
508          ;
509          ; STORAGE FOR DEVICE REGISTERS
510          ;
511 003060 000000 100000 000000 DUMMY: 0,100000,0,0 ; DUMMY DEVICE REGISTERS...
512 003070 000000 000000 000000      0,0,0,0,0,0,0,0,0
513          ; ...FOR MULTI-UNIT CHECKOUT.
514          ;
515 003110 000000 DUFFLG:: .WORD 0 ; "DROPPED UNIT" FLAG.
516          ; INHIBITS CODE IN "CLEAN-UP".
517 003112 000000 NODEV:: .WORD 0 ; FLAG TO SAY NO DEVICE.
518          ;
519 003114 000000 TEMP1:: .WORD 0 ; SOME TEMP LOCATIONS.
520 003116 000000 TEMP2:: .WORD 0
521 003120 000000 XXCOMM:: .WORD 0 ; XXDP* COMM BLOCK POINTER.
522 003122 000000 FREE:: .WORD 0 ; 1ST FREE MEMORY ADDRESS...
523 003124 000000 FRESIZ:: .WORD 0 ; ...AND SIZE (IN WORDS).
524 003126 000000 FREEHI: .WORD 0 ; LAST WORD IN FREE SPACE
525 003130 000000 KTF LG:: .WORD 0 ; KT11, MEM AVAIL FLAG -
526          ; - .WORD 0 = <24K OR NO KT -
527          ; - NZ = >24K AND KT.
528 003132 000000 KTENABLE:: .WORD 0 ; SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
529 003134 000000 NXMFLG:: .WORD 0 ; SET IF WE CAN TEST CLEARED OTHERWISE
530 003136 000000 NXMLO:: .WORD 0 ; NXM LO ADDRESS BITS
531 003140 000000 NXMHI:: .WORD 0 ; NXM HI ADDRESS BITS FOR DAL'S 16-21
532 003142 000000 T23A:: .WORD 0 ; 11/23A FLAG
533 003144 000000 T23B:: .WORD 0 ; 11/23B FLAG
534 003146 000000 T3BFLG:: .WORD 0 ; TEST 3B FLAG +0
535 003150 002000 PST32W:: .WORD 2000 ; 32W BLOCK ADDRESS FOR 32K START
536 003152 000000 SIFLAG:: .WORD 0 ;
537 003154 000000 BADDAT:: .WORD 0 ; ACTUAL DATA
538 003156 000000 GDDAT:: .WORD 0 ; EXPECTED DATA
539 003160 000000 LOOPFL:: .WORD 0 ;
540 003162 CTAB:: .WORD 0 ; CONFIGURATION TABLES.
541 003162 000000 CTABM:: .WORD 0 ; CONFIG WORK.
542 003164 000000 .WORD 0
543 003166 000000 .WORD 0
544 003170 000000 .WORD 0
545 003172 177777 .WORD -1 ; END OF MEM TABLE.
546 003174
547          CTASE::
548          ; ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
549          ;
550          ; 0 = UNIT NOT TESTED
551          ; 100000 = UNIT ONLINE, NO ERRORS
552          ; 10XXXX = UNIT ONLINE, ENCOUNTERED XXXX ERRORS
553          ; 160000 = UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
554          ; 160001 = UNIT DROPPED, NOT IDLE AT START
555          ; 14XXXX = UNIT DROPPED, ENCOUNTERED XXXX ERRORS
556 003174          ;
557 003374 000000 ERTABL: .BLKW 64.
558          ERTABE: .WORD 0
559 003376 000000 SKIPT: .WORD 0 ; 1=SKIP SUBTEST 0=NO SKIP OF SUBTEST

```

```

561 .SBTTL GLOBAL TEXT MESSAGES
562 ;**
563 ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
564 ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
565 ; MORE THAN ONE TEST.
566 ;--
567 ;*
568 ;NAMES OF DEVICES SUPPORTED
569 ;-
570 003400 DEVTYP <TSV05>
003400 L#DVTYP::
003400 124 123 126 .ASCIZ /TSV05/
.EVEN

571 ;*
573 ;TEST DESCRIPTION
574 ;-
575 ;
576 003406 DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****>
003406 L#DESC::
003406 052 052 052 .ASCIZ /**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****/
.EVEN

597 ;*
598 ;BIT TO ASCII CONVERSION FOR TSSR REGISTER
599 ;-
600 ;
601 003500 003540 003543 003547 TSSRBIT:: .WORD 1#,2#,3#,4#,5#,6#,7#,8#
602 003520 003601 003605 003611 .WORD 9#,10#,11#,12#,13#,14#,15#,16#
603 003540 123 103 000 1#: .ASCIZ 'SC'
604 003543 102 111 105 2#: .ASCIZ 'BIE'
605 003547 123 103 105 3#: .ASCIZ 'SCE'
606 003553 122 115 122 4#: .ASCIZ 'RMR'
607 003557 116 130 115 5#: .ASCIZ 'NXM'
608 003563 116 102 101 6#: .ASCIZ 'NBA'
609 003567 102 111 124 7#: .ASCIZ 'BIT9'
610 003574 102 111 124 8#: .ASCIZ 'BIT8'
611 003601 123 123 122 9#: .ASCIZ 'SSR'
612 003605 117 106 114 10#: .ASCIZ 'OFL'
613 003611 102 111 124 11#: .ASCIZ 'BIT5'
614 003616 102 111 124 12#: .ASCIZ 'BIT4'
615 003623 102 111 124 13#: .ASCIZ 'BIT3'
616 003630 102 111 124 14#: .ASCIZ 'BIT2'
617 003635 102 111 124 15#: .ASCIZ 'BIT1'
618 003642 102 111 124 16#: .ASCIZ 'BIT0'
619 .EVEN
620 003650 124 123 123 SFIERR: .ASCIZ 'TSSR ERROR AFTER SOFT INIT'
621 003703 124 123 123 SFHERR: .ASCIZ 'TSSR ERROR AFTER BUS RESET'
622 003736 040 040 116 NXR: .ASCIZ / NON-EXISTANT DEVICE REGISTER/
623 003775 045 101 040 NXR# : .ASCIZ /#A ADDRESS: #06/
624 004016 045 101 040 TSSX: .ASCII /#A TSBA,TSSR EXP'D: #06#A,#06#N/
625 004056 045 101 040 .ASCIZ /#A TSBA,TSSR REC'D: #06#A,#06/
626 004115 045 116 045 FUSI: .ASCII /#N#A/
627 004121 040 040 125 USI: .ASCIZ / UNEXPECTED INTERRUPT/
628 004150 040 040 111 NSI: .ASCIZ / INTERRUPT EXPECTED, NOT RECEIVED/
629 004213 045 116 045 FNOINTR: .ASCII /#N#A/
630 004217 040 040 116 NOINTR: .ASCIZ / NO INTERRUPT WAS GENERATED/
631 004254 040 040 111 IFALT: .ASCIZ / INTERRUPT FAULT/
632 004276 045 101 040 INTX: .ASCIZ /#A CPU PC: #06#A TSBA: #06/

```

```

633 004333      040      040      042 NOINIT: .ASCIZ / "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
634 004405      040      040      042 NSINIT: .ASCIZ / "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
635 004455      040      040      042 BRINIT: .ASCIZ / "BUS-RESET" DIDN'T INITIALIZE THE DPU/
636 004525      000      000      000 NUL: .ASCIZ //
637 004526      045      116      000 NULCR: .ASCIZ /#N/
638 004531      045      101      040 EXPGOT: .ASCIZ /#A EXP'D: #06#A, REC'D: #06/
639 004565      045      116      045 EXPGT2: .ASCIZ /##A EXP'D: #06#A, #06##A REC'D: #0#A, #06/
640 004641      045      101      040 DUAD12: .ASCIZ /#A REG(W) WRITTEN TO: #06#A REG(R) READ; EXP'D: #06#A, REC'D: #06/
641 004743      122      101      115 PKTRAM: .ASCIZ 'RAM Contents Do Not Match Packet Sent'
642 005011      040      040      103 SCME: .ASCIZ / CONFIG DOESN'T MATCH MFG. MASTER/
643 005054      127      122      111 WRTMSG: .ASCIZ 'WRITE CHARACTERISTICS Failed'
644 005111      124      123      123 WRTERR: .ASCIZ 'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
645 005204      124      123      123 RDERR: .ASCIZ 'TSSR Incorrect After READ Command, More Bits Set Than SSR'
646 005276      106      101      124 SCHERR: .ASCIZ 'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
647 005370      105      122      122 RETERR: .ASCIZ 'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
648 005456      045      116      045 NOMEM: .ASCIZ '#N#A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****N'
649 005552      045      116      045 M8186: .ASCIZ '#N#A ***** 11/23A SYSTEM *****N'
650 005643      045      116      045 M8189: .ASCIZ '#N#A ***** 11/23B SYSTEM *****N'

```

```

651 .EVEN
652 .SBTTL GLOBAL ERROR REPORT SECTION
653
654
655
656
657
658

```

```

; **
; 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.
; --

```

```

659 005734      005734      005734      BGNMSG NXRERR ;NON-EXISTANT DEVICE REGISTER.
660 005734      013746 003112      PRINTX #NXRX,MODEV ;NODEV = NEXM ADDRESS.
        005740 012746 003775      MOV NODEV,-(SP)
        005744 012746 000002      MOV #NXRX,-(SP)
        005750 010600      MOV #2,-(SP)
        005752 104415      MOV SP,R0
        005754 062706 000006      TRAP C#PNTX
661 005760 004737 005766      ADD #6,SP
        005764      JSR PC,EXTEND ; PRINT EXTENSION IF REQUIRED.
        005764      ENDMSG
662 005764      L10002: TRAP C#MSG
        005764 104423

```

```

663 ;
664 ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
665 ; TO ANY OF THE ABOVE ERROR SIGNATURES.
666 ;

```

```

667 005766 005727      EXTEND: TST (PC).
668 005770 000000      EXTA: 0 ; 0 = NO EXTENSION.
669 005772 001402      BEQ 1#
670 005774 004777 177770      JSR PC,BEXTA ; APPEND EXTENSION TEXT.
671 006000      1#: PRINTX #NULCR ; PRINT A BLANK LINE
        006000 012746 004526      MOV #NULCR,-(SP)
        006004 012746 000001      MOV #1,-(SP)
        006010 010600      MOV SP,R0
        006012 104415      TRAP C#PNTX
        006014 062706 000004      ADD #4,SP
672 006020 000207      RTS PC

```

```

674                                    .SBTTL PRITSSR - PRINT TSSR CONTENTS
675
676                                    ; *
677                                    ;
678                                    ; ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
679                                    ; THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
680                                    ; BY A MESSAGE PRINTING ROUTINE
681                                    ;
682                                    ; INPUTS:
683                                    ;
684                                    ;       R1       CONTENTS OF TSSR
685                                    ;
686                                    ; SUBORDINATE ROUTINES:
687                                    ;
688                                    ;       CHKAMB   CHECK FOR AMBIGUOUS CONTENTS
689                                    ;
690                                    ; -
691
692 006022                            PRITSSR:
693 006022                            SAVREG                            ; SAVE GENERAL REGISTERS
694 006026   010104                   MOV       R1,R4                       ; SAVE THE TSSR CONTENTS
695 006030                            PRINTB   #TSSRFOR,R4               ; PRINT THE CONTENTS OF TSSR
                                  MOV       R4,-(SP)
                                  006032   012746   006413               MOV       #TSSRFOR,-(SP)
                                  006036   012746   000002               MOV       #2,-(SP)
                                  006042   010600                       MOV       SP,R0
                                  006044   104414                       TRAP      C#PNTB
                                  006046   062706   000006               ADD       #6,SP
696 006052   010400                   MOV       R4,R0                       ; GET TSSR BACK FOR CHKAMB
697 006054   004737   016044           JSR      PC,CHKAMB               ; ARE CONTENTS AMBIGUOUS ?
698 006060   103410                   BCS      5#                       ; BRANCH IF NOT
699 006062                            PRINTX   #AMBTSSR               ; SHOW CONTENTS ARE AMBIGUOUS
                                  006062   012746   006633               MOV       #AMBTSSR,-(SP)
                                  006066   012746   000001               MOV       #1,-(SP)
                                  006072   010600                       MOV       SP,R0
                                  006074   104415                       TRAP      C#PNTX
                                  006076   062706   000004               ADD       #4,SP
700 006102   010403                   5#:   MOV       R4,R3                       ; CONTENTS OF TSSR
701 006104   042703   001476           BIC      #HIADDR!FATERR!TERCLS,R3   ; CLEAR ALL MULTIPLE BIT FIELDS
702 006110   001434                   BEQ      20#                       ; NO BITS ARE SET
703 006112   012702   002630           MOV      #TMPBFR,R2               ; TEMPORARY ASCII BUFFER
704 006116   012701   003500           MOV      #TSSRBIT,R1              ; ASCII EQUIVALENT OF BITS
705 006122   005703                   10#:   TST       R3                       ; REMAINING BITS TO CONVERT
706 006124   001413                   BEQ      15#                       ; BRANCH WHEN ALL ARE DONE
707 006126   000241                   CLC                               ; CLEAR CARRY FOR SHIFT
708 006130   006103                   ROL       R3                       ; SHIFT NEXT BIT TO CARRY
709 006132   103006                   BCC      13#                       ; BRANCH IF BIT NOT SET
710 006134   011100                   MOV      (R1),R0                   ; POINTER TO BIT DEFINITION
711 006136   112022                   11#:   MOVB     (R0)+,(R2)+              ; MOVE ASCII TO BUFFER
712 006140   001376                   BNE      11#                       ; MOVE ALL BITS
713 006142   112762   000054   177777   MOVB     #',,-1(R2)               ; INSERT A COMMA TO TERMINATE
714 006150   005721                   13#:   TST       (R1)+                   ; POINT TO NEXT DESCRIPTION
715 006152   000763                   BR       10#                       ; GET THE REMAINING BITS
716 006154   105042                   15#:   CLRB     -(R2)                   ; TERMINATE THE LINE
717 006156                            PRINTX   #TSSDEF,#TMPBFR       ; PRINT THE BIT DEFINITIONS
                                  006156   012746   002630               MOV       #TMPBFR,-(SP)
                                  006162   012746   006604               MOV       #TSSDEF,-(SP)

```

TSV3 - GLOBAL AREAS MACRO M1113 14-JUN-84 15:15
 PRITSSR - PRINT TSSR CONTENTS

SEQ 0044

006166	012746	000002		MOV	#2,-(SP)		
006172	010600			MOV	SP,R0		
006174	104415			TRAP	C#PNTX		
006176	062706	000006		ADD	#6,SP		
718							
719	006202	010403	20#:	MOV	R4,R3	;GET THE TSSR CONTENTS	
720	006204	042703		BIC	#+CTERCLS,R3	;CLEAR ALL BUT TERMINATION	
721	006210	016303		MOV	TCOCOD(R3),R3	;GET THE TERMINATION CODE MEANING	
722	006214			PRINTX	#TCOASC,R3	;PRINT THE TERMINATION CODE	
	006214	010346		MOV	R3,-(SP)		
	006216	012746	006474	MOV	#TCOASC,-(SP)		
	006222	012746	000002	MOV	#2,-(SP)		
	006226	010600		MOV	SP,R0		
	006230	104415		TRAP	C#PNTX		
	006232	062706	000006	ADD	#6,SP		
723	006236	010403		MOV	R4,R3	;TSSR CONTENTS AGAIN	
724	006240	042703	177717	BIC	#+CFATERR,R3	;CLEAR ALL BUT FATAL TERMINATION	
725	006244	001416		BEQ	25#	;DON'T PRINT IF ZERO	
726	006246	006203		ASR	R3		
727	006250	006203		ASR	R3		
728	006252	006203		ASR	R3	;ALINE TERMINATION CODE FOR INDEX	
729	006254	016303	007234	MOV	TSFCOD(R3),R3	;GET THE FATAL TERMINATION CODE	
730	006260			PRINTX	#TFCASC,R3	;PRINT THE FATAL TERMINATION CODE	
	006260	010346		MOV	R3,-(SP)		
	006262	012746	006535	MOV	#TFCASC,-(SP)		
	006266	012746	000002	MOV	#2,-(SP)		
	006272	010600		MOV	SP,R0		
	006274	104415		TRAP	C#PNTX		
	006276	062706	000006	ADD	#6,SP		
731	006302	042704	176377	25#:	BIC	#+CHIADDR,R4	;CLEAR ALL BUT EXTENDED ADDRESS
732	006306	001411		BEQ	30#	;DON'T PRINT IF ZERO	
733	006310			PRINTX	#TEXASC,R4	;PRINT THE EXTENDED ADDRESS BITS	
	006310	010446		MOV	R4,-(SP)		
	006312	012746	006433	MOV	#TEXASC,-(SP)		
	006316	012746	000002	MOV	#2,-(SP)		
	006322	010600		MOV	SP,R0		
	006324	104415		TRAP	C#PNTX		
	006326	062706	000006	ADD	#6,SP		
734	006332	013703	002176	30#:	MOV	EPRTSW,R3	;PRINT MESSAGE BUFFER ADDRESS
735	006336			PRINTX	R3	;PRINT PROPER MESSAGE	
	006336	010346		MOV	R3,-(SP)		
	006340	012746	000001	MOV	#1,-(SP)		
	006344	010600		MOV	SP,R0		
	006346	104415		TRAP	C#PNTX		
	006350	062706	000004	ADD	#4,SP		
736	006354	000207		RTS	PC	;RETURN TO CALLER	

TSV3 - GLOBAL AREAS MACRO M1113 14-JUN-84 15:15
 PRITSSR - PRINT TSSR CONTENTS

SEQ 0045

739	006356				EPRT2:				
740	006356	045	116	045	EPRT1:	.ASCIZ	'#N#A *****REPLACE M7196*****'		
755	006413	045	116	045	TSSRFOR:	.ASCIZ	'#N#A TSSR = #06'		
756	006433	045	116	045	TEXASC:	.ASCIZ	'#N#A Extended Address Bits = #06'		
757	006474	045	116	045	TCOASC:	.ASCIZ	'#N#A Termination Class Code = #T'		
758	006535	045	116	045	TFCASC:	.ASCIZ	'#N#A Fatal Termination Class Code = #T'		
759	006604	045	116	045	TSSDEF:	.ASCIZ	'#N#A TSSR Bits Set: #T'		
760	006633	045	116	045	AMBTSSR:	.ASCIZ	'#N#A TSSR Contents Are Ambiguous'		
761						.EVEN			
762	006674	006714	006737	006765	TCOCOD:	.WORD	1#,2#,3#,4#,5#,6#,7#,8#		
763	006714	116	157	162	1#:	.ASCIZ	'Normal Termination'		
764	006737	124	145	162	2#:	.ASCIZ	'Termination Condition'		
765	006765	124	141	160	3#:	.ASCIZ	'Tape Status Alert'		
766	007007	106	165	156	4#:	.ASCIZ	'Function Reject'		
767	007027	122	145	143	5#:	.ASCIZ	'Recoverable Error - Tape Position One Record Down'		
768	007111	122	145	143	6#:	.ASCIZ	'Recoverable Error - Tape Was Not Moved'		
769	007160	125	156	162	7#:	.ASCIZ	'Unrecoverable Error'		
770	007204	106	141	164	8#:	.ASCIZ	'Fatal Controller Error'		
771						.EVEN			
772									
773	007234	007244	007300	007311	TSFCOD:	.WORD	1#,2#,3#,4#		
774	007244	111	156	164	1#:	.ASCIZ	'Internal Diagnostic Failure'		
775	007300	122	145	163	2#:	.ASCIZ	'Reserved'		
776	007311	102	165	163	3#:	.ASCIZ	'Bus Interface or Sanity Check Error'		
777	007355	122	145	163	4#:	.ASCIZ	'Reserved'		
778						.EVEN			

```

780 .SBTTL PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET
781
782
783 ;*
784 ;THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
785 ;THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
786 ;
787 ;INPUT:
788 ;
789 ; R0 NUMBER OF WORDS IN PACKET
790 ; R3 HIGH ORDER COMMAND PACKET ADDRESS
791 ; R4 ADDRESS OF COMMAND PACKET
792 ;
793 ; NOTE: R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
794 ;-
795 PRIPKT::
796 SAVREG ;SAVE THE REGISTERS
797 MOV R0,R5 ;SAVE NO. OF WORDS IN PACKET
798 TST KTENABLE ;ABOVE 28K UNDER TEST?
799 BNE 10$ ;BR IF YES
800 CLR R3 ;SET HIGH ORDER ADDRESS TO 0
801 10$: MOV R3,R1 ;COPY HIGH ORDER ADDRESS
802 MOV R4,R0 ;GET LOWER ADDRESS
803 ROL R0 ;SHIFT BIT 15 INTO C BIT
804 ROL R1 ;AND INTO HIGH ORDER.
805 PRINTB #PKTADD,R1,R4 ;PRINT PACKET ADDRESS
      MOV R4,-(SP)
      MOV R1,-(SP)
      MOV #PKTADD,-(SP)
      MOV #3,-(SP)
      MOV SP,R0
      TRAP C#PNTB
806 15$: ADD #10,SP
807 MOV R3,R0 ;GET HIGH ORDER ADDRESS
808 BEQ 20$ ;BR IF NOT ABOVE 28K.
809 MOV R4,R1 ;GET LOW ORDER ADDRESS
810 JSR PC,SETMAP ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
811 20$: MOV R0,R4 ;GET RETURNED PAR6 ADDRESS BIAS
812 25$: CLR R1 ;SAVE WORD NUMBER
813 MOV (R4)+,R2 ;GET PACKET CONTENTS
      PRINTB #PKTFRM,R1,R2 ;PRINT THE DATA
      MOV R2,-(SP)
      MOV R1,-(SP)
      MOV #PKTFRM,-(SP)
      MOV #3,-(SP)
      MOV SP,R0
      TRAP C#PNTB
814 000010: ADD #10,SP
815 INC R1 ;NEXT WORD NUMBER
816 CMP R1,R5 ;DONE ALL PACKET WORDS?
817 BLT 25$ ;LOOP TILL ALL DONE
818 RTS PC ;RETURN
819 045 116 045 PKTFRM: .ASCIZ '#N$A Packet Word #D1$A = #06'
820 045 116 045 PKTADD: .ASCIZ '#N$A Packet Address = #01$05'
821 .EVEN
    
```



```

007756 010346            MOV    R3,-(SP)
007760 010146            MOV    R1,-(SP)
007762 010246            MOV    R2,-(SP)
007764 012746 010010    MOV    @XORFOR,-(SP)
007770 012746 000004    MOV    @4,-(SP)
007774 010600            MOV    SP,R0
007776 104414            TRAP  C:PNTB
010000 062706 000012    ADD    @12,SP
872 010004 010300        MOV    R3,R0            ;R0 HAS XOR ON RETURN
873 010006 000207        RTS    PC             ;RETURN TO CALLER
874
875 010010        045        116        045 XORFOR: .ASCIZ 'N#A EXPD: #06#A RECV: #06#A XOR: #06'
876                .EVEN

```

```

878                               .SBTTL PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT
879
880                               ;+
881                               ;
882                               ;ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
883                               ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
884                               ;
885                               ;INPUTS:
886                               ;
887                               ;       R0       OCTAL VALUE TO CONVERT
888                               ;       R1       TABLE OF POINTERS TO ASCII EQUIVALENT
889                               ;
890                               ;-
891
892 010056                       PRIEQU:       SAVREG                               ;SAVE THE REGISTERS
893 010056                       RTS       PC                               ;RETURN TO CALLER
894 010062   000207
895
896                               .SBTTL PRIRAM - PRINT RAM ADDRESS
897                               ;+
898                               ;
899                               ;PRINT CONTROLLER RAM ADDRESS.
900                               ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
901                               ;
902                               ;INPUTS:
903                               ;
904                               ;       R4       RAM ADDRESS
905                               ;
906                               ;-
907 010064                       PRIRAM:       SAVREG                               ;SAVE R1-R5 UNTIL NEXT RETURN
908 010064                       PRINTB    #RAMFOR,R4                       ;PRINT RAM ADDRESS IN ERROR
909 010070                       MOV       R4,-(SP)
                              010070   010446                       MOV       #RAMFOR,-(SP)
                              010072   012746   010114               MOV       #2,-(SP)
                              010076   012746   000002               MOV       SP,R0
                              010102   010600                       TRAP     C#PNTB
                              010104   104414                       ADD       #6,SP
910 010112   000207                       RTS       PC                               ;RETURN
911
912 010114       045       116       045   RAMFOR: .ASCIZ '##A CONTROLLER RAM ADDRESS = #06'
913                               .EVEN

```

```

915          .SBTTL  PRIADD - PRINT MEMORY ERROR ADDRESS
916          ;+
917          ;
918          ;PRINT MEMORY ADDRESS
919          ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
920          ;
921          ; IMPLICIT INPUTS
922          ;
923          ;     ERRHI  - HIGH ORDER ADDRESS
924          ;     ERRLO  - LOW ORDER ADDRESS
925          ;
926          ;-
927  010156  PRIADD:  SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
928  010156          MOV     ERRHI,R0   ;GET HIGH ADDRESS
929  010162  013700  002234  MOV     ERRLO,R1   ;GET LOW ADDRESS
930  010166  013701  002236  MOV     R1,R2      ;COPY LOW ADDRESS
931  010172  010102          ROL     R1      ;SHIFT BIT 15 TO C BIT
932  010174  006101          ROL     R0      ;SHIFT INTO HIGH ORDER
933  010176  006100          PRINTB  #PRIA0,R0,R2 ;PRINT MEMORY ADDRESS IN ERROR
934  010200          MOV     R2,-(SP)
          010200  010246  MOV     R0,-(SP)
          010202  010046  MOV     #PRIA0,-(SP)
          010204  012746  010226  MOV     #3,-(SP)
          010210  012746  000003  MOV     SP,R0
          010214  010600          TRAP   C#PNTB
          010216  104414          ADD    #10,SP
          010220  062706  000010  RTS     PC          ;RETURN
935  010224  000207
936
937  010226      045      116      045  PRIA0: .ASCIZ  '#N#A MEMORY ERROR ADDRESS = #01#05'
938          .EVEN
939
940          .SBTTL  PRITADD - PRINT MEMORY TEST ADDRESS
941          ;+
942          ;
943          ;PRINT MEMORY ADDRESS
944          ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
945          ;
946          ; IMPLICIT INPUTS
947          ;
948          ;     ERRHI  - HIGH ORDER ADDRESS
949          ;     ERRLO  - LOW ORDER ADDRESS
950          ;
951          ;-
952  010272  PRITADD: SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
953  010272          MOV     ERRHI,R2   ;GET HIGH ADDRESS
954  010276  013702  002234  MOV     ERRLO,R1   ;GET LOW ADDRESS
955  010302  013701  002236  ;MOV    R1,R2      ;COPY LOW ADDRESS
956          ;ROL    R1      ;SHIFT BIT 15 TO C BIT
957          ;ROL    R0      ;SHIFT INTO HIGH ORDER
958          ;PRINTB  #PRIT0,R1 ;PRINT MEMORY ADDRESS LOW IN ERROR
959  010306          MOV     R1,-(SP)
          010306  010146  MOV     #PRIT0,-(SP)
          010310  012746  010354  MOV     #2,-(SP)
          010314  012746  000002  MOV     SP,R0
          010320  010600          TRAP   C#PNTB
          010322  104414

```

```

960 010324 062706 000006      ADD    #6,SP
010330      PRINTB  @PRIT1,R2      ;PRINT MEMORY ADDRESS HIGH IN ERROR
010330 010246      MOV    R2,-(SP)
010332 012746 010417      MOV    @PRIT1,-(SP)
010336 012746 000002      MOV    #2,-(SP)
010342 010600      MOV    SP,R0
010344 104414      TRAP  C#PNTB
010346 062706 000006      ADD    #6,SP
961 010352 000207      RTS    PC      ;RETURN
962
963 010354      045      116      045  PRIT0: .ASCIZ 'N/A MEMORY TEST ADDRESS LOW = #06'
964 010417      045      116      045  PRIT1: .ASCIZ 'N/A MEMORY TEST ADDRESS HIGH = #06'
965      .EVEN

```

```

967 .SBTTL SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND
968
969 ;*
970 ;
971 ;ROUTINE TO ISSUE A SPACE RECORDS
972 ;COMMAND (FORWARD OR REVERSE)
973 ;
974 ;INPUT:
975 ;
976 ; R3 NUMBER OF RECORDS TO BE SPACED OVER
977 ; BIT15 CONTROLS DIRECTION
978 ; BIT15 = 0 IS FORWARD
979 ; BIT15 = 1 IS REVERSE
980 ; R5 FIRST DEVICE UNIBUS ADDRESS
981 ;
982 ; REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
983 ;
984 ;OUTPUT:
985 ;
986 ; CARRY SET - SPACE RECORDS COMMAND OK
987 ; CLR - SPACE RECORDS FAILED
988 ;
989 ;
990 ; R0 THE CONTENTS OF R4 IS MOVED TO R0
991 ;
992 ;
993 ;IMPLICIT OUTPUT:
994 ;
995 ; TAPE HAS BEEN MOVED
996 ;
997 ;SIDE EFFECTS:
998 ;
999 ;
1000 ;
1001 ;
1002 SPACE::
1003 SAVREG ;SAVE THE GENERAL REGISTERS
1004 MOV #500.,SDELAY ;SET UP DELAY
1005 MOV #140010,80# ;SET UP COMMAND, SPACE FORWARD
1006 TST R3 ;CHECK FOR DIRECTION
1007 BMI 5# ;BR, IF REVERSE INDICATED
1008 MOV R3,90# ;LOAD UP NUMBER OF RECORDS TO SPACE
1009 BR 10# ;GO DO COMMAND
1010 5#: BIC #BIT15,R3 ;CLEAR DIRECTION BIT
1011 MOV R3,90# ;LOAD UP NUMBER OF RECORDS TO SPACE
1012 BIS #BIT8,80# ;SET REVERSE BIT IN COMMAND PACKET
1013 10#: MOV #80#,R4 ;SET UP R4 WITH PACKET ADDRESS
1014 MOV R4,TSD8(R5) ;SEND OUT COMMAND
1015 15#: JSR PC,WAITF ;WAIT FOR SSR
1016 BCS 20# ;BR, IF SSR IS SET AND OK
1017 DELAY 250 ;DELAY ABOUT .25 SECONDS
1018 MOV #250,(PC)+
1019 .WORD 0
1020 MOV L#DLY,(PC)+
1021 .WORD 0
1022 DEC -6(PC)
1023 BNE -.4
    
```

TSV3 - GLOBAL AREAS MACRO M1113 14-JUN-84 15:15
 SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

SEQ 0053

```

010574 005367 177756          DEC    -22(PC)
010600 001367                BNE    .-20
1018 010602 005337 010660     DEC    SDELAY          ;BUMP DELAY COUNTER DOWN
1019 010606 001356                BNE    15$            ;BR, IF MORE DELAY
1020 010610 000411                BR     60$            ;BR IF TROUBLE CARRY = CLEAR
1021 010612 016501 000002     20$:  MOV    TSSR(R5),R1 ;READ TSSR
1022 010616 012702 000200     MOV    #SSR,R2        ;SET UP EXPECTED
1023 010622 020201     25$:  CMP    R2,R1      ;ARE THEY OK
1024 010624 001401                BEQ    40$            ;BR, IF EQUAL = OK
1025 010626 000402                BR     60$            ;TROUBLE EXIT
1026 010630 000261     40$:  SEC                    ;SET CARRY NO TROUBLE
1027 010632 000401                BR     70$            ;EXIT
1028 010634 000241     60$:  CLC                    ;CARRY CLEAR = ERROR
1029 010636 000241     70$:
1030 010636 010400                MOV    R4,R0          ;PASS PACKET ADDRESS
1031 010640 000207                RTS    PC              ;RETURN
1032
1033
1034
1035 ;PACKET FOR SPACE COMMAND
1036
1038          010650                ;
1040                .=<..10>E177770
1041 ;COMMAND WORD
1042 010650 000000     80$:  .WORD
1043 ;NUMBER OF RECORDS TO BE SPACED OVER WORD
1044 010652 000000     90$:  .WORD
1045 010654 000000                .WORD
1046 010656 000000                .WORD
1047 010660 000000     SDELAY: .WORD    0          ;DELAY COUNTER
1048                .EVEN
1049                .SBTTL  WRTCHR - WRITE CHARACTERISTICS COMMAND

```

```

1051 ;
1052 ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
1053 ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
1054 ;
1055 ;INPUT:
1056 ; R4 ADDRESS OF PACKET FROM TEST
1057 ; R5 FIRST DEVICE UNIBUS ADDRESS
1058 ; REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
1059 ;
1060 ;OUTPUT:
1061 ; R0 TSSR CONTENTS
1062 ; CARRY SET - WRITE CHARACTERISTICS COMMAND OK
1063 ; CLR - WRITE CHARACTERISTICS FAILED
1064 ;
1065 ;IMPLICIT OUTPUT:
1066 ;
1067 ; MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
1068 ; SOFTWARE SWITCHES SET AS FOLLOWS:
1069 ; EXTFEA = EXTENDED FEATURES PRESENT
1070 ; BENBSW = BUFFER ENABLE SWITCH ON OR OFF
1071 ;
1072 ;SIDE EFFECTS:
1073 ;
1074 010662 WRTCHR::
1075 010662 SAVREG ;SAVE THE GENERAL REGISTERS
1076 010666 005037 002226 CLR BENBSW ;CLEAR BUFFER ENABLE SWITCH
1077 010672 005037 002224 CLR EXTFEA ;CLEAR EXTENDED FEATURES SW SWITCH
1078 010676 010465 000000 10#: MOV R4,TSDB(R5) ;SEND OUT COMMAND
1079 010702 004737 016336 JSR PC,CHKTSSR ;WAIT FOR SSR
1080 010706 103401 BCS 20# ;BR, IF SSR IS SET AND OK
1081 010710 000435 BR 60# ;BR IF TROUBLE CARRY = CLEAR
1082 010712 016501 000002 20#: MOV TSSR(R5),R1 ;READ TSSR
1083 010716 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
1084 010722 032701 000100 BIT #OFL,R1 ;WAS OFF LINE SET IN TSSR
1085 010726 001402 BEQ 25# ;BR, IF NO OFL SET
1086 010730 052702 000100 BIS #OFL,R2 ;MAKE THEM LOOK ALIKE
1087 010734 020201 25#: CMP R2,R1 ;ARE THEY OK
1088 010736 001401 BEQ 40# ;BR, IF EQUAL = OK
1089 010740 000421 BR 60# ;TROUBLE EXIT
1090 010742 062704 000010 40#: ADD #8,R4 ;POINT TO WRT CHARA DATA PACKET
1091 010746 011403 MOV (R4),R3 ;GET ADDRESS OF MESSAGE BUFFER
1092 010750 032763 000200 000012 BIT #X2.EXTF,XST2(R3) ;EXTENDED FEATURES BIT SET?
1093 010756 001402 BEQ 45# ;BR IF NO
1094 010760 005237 002224 INC EXTFEA ;SET EXTENDED FEATURES SW SWITCH
1095 010764 45#:
1096 010764 032763 000100 000012 BIT #X2.BUFE,XST2(R3) ;BUFFER ENABLE SWITCH SET
1097 010772 001402 BEQ 50# ;BR, IF SWITCH NOT SET
1098 010774 005237 002226 INC BENBSW ;SET SOFTWARE SWITCH FOR ENABLED
1099 011000 50#:
1100 011000 000261 SEC ;SET CARRY NO TROUBLE
1101 011002 000401 BR 70# ;EXIT
1102 011004 000241 60#: CLC ;CARRY CLEAR = ERROR
1103 011006 016500 000002 70#: MOV TSSR(R5),R0 ;RETURN TSSR CONTENTS
1104 011012 000207 RTS PC ;RETURN

```

```

1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133 011014
1134 011014
1135 011020 012704 011110
1136 011024 010465 000000
1137 011030 012703 000550
1138 011034 004737 016250
1139 011040 103417
1140 011042
    011042 012727 000372
    011046 000000
    011050 013727 002116
    011054 000000
    011056 005367 177772
    011062 001375
    011064 005367 177756
    011070 001367
1141 011072 005303
1142 011074 001357
1143 011076 000241
1144 011100 010400
1145 011102 000207
1146
1148      011110
1150 011110
1151 011110 102010
1152 011112 000000
    
```

```

.SBTTL REWIND - POSITION TAPE (REWIND) COMMAND
;
; THIS ROUTINE WILL REWIND THE SELECTED TAPE.
;
; CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
; TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
; SSR TO SET IN THE TSSR
;
; CALLING SEQUENCE:
;
; DO A SOFT INIT
; DO A WRITE CHARACTERISTICS
; JSR PC,REWIND
;
; INPUT:
;
; R5 FIRST DEVICE UNIBUS ADDRESS
;
; OUTPUT
;
; R0 THE CONTENTS OF R4 IS PASSED TO R0
;
; -
REWIND::
    SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
    MOV #RMPACK,R4                        ;GET PACKET ADDRESS
    MOV R4,TSDB(R5)                       ;SEND PACKET ADDRESS TO EXECUTE
    MOV #360,R3                            ;ENOUGH TIME FOR 2400' REEL TO REWIND
10$: JSR PC,WAITF                          ;WAIT FOR SSR TO SET
    BCS 20$                                ;LEAVE WHEN SSR IS SET
    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
20$: MOV R4,R0                              ;PASS THE PACKET ADDRESS
    RTS PC                                  ;RETURN
;
RMPACK: .=<.*10>&177770
    .WORD 102010                          ;POSTION COMMAND (REWIND)
    .WORD 0                                ;NOT USED
    
```



```

1154                   .SBTTL CKRAM - COMPARE RAM TO I/O PACKET
1155                   ;*
1156                   ;
1157                   ;ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
1158                   ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
1159                   ;
1160                   ;INPUT:
1161                   ;
1162                   ;       R4       ADDRESS OF THE COMMAND PACKET
1163                   ;       R5       FIRST DEVICE UNIBUS ADDRESS
1164                   ;
1165                   ;OUTPUT:
1166                   ;
1167                   ;       CARRY   SET - RAM MATCHES PACKET
1168                   ;       CLR     RAM DOES NOT MATCH PACKET
1169                   ;
1170                   ;IMPLICIT OUTPUT:
1171                   ;
1172                   ;       THE TABLE RAMDATA IS FILLED WITH THE
1173                   ;       DATA HELD IN RAM.
1174                   ;       RAMSIZ IS SET TO 8. FOR PRAMPKT ROUTINE
1175                   ;
1176                   ;SIDE EFFECTS:
1177                   ;
1178                   ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1179                   ;
1180                   ;-
1181
1182 011114             CKRAM::
1183 011114             SAVREG                   ;SAVE THE GENERAL REGISTERS
1184 011120 012701 002240   MOV    @RAMDATA,R1       ;ADDRESS TO SAVE THE RAM DATA
1185 011124 012702 000201   MOV    @RMPKTBEGR,R2       ;BYTE ADDRESS OF FIRST RAM DATA
1186 011130 005003       CLR     R3               ;CLEAR THE ERROR FLAG
1187 011132 004737 016336   JSR   PC,CHKTSSR       ;WAIT FOR SSR
1188 011136 112765 000000 000000   MOVB   #0,TSDB(R5)       ;SET MAINTENANCE MODE
1189 011144 004737 016336   JSR   PC,CHKTSSR       ;WAIT FOR SSR TO SET
1190 011150 010265 000000   MOV    R2,TSDB(R5)       ;SELECT NEXT RAM ADDRESS
1191 011154 004737 016336   JSR   PC,CHKTSSR       ;WAIT FOR SSR TO SET
1192 011160 116511 000000   MOVB   TSBA(R5),(R1)     ;READ THE RAM DATA
1193 011164 122124       CMPB   (R1)*,(R4)*       ;COMPARE TO EXPECTED
1194 011166 001401       BEQ     20$           ;BRANCH IF OK
1195 011170 005203       INC     R3           ;SET ERROR FLAG
1196 011172 005202       INC     R2           ;ADDRESS OF NEXT RAM LOCATION
1197 011174 020227 000210   CMP    R2,@RMPKTEND     ;REACHED END YET ?
1198 011200 003761       BLE     10$           ;BRANCH TILL ALL READ
1199 011202 005703       TST     R3           ;WAS AN ERROR FOUND ?
1200 011204 001402       BEQ     30$           ;BRANCH IF NOT
1201 011206 000241       CLC               ;CLEAR CARRY TO SHOW ERROR
1202 011210 000401       BR      50$           ;AND EXIT
1203 011212 000261       SEC               ;SHOW GOOD COMPARE
1204 011214 012737 000010 002300 50$:   MOV    #8.,RAMSIZ       ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
1205 011222 000207       RTS               ;RETURN

```

```

1207 .SBTTL CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA
1208 ;*
1209 ;
1210 ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
1211 ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
1212 ;
1213 ;INPUT:
1214 ;
1215 ; R4 ADDRESS OF THE CHARACTERISTICS DATA
1216 ; R5 FIRST DEVICE UNIBUS ADDRESS
1217 ;
1218 ;OUTPUT:
1219 ;
1220 ; CARRY SET - RAM MATCHES PACKET
1221 ; CLR - RAM DOES NOT MATCH PACKET
1222 ;
1223 ;IMPLICIT OUTPUT:
1224 ;
1225 ; THE TABLE RAMDATA IS FILLED WITH THE
1226 ; DATA HELD IN RAM.
1227 ; RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
1228 ;
1229 ;SIDE EFFECTS:
1230 ;
1231 ; THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1232 ;-
1233 CKRAM2::
1234 SAVREG ;SAVE THE GENERAL REGISTERS
1235 MOV @RAMDATA,R1 ;ADDRESS TO SAVE THE RAM DATA
1236 MOV @RMCHBEG,R2 ;BYTE ADDRESS OF FIRST RAM DATA
1237 CLR R3 ;CLEAR THE ERROR FLAG
1238 JSR PC,CHKTSSR ;WAIT FOR SSR
1239 MOVB #0,TSDB(R5) ;SET MAINTENANCE MODE
1240 JSR PC,CHKTSSR ;WAIT FOR SSH TO SET
1241 MOV R2,TSDB(R5) ;SELECT NEXT RAM ADDRESS
1242 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
1243 MOVB TSBA(R5),(R1) ;READ THE RAM DATA
1244 CMPB (R1)+,(R4)+ ;COMPARE TO EXPECTED
1245 BEQ 20$ ;BRANCH IF OK
1246 INC R3 ;SET ERROR FLAG
1247 INC R2 ;ADDRESS OF NEXT RAM LOCATION
1248 MOV #8,RAMSIZ ;ASSUME EXTFEA NOT SET
1249 TST EXTFEA ;IS THE SOFTWARE EXTENDED FEATURES SET
1250 BEQ 25$ ;BR, IF NOT SET
1251 MOV #10,RAMSIZ ;SET RAMSIZ FOR EXTEND FEATURES
1252 CMP R2,@RMCHEND ;AT END OF EXTENDED BUFFER
1253 BLE 10$ ;BR, IF NOT AT END YET
1254 BR 27$ ;AT END BRANCH
1255 CMP R2,@RMCHEND-2 ;REACHED END YET ?
1256 BLE 10$ ;BRANCH TILL ALL READ
1257 TST R3 ;WAS AN ERROR FOUND ?
1258 BEQ 30$ ;BRANCH IF NOT
1259 CLC ;CLEAR CARRY TO SHOW ERROR
1260 BR 50$ ;AND EXIT
1261 SEC 50$ ;SHOW GOOD COMPARE
1262 RTS PC ;RETURN
    
```

1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289 011360
1290 011360
1291 011364 010037 002302
1292 011370 010137 002304
1293 011374 005737 003132
1294 011400 001403
1295 011402 004737 017316
1296 011406 010001
1297 011410 005004
1298 011412 005003
1299 011414 010205
1300 011416 011264 002320
1301 011422 011164 002464
1302 011426 022221
1303 011430 001401
1304 011432 005203
1305 011434 062704 000002
1306 011440 020427 000014
1307 011444 003764
1308 011446 032765 000200 000012
1309 011454 001403
1310 011456 020427 000016
1311 011462 003755
1312 011464 005703
1313 011466 001402
1314 011470 000241
1315 011472 000401
1316 011474 000261
1317 011476 000207

.SBTTL CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS

```

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

```

1319 .SBTTL CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS
1320 ;*
1321 ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
1322 ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1323 ;ERROR PRINT ROUTINES.
1324 ;
1325 ;INPUT:
1326 ; R0 RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1327 ; R1 RECV MESSAGE BUFFER LOW ORDER ADDRESS
1328 ; R2 EXPD MESSAGE BUFFER ADDRESS
1329 ; R3 NUMBER OF BYTES TO COMPARE
1330 ;
1331 ;OUTPUT:
1332 ; CARRY SET - MESSAGE BUFFERS MATCH
1333 ; CLR - MESSAGE BUFFERS DON'T MATCH
1334 ;
1335 ;IMPLICIT OUTPUT:
1336 ; EXPMSG BUFFER IS SET TO EXPD DATA
1337 ; RECVMSG BUFFER IS SET TO RECV DATA
1338 ; RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
1339 ; RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
1340 ;-
1341 CKMSG2::
1342 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1343 CMP R3,#RECVMSG-EXPMSG,800 IS COUNT ABOVE MAX ALLOWED?
1344 BLE 5# ;800 BR IF NO
1345 MOV #RECVMSG-EXPMSG,R3,800
1346 PRINTF #DEBUGMSG ;800
1347 MOV #DEBUGMSG,-(SP)
1348 MOV #1,-(SP)
1349 MOV SP,R0
1350 TRAP C#PNTF
1351 ADD #4,SP
1352 MOV R0,RCVHIADD ;SAVE RECV HIGH ADDRESS
1353 MOV R1,RCVLOADD ;SAVE RECV LOW ADDRESS
1354 TST KTENABLE ;TESTING ABOVE 28K?
1355 BEQ 10# ;BR IF NO
1356 JSR PC,SETMAP ;RETURN ADDRESS BIASED TO PAR6 IN R0
1357 MOV R0,R1 ;GET RETURNED ADDRESS BIASED TO PAR6
1358 CLR R4 ;WORD IN BUFFER
1359 CLR R5 ;CLEAR ERROR SEEN FLAG
1360 MOVB (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
1361 MOVB (R1),RECVMSG(R4) ;SAVE RECV FOR ERROR REPORT
1362 CMPB (R2)*,(R1)* ;EXPD EQUAL RECV?
1363 BEQ 25# ;BR IF YES
1364 INC R5 ;SET ERROR SEEN FLAG
1365 ADD #1,R4 ;POINT TO NEXT BYTE
1366 CMP R4,R3 ;DONE ALL BYTES?
1367 BCE 50# ;BR IF YES
1368 BR 15# ;DO NEXT BYTE
1369 TST R5 ;ANY ERRORS SEEN?
1370 BEQ 55# ;BR IF NO
1371 CLC ;SET FAILURE
1372 BR 60# ;
1373 SEC ;SET SUCCESS
1374 RTS PC ;RETURN

```

```

1371 011632      120      122      117  DEBUGMSG: .ASCIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-' ;@@D
1372 011722      045      116      045  FERCM:  .ASCII /#N#A ***/
1373 011733      040      040      124  ERCH:   .ASCIZ / TSSR ERROR CODE REC'D = /
1374 011766      056      056      056  SIMSG:  .ASCIZ /.... AFTER DOING SOFT INIT/
1375 012021      124      105      123  TINERR: .ASCIZ /TEST: .../
1376                                     .EVEN
1377                                     ;*
1378                                     ;
1379                                     ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
1380                                     ;
1381                                     ;INPUT:
1382                                     ;
1383                                     ;      R1      CONTENTS OF TSSR AT ERROR
1384                                     ;
1385                                     ;SIDE EFFECTS:
1386                                     ;
1387                                     ;      EXECUTES DROP UNIT TO CEASE TESTING
1388                                     ;
1389                                     ;-
1390
1391 012034          BGNMSG  SFIMSG
1391 012034          SFIMSG::
1392 012034 004737 006022      JSR      PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1393 012040 004737 017202      JSR      PC,CKDROP      ;DROP UNIT, IF ALLOWED
1394 012044          ENDMSG
1394 012044          L10003:
1394 012044 104423          TRAP      C#MSG
1395
1396                                     ;*
1397                                     ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1398                                     ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
1399                                     ;
1400                                     ;INPUTS:
1401                                     ;
1402                                     ;      R1      TSSR CONTENTS
1403                                     ;      R4      ADDRESS OF COMMAND PACKET
1404                                     ;
1405                                     ;-
1406
1407 012046          BGNMSG  PKTSSR
1407 012046          PKTSSR::
1408 012046 004737 006022      JSR      PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1409 012052 012700 000004      MOV      #4,R0          ;NO. OF WORDS IN PACKET
1410 012056 004737 007366      JSR      PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
1411 012062          ENDMSG
1411 012062          L10004:
1411 012062 104423          TRAP      C#MSG

```

```

1413
1414
1415
1416
1417
1418
1419
1420
1421
1422 012064
      012064
1423 012064 004737 006022
1424 012070 012700 000002
1425 012074 004737 007366
1426 012100
      012100
      012100 104423
1427
1428
1429
1430
1431
1432
1433
1434 012102
      012102
1435 012102 004737 006022
1436 012106
      012106
      012106 104423
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449 012110
      012110
1450 012110 004737 006022
1451 012114 010200
1452 012116 010301
1453 012120 004737 014242
1454 012124
      012124
      012124 104423

```

```

;*
;PRINT ROUTINE TO PRINT THE CONTENTS OF
;TSSR AND A GET STATUS COMMAND PACKET.
;
;INPUTS:
;
;   R1      TSSR CONTENTS
;   R4      ADDRESS OF COMMAND PACKET
;-
      BGNMSG  PKTGETS
PKTGETS:
      JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
      MOV     #2,R0          ;NO. OF WORDS IN GET STATUS PACKET
      JSR     PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
      ENDMSG
L10005:
      TRAP    C#MSG
;*
;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
;
;INPUTS:
;
;   R1      TSSR CONTENTS
;   R4      ADDRESS OF COMMAND PACKET
;-
      BGNMSG  SFFMSG
SFFMSG:
      JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
      ENDMSG
L10006:
      TRAP    C#MSG
      .SBTTL  PKTMES - PRINT TSSR AND MESSAGE BUFFER
;*
;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
;BUFFER FOR ERROR REPORTS
;
;INPUTS:
;
;   R1      CONTENTS OF TSSR
;   R2      LOW ORDER MESSAGE BUFFER
;   R3      HIGH ORDER MESSAGE BUFFER ADDRESS
;   NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
;-
      BGNMSG  PKTMES
PKTMES:
      JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR
      MOV     R2,R0          ;LOW ORDER ADDRESS
      MOV     R3,R1          ;HIGH ORDER ADDRESS
      JSR     PC,PRMESS      ;PRINT THE MESSAGE BUFFER
      ENDMSG
L10007:
      TRAP    C#MSG

```

```

1456                                     .SBTTL  ADDSSR - PRINT TEST ADDRESS AND TSSR
1457                                     ;*
1458                                     ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1459                                     ;TSSR AND A MEMORY TEST ADDRESS
1460                                     ;
1461                                     ;INPUTS:
1462                                     ;
1463                                     ;      R5      FIRST DEVICE UNIBUS ADDRESS
1464                                     ;      ERRHI   HIGH ORDER MEMORY TEST ADDRESS
1465                                     ;      ERRLO   LOW ORDER MEMORY TEST ADDRESS
1466                                     ;
1467                                     ;
1468 012126                               BGNMSG  ADDSSR
012126                               ADDSSR::
1469 012126 004737 010272                JSR      PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
1470 012132 016501 000002                MOV      TSSR(R5),R1    ;GET CURRENT TSSR
1471 012136 004737 006022                JSR      PC,PRITSSR    ;PRINT THE CONTENTS OF TSSR REGISTER
1472 012142                               ENDMSG
012142                               L10010:
012142 104423                            TRAP    C#MSG

1473                                     .SBTTL  MSGEXP - PRINT WRITE CHAR. EXPD-RCV MESSAGE BUFFERS
1474                                     ;*
1475                                     ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
1476                                     ;
1477                                     ;IMPLICIT INPUTS:
1478                                     ;
1479                                     ;      EXPMSG  - EXPECTED MESSAGE BUFFER
1480                                     ;      RECMSG  - RECEIVED MESSAGE BUFFER
1481                                     ;      RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1482                                     ;      RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1483                                     ;
1484                                     ;
1485                                     ;
1486 012144                               BGNMSG  MSGEXP
012144                               MSGEXP::
1487 012144 012700 000007                MOV      #7,R0          ;ASSUME NO EXT FEATURES
1488 012150 005737 002224                TST     EXTFEA         ;EXT FEATURES SET?
1489 012154 001402                        BEQ     5#              ;BR IF NO
1490 012156 012700 000010                MOV      #8.,R0        ;EXT FEATURE BUFFER IS 8 WORDS
1491 012162 004737 014552                JSR     PC,PRMSGEXP    ;PRINT EXPD/RCV MESSAGE BUFFERS
1492 012166                               ENDMSG
012166                               L10011:
012166 104423                            TRAP    C#MSG

```

```

1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506 012170
      012170
1507 012170
      012170 010146
      012172 012746 012242
      012176 012746 000002
      012202 010600
      012204 104415
      012206 062706 000006
1508 012212
      012212 012746 012311
      012216 012746 000001
      012222 010600
      012224 104415
      012226 062706 000004
1509 012232 010100
1510 012234 004737 015122
1511 012240
      012240
      012240 104423
1512 012242 045 116
1513 012311 045 116
1514

```

```

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

```



```

1516                                     .SBTTL MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
1517                                     ;+
1518                                     ;
1519                                     ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
1520                                     ;
1521                                     ;
1522                                     ;IMPLICIT INPUTS:
1523                                     ;
1524                                     ;     EXPMSG - EXPECTED MESSAGE BUFFER
1525                                     ;     RECMMSG - RECEIVED MESSAGE BUFFER
1526                                     ;     RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1527                                     ;     RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1528                                     ;-
1529 012350                               BGNMSG  MSGSTAT
1530 012350                               MSGSTAT:
1531 012350 012701 012412                 MOV     #STATCOD,R1      ;ASCII ADDRESS TABLE
1532 012354 012100 10$:                 MOV     (R1),R0          ;DONE ALL MSG LINES?
1533 012356 001410                       BEQ     20$             ;BR IF YES
1534 012360                               PRINTX  R0              ;PRINT STATUS BIT NAMES
1535 012360 010046                       MOV     R0,-(SP)
1536 012362 012746 000001               MOV     #1,-(SP)
1537 012366 010600                       MOV     SP,R0
1538 012370 104415                       TRAP   C#PNTX
1539 012372 062706 000004               ADD     #4,SP
1540 012376 000766                       BR      10$           ;DO ANOTHER MSG LINE
1541 012400 012700 000012               20$:   MOV     #10,R0   ;NUMBER OF WORDS IN A READ STATUS BUFFER
1542 012404 004737 014552               JSR    PC,PRMSGEXP   ;PRINT EXPD/RCV MESSAGE BUFFERS
1543 012410                               ENDMMSG
1544 012410                               L10013:
1545 012410 104423                       TRAP   C#MSG
1546 012412 012430 012472 012563        STATCOD: .WORD 1$,2$,3$,4$,5$,6$,0
1547 012430 045 116 045 1$: .ASCIZ '###A Tape Bus Signale in Word #8:'
1548 012472 045 116 045 2$: .ASCIZ '###A PARRR<15> IEOT <12> IFMK <9> IRDY<6> IRWD<2>'
1549 012563 045 116 045 3$: .ASCIZ '###A IRESV2<14> IIDENT<11> IHER <8> IONL<5> IFBY<1>'
1550 012654 045 116 045 4$: .ASCIZ '###A IRESV1<13> ICER <10> ISPEED<7> ILDP<4> IFPT<0>'
1551 012745 045 116 045 5$: .ASCIZ '###A Tape Bus Signale in Word #9:'
1552 013007 045 116 045 6$: .ASCIZ '###A DATMIS<7> ILW<6> OUTRDY<5> INRDY<4>'
1553                                     .EVEN
    
```

```

1549 .SBTTL MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
1550 ;*
1551 ;
1552 ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
1553 ;
1554 ;IMPLICIT INPUTS:
1555 ;
1556 ; EXPMSG - EXPECTED MESSAGE BUFFER
1557 ; RECMSG - RECEIVED MESSAGE BUFFER
1558 ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1559 ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1560 ;
1561 BGNMSG MSGLOOP
1562 MSGLOOP::
1563 MOV @LOOPCOD,R1 ;ASCII ADDRESS TABLE
1564 MOV (R1),R0 ;DONE ALL MSG LINES?
1565 BEQ 20$ ;BR IF YES
1566 PRINTX R0 ;PRINT STATUS BIT NAMES
1567 MOV R0,-(SP)
1568 MOV @1,-(SP)
1569 MOV SP,R0
1570 TRAP C$PNTX
1571 ADD @4,SP
1572 BR 10$ ;DO ANOTHER MSG LINE
1573 MOV @10.,R0 ;NUMBER OF WORDS IN A READ STATUS BUFFER
1574 JSR PC,PRMSGEXP ;PRINT EXPD/RCV MESSAGE BUFFERS
1575 ENDMSG
1576 L10014:
1577 TRAP C$MSG
1578 LOOPCOD: .WORD 1$,2$,3$,4$,5$,6$,7$,0
1579 1$: .ASCIZ 'N/A Tape Bus Loopback Signals in Word #8:'
1580 2$: .ASCIZ 'N/A PARERR<15> IRESV2<14> IRESV1<13>'
1581 3$: .ASCIZ 'N/A IHISP=>IEOT<12> IWRT=>IIDENT<11> IREV =>ICER <10>'
1582 4$: .ASCIZ 'N/A IWM =>IFMK<09> IEDIT=>IMER <08> IFAD =>ISPEED<07>'
1583 5$: .ASCIZ 'N/A ITADO=>IRDY<06> ITAD1=>IONL <05> IERASE=>ILDOP <04>'
1584 6$: .ASCIZ 'N/A IREW =>IDBY<03> IRWU =>IRWD <02> IFEN =>IFBY <01>'
1585 7$: .ASCIZ 'N/A IGO =>IFPT<00>'
1586 .EVEN
  
```



```

1663          .SBTTL PRMESS - PRINT CONTENTS OF MESSAGE BUFFER
1664          ;*
1665          ; THIS ROUTINE PRINTS THE CONTENTS OF
1666          ; THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE TSV-05.
1667          ;
1668          ; INPUT:
1669          ;     R0     LOW ORDER ADDRESS OF MESSAGE BUFFER
1670          ;     R1     HIGH ORDER ADDRESS OF MESSAGE BUFFER
1671          ;     NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
1672          ; THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
1673          ;-
1674 014242 PRMESS: SAVREG          ;SAVE THE REGISTERS
1675 014246 010005 MOV          R0,R5          ;SAVE LOW ORDER ADDRESS
1676 014250 005737 003132 TST          KTENABLE          ;ADDRESS ABOVE 28K?
1677 014254 001001 BNE          10$          ;BR IF YES
1678 014256 005001 CLR          R1          ;SET HIGH ORDER ADDRESS TO 0
1679 014260 010103 10$: MOV          R1,R3          ;SAVE HIGH ORDER ADDRESS
1680 014262 006100 ROL          R0          ;SHIFT BIT15 TO C BIT
1681 014264 006101 ROL          R1          ;SHIFT TO HIGH ORDER FOR PRINTOUT
1682 014266 PRINTX          #PROASC,R1,R5 ;PRINT MESSAGE BUFFER ADDRESS
      014266 010546 MOV          R5,-(SP)
      014270 010146 MOV          R1,-(SP)
      014272 012746 014420 MOV          #PROASC,-(SP)
      014276 012746 000003 MOV          #3,-(SP)
      014302 010600 MOV          SP,R0
      014304 104415 TRAP          C:PNTX
      014306 062706 000010 ADD          #10,SP
1683 014312 PRINTX          #PRIASC          ;PRINT HEADER FOR CONTENTS
      014312 012746 014465 MOV          #PRIASC,-(SP)
      014316 012746 000001 MOV          #1,-(SP)
      014322 010600 MOV          SP,R0
      014324 104415 TRAP          C:PNTX
      014326 062706 000004 ADD          #4,SP
1684 014332 005004 CLR          R4          ;NUMBER OF THE NEXT WORD
1685 014334 010501 MOV          R5,R1          ;COPY LOW ORDER ADDRESS
1686 014336 010300 MOV          R3,R0          ;COPY HIGH ORDER ADDRESS
1687 014340 001403 BEQ          20$          ;BR IF NOT ABOVE 28K
1688 014342 004737 017316 JSR          PC,SETMAP          ;SETUP PAR ADDRESS IN R0
1689 014346 010005 MOV          R0,R5          ;GET PAR FORMAT ADDRESS ABOVE 28K
1690 014350 20$: PRINTX          #PRASC,R4,(R5) ;PRINT THE CONTENTS OF MEMORY BUFFER
      014350 012546 MOV          (R5)+,-(SP)
      014352 010446 MOV          R4,-(SP)
      014354 012746 014523 MOV          #PRASC,-(SP)
      014360 012746 000003 MOV          #3,-(SP)
      014364 010600 MOV          SP,R0
      014366 104415 TRAP          C:PNTX
      014370 062706 000010 ADD          #10,SP
1691 014374 005204 INC          R4          ;NUMBER OF THE NEXT
1692 014376 020427 000007 CMP          R4,#7          ;DONE ALL YET ?
1693 014402 003005 BGT          50$          ;BRANCH IF ALL DONE
1694 014404 002761 BLT          20$          ;PRINT FIRST 7 WORDS
1695 014406 032763 000200 000012 BIT          #X2.EXTF,XST2(R3) ;EXTENDED FEATUTES ON ?
1696 014414 001355 BNE          20$          ;PRINT EXTENDED STATUS WORD
1697 014416 000207 50$: RTS          PC          ;RETURN
1698 014420 045 116 045 PROASC: .ASCIZ '##N#A Message Buffer Address = #01#05'
1699 014465 045 116 045 PRIASC: .ASCIZ '##N#A Message Buffer Contents:'
1700 014523 045 116 045 PRASC: .ASCIZ '##N#A Word#D1#A: #0'
    
```

```

1702          .EVEN
1703          .SBTTL  PRMSGEXP - PRINT EXPD/RCV MESSAGE BUFFERS
1704          ;*
1705          ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
1706          ;      R0      - NUMBER OF WORDS IN BUFFER
1707          ;IMPLICIT INPUTS:
1708          ;      EXPMSG  - EXPECTED MESSAGE BUFFER
1709          ;      RECMSG  - RECEIVED MESSAGE BUFFER
1710          ;      RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1711          ;      RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1712          ;-
1713          PRMSGEXP::
1714          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
1715          MOV            R0,R5          ;SAVE NUMBER OF WORDS
1716          MOV            RCVLOADD,R0   ;GET RECV LOW ADDRESS
1717          MOV            R0,R4          ;COPY LOW ADDRESS
1718          MOV            RCVHIADD,R1   ;GET RECV HIGH ADDRESS
1719          ROL            R0             ;SHIFT BIT15 TO C BIT
1720          ROL            R1             ;SHIFT TO HIGH ORDER FOR PRINTOUT
1721          PRINTX        @PRMSG0,R1,R4  ;PRINT MESSAGE BUFFER ADDRESS
1722          MOV            R4,-(SP)
1723          MOV            R1,-(SP)
1724          MOV            @PRMSG0,-(SP)
1725          MOV            @3,-(SP)
1726          MOV            SP,R0
1727          TRAP          C:PNTX
1728          ADD            @10,SP
1729          PRINTX        @PRMSG1          ;PRINT HEADER FOR CONTENTS
1730          MOV            @PRMSG1,-(SP)
1731          MOV            @1,-(SP)
1732          MOV            SP,R0
1733          TRAP          C:PNTX
1734          ADD            @4,SP
1735          CLR            R4             ;NUMBER OF THE CURRENT WORD
1736          MOV            @EXPMSG,R1     ;GET EXPD BUFFER ADDRESS
1737          MOV            @RECMSG,R2     ;GET RECV BUFFER ADDRESS
204:          MOV            (R1),R0      ;GET EXPD
1738          MOV            (R2),R3      ;GET RECV
1739          XOR            R0,R3         ;XOR EXPD/RCV
1740          PRINTX        @PRMSG2,R4,(R1)+,(R2)+,R3
1741          MOV            R3,-(SP)
1742          MOV            (R2)+,-(SP)
1743          MOV            (R1)+,-(SP)
1744          MOV            R4,-(SP)
1745          MOV            @PRMSG2,-(SP)
1746          MOV            @5,-(SP)
1747          MOV            SP,R0
1748          TRAP          C:PNTX
1749          ADD            @14,SP
1750          INC            R4             ;NUMBER OF THE NEXT
1751          CMP            R4,R5         ;DONE ALL YET?
1752          BGE            504          ;BR IF YES
1753          BR             204          ;DO ANOTHER
1754          BR             504          ;RETURN
1755          RTS            PC
1756          504:          .ASCIZ  '##N##A Message Buffer Address = #01#05'
1757          045 PRMSG0:
1758          045 PRMSG1: .ASCIZ  '##N##A Message Buffer Contents:'
1759          045 PRMSG2: .ASCIZ  '##N##A WORD #D2##A EXPD: #06##A RECV: #06##A XOR: #06'

```

```

1739 .EVEN
1740 .SBTTL PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
1741 ;*
1742 ;
1743 ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
1744 ; ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
1745 ;
1746 ; RO - NUMBER OF BYTES IN BUFFER
1747 ;
1748 ;IMPLICIT INPUTS:
1749 ;
1750 ; EXPMSG - EXPECTED MESSAGE BUFFER
1751 ; RECMG - RECEIVED MESSAGE BUFFER
1752 ;-
1753 015122 PRBYTEXP::
1754 015122 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1755 015126 010005 MOV RO,R5 ;SAVE NUMBER OF BYTES
1756 015130 005037 002316 CLR PRMNO ;INIT ERROR COUNT
1757 015134 005004 CLR R4 ;NUMBER OF THE CURRENT BYTE
1758 015136 012701 002320 MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
1759 015142 012702 002464 MOV #RECMG,R2 ;GET RECV BUFFER ADDRESS
1760 015146 111100 20$: MOVB (R1),R0 ;GET EXPD BYTE
1761 015150 042700 177400 BIC #1C<377>,R0 ;CLEAR UPPER BYTE
1762 015154 110037 015470 MOVB RO,PRBEXP ;SAVE FOR ERROR REPORT
1763 015160 111203 MOVB (R2),R3 ;GET RECV BYTE
1764 015162 042703 177400 BIC #1C<377>,R3 ;CLEAR UPPER BYTE
1765 015166 110337 015472 MOVB R3,PRBREC ;FOR ERROR REPORT
1766 015172 XOR RO,R3 ;XOR EXPD/RECV
1767 015202 122122 CMPB (R1),.(R2). ;EXPD = RECV?
1768 015204 001431 BEQ 30$ ;BR IF YES
1769 015206 005237 002316 INC PRMNO ;UPDATE ERROR COUNT
1770 015212 023727 002316 000010 CMP PRMNO,#8. ;PRINTED 8?
1771 015220 101023 BHI 30$ ;BR IF YES
1772 015222 27$: PRINTX #PRBMSG,R4,PRBEXP,PRBREC,R3
015222 010346 MOV R3,-(SP)
015224 013746 015472 MOV PRBREC,-(SP)
015230 013746 015470 MOV PRBEXP,-(SP)
015234 010446 MOV R4,-(SP)
015236 012746 015336 MOV #PRBMSG,-(SP)
015242 012746 000005 MOV #5,-(SP)
015246 010600 MOV SP,R0
015250 104415 TRAP C#PNTX
015252 062706 000014 ADD #14,SP
1773 015256 FORCEEXIT 50$ ;880
1774 015266 000404 BR 35$ ;880
1775 015270 30$: FORCERROR 27$,NOTSSR ;880
1776 015270 35$: ;880
1777 015300 INC R4 ;NUMBER OF THE NEXT
1778 015300 005204 CMP R4,R5 ;DONE ALL YET?
1779 015302 020405 BGE 50$ ;BR IF YES
1780 015304 002001 BR 20$ ;DO ANOTHER
1781 015306 000717 50$: PRINTX #PRBTOT,PRMNO ;PRINT TOTAL ERROR COUNT
1782 015310 MOV PRMNO,-(SP)
015310 013746 002316 MOV #PRBTOT,-(SP)
015314 012746 015423 MOV #2,-(SP)
015320 012746 000002 MOV SP,R0
015324 010600

```

TSV3 - GLOBAL AREAS MACRO M1113 14-JUN-84 15:15
 PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

SEQ 0071

```

015326 104415
015330 062706 000006
1783 015334 000207 TRAP C:PNTX
1784 ADD #6,SP
1785 015336 045 116 045 PRBMSG: .ASCIZ 'N#A BYTE #D2#A EXPD: #03#A RECV: #03#A XOR: #03'
1786 015423 045 116 045 PRBTOT: .ASCIZ 'N#A NUMBER OF BYTES IN ERROR = #D2'
1787 .EVEN
1788 015470 000000 PRBEXP: .WORD 0 ;EXPD
1789 015472 000000 PRBREC: .WORD 0 ;RECV
1790 .SBTTL EXPREC - PRINT EXPD/RECV WORD DATA
1791 ;*
1792 ;
1793 ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1794 ;
1795 ;INPUTS:
1796 ;
1797 ; R1 RECEIVED DATA
1798 ; R2 EXPECTED DATA
1799 ;
1800 ;-
1801
1802 015474 BGNMSG EXPREC
015474 EXPREC::
1803 015474 004737 007740 JSR PC,PRIXOR ;PRINT THE DATA
1804 015500 ENDMSG
015500 L10017:
015500 104423 TRAP C:MSG
.SBTTL EXPBREC - PRINT EXPD/RECV BYTE DATA
1805 ;*
1806 ;
1807 ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
1808 ;
1809 ;
1810 ;INPUTS:
1811 ;
1812 ; R1 RECEIVED DATA BYTE
1813 ; R2 EXPECTED DATA BYTE
1814 ;
1815 ;-
1816
1817
1818 015502 BGNMSG EXPBREC
015502 EXPBREC::
1819 015502 004737 007610 JSR PC,PRIBXOR ;PRINT THE DATA
1820 015506 ENDMSG
015506 L10020:
015506 104423 TRAP C:MSG
.SBTTL RAMERR - PRINT RAM AND PACKET DATA
1821 ;*
1822 ;
1823 ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1824 ;
1825 ;
1826 ;INPUTS:
1827 ;
1828 ; R4 POINTER TO COMMAND PACKET
1829 ;
1830 ;
1831 ;

```



```

1832      ;IMPLICIT INPUTS:
1833      ;
1834      ;       RAMDATA      DATA AS READ FROM THE RAM
1835      ;       RAMSIZ      NUMBER OF BYTES IN PACKET
1836      ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
1837      ;
1838      ;IMPLICIT OUTPUTS:
1839      ;
1840      ;       RAMSIZ  SET TO 0
1841      ;-
1842
1843 015510      BGNMSG  RAMERR
1844 015510      RAMERR:: JSR   PC,PRAMPKT      ;PRINT RAM/PACKET DATA
1845 015510 004737 013776  ENDMSG
1846 015514      L10021: TRAP  C#MSG
1847 015514 104423      .SBTTL  RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
1848      ;+
1849      ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1850      ;
1851      ;IMPLICIT INPUTS:
1852      ;
1853      ;       R4      POINTER TO COMMAND PACKET
1854      ;
1855      ;IMPLICIT INPUTS:
1856      ;
1857      ;       RAMDATA      DATA AS READ FROM THE RAM
1858      ;       RAMSIZ      NUMBER OF BYTES IN PACKET
1859      ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
1860      ;
1861      ;       ERRHI      HIGH ORDER TEST ADDRESS
1862      ;       ERRLO      LOW ORDER TEST ADDRESS
1863      ;
1864      ;IMPLICIT OUTPUTS:
1865      ;
1866      ;       RAMSIZ  SET TO 0
1867      ;-
1868
1869 015516      BGNMSG  RAMTADD
1870 015516      RAMTADD:: JSR   PC,PRITADD      ;PRINT TEST ADDRESS
1871 015522 004737 010272  JSR   PC,PRAMPKT      ;PRINT RAM/PACKET DATA
1872 015526      ENDMSG
1873 015526      L10022: TRAP  C#MSG
1874 015526 104423      .SBTTL  RAMEXP - PRINT RAM EXPD/RECV DATA
1875      ;+
1876      ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1877      ;
1878      ;IMPLICIT INPUTS:
1879      ;
1880      ;       R1      RECEIVED DATA
1881      ;       R2      EXPECTED DATA
1882

```

```

1883          ;      R4      CONTROLLER RAM ADDRESS
1884          ; -
1885
1886 015530    BGNMSG  RAMEXP
          RAMEXP::
1887 015530    042701  177400    BIC      @+C<377>,R1      ;SAVE EXPD RAM DATA BYTE
1888 015534    042702  177400    BIC      @+C<377>,R2      ;SAVE EXPD RAM DATA BYTE
1889 015540    004737  010064    JSR      PC,PRIRAM      ;PRINT THE RAM ADDRESS
1890 015544    004737  007740    JSR      PC,PRIXOR      ;PRINT THE DATA
1891 015550    ENDMSG
          L10023:
          TRAP      C#MSG
1892
1893          .SBTTL  TIMEXP - PRINT TIMER A,B AND EXP/REC
1894          ;+
1895          ;
1896          ;PRINT ROUTINE TO DISPLAY EXPD/RCV DATA
1897          ;AND TIMER A,B HEADER MESSAGE
1898          ;
1899          ;INPUTS:
1900          ;
1901          ;      R1      RECEIVED DATA
1902          ;      R2      EXPECTED DATA
1903          ; -
1904
1905 015552    BGNMSG  TIMEXP
          TIMEXP::
1906 015552    PRINTX  @TIMSGO          ;PRINT HEADER
          MOV      @TIMSGO,-(SP)
          MOV      @1,-(SP)
          MOV      SP,R0
          TRAP    C#PNTX
          ADD      @4,SP
1907 015572    004737  007740    JSR      PC,PRIXOR      ;PRINT THE DATA
1908 015576    ENDMSG
          L10024:
          TRAP      C#MSG
1909
1910 015600    045      116      045  TIMSGO: .ASCIZ  '###A TIMER A STATUS IS IN BIT 3###A TIMER B STATUS IS IN BIT 2'
1911          .EVEN
1912          .SBTTL  BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS
1913          ;+
1914          ;
1915          ;PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
1916          ;
1917          ;INPUTS:
1918          ;
1919          ;      R1      CONTENTS OF TSSR
1920          ;      R2      DATA WRITTEN (8 BITS)
1921          ;
1922          ; -
1923
1924
1925 015700    BGNMSG  BADSSR
          BADSSR::
1926 015700    010246    MOV      R2,-(SP)      ;SAVE DATA TRANSFERRED
1927 015702    042702  177400    BIC      @177400,R2    ;GET JUST ONE BYTE

```

```

1928 015706          PRINTB  @XFERASC,R2
      015706 010246      MOV      R2,-(SP)
      015710 012746 015740  MOV      @XFERASC,-(SP)
      015714 012746 000002  MOV      @2,-(SP)
      015720 010600      MOV      SP,R0
      015722 104414      TRAP     C#PNTB
      015724 062706 000006  ADD      #6,SP
1929 015730 012602      MOV      (SP)+,R2          ;RESTORE R2
1930 015732 004737 006022  JSR      PC,PRITSSR      ;DECODE TSSR CONTENTS
1931 015736          ENDMSG
      015736          L10025:
      015736 104423      TRAP     C#MSG
1932 015740   045   116   045  XFERASC:  .ASCIZ  '#N#A Data Transferred = #03'

```

1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967

```
.SBTTL GLOBAL SUBROUTINES SECTION
;+
; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
; THAT ARE USED IN MORE THAN ONE TEST.
;--
.SBTTL SOFINIT - SOFT INITIALIZE OF CONTROLLER
;+
;
; ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
; BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
; THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
; DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
;
; INPUTS:
;
; R5 ADDRESS OF FIRST REGISTER
;
; OUTPUTS:
;
; R0 CONTENTS OF TSSR, IF ERROR
; CARRY SET IF INIT WAS OKAY
; CLEAR IF FATAL ERROR
;
; CALLING SEQUENCE:
;
; MOV #ADDRESS,R5
; JSR PC,SOFINIT
; BCS CONTINUE
; ERDF ;REPORT FATAL ERROR
;--
```

1968 015774
1969 015774
1970 016000 012765 000000 000002
1971 016006 004737 016250
1972 016012 016500 000002
1973 016016 010004
1974 016020 042704 176277
1975 016024 052704 002200
1976 016030 020400
1977 016032 001402
1978 016034 000241
1979 016036 000401
1980 016040 000261
1981 016042 000207

```
SOFINIT::
    SAVREG ; SAVE THE REGISTERS
    MOV #0,TSSR(R5) ; DO THE INIT.
    JSR PC,WAITF ; WAIT FOR SSR
    MOV TSSR(R5),R0 ;GET THE TSSR REGISTER
    MOV R0,R4 ;TSSR CONTENTS
    BIC #C<HIADDR!OFL>,R4
    BIS #SSR!NBA,R4 ;R4 HAS EXPECTED CONTENTS
    CMP R4,R0 ;ONLY EXPECTED BITS SET ?
    BEQ 5f ;BRANCH IF OKAY
    CLC ;CLEAR THE CARRY FOR ERROR
    BR 10f ;GO TO EXIT
5f: SEC ;SET THE CARRY BIT
10f: RTS PC ;RETURN TO CALLER
```

```

1983      .SBTTL  CHKAMB - CHECK TSSR FOR AMBIGUITY
1984
1985      ;+
1986      ;
1987      ;THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
1988      ;FOR AMBIGUITY
1989      ;
1990      ;INPUT:
1991      ;
1992      ;      RO      CONTENTS OF TSSR
1993      ;
1994      ;OUTPUT:
1995      ;
1996      ;      RO      CONTENTS OF TSSR
1997      ;
1998      ;      CARRY   SET - NO AMBIGUITY
1999      ;              CLR - AMBIGUOUS CONTENTS
2000      ;
2001      ;-
2002
2003      CHKAMB:
2004      SAVREG      ;SAVE THE GENERAL REGISTERS
2005      MOV         RO,R4      ;CONTENTS OF TSSR
2006      BIT         #SC,RO     ;IS BIT 15 SET ?
2007      BNE        5$         ;BRANCH IF YES
2008      BIT         #+C<NBA!OFL!SSR!MIADDR>,RO ;ANY OTHER BITS SET ?
2009      BNE        40$        ;MUST BE AN ERROR
2010      BR         45$        ;RETURN WITH SUCCESS
2011      BIT         #SSR,RO    ;IS READY BIT SET ?
2012      BNE        10$        ;BRANCH IF READY BIT IS SET.
2013      BIT         #BIT5,RO   ;IS FATAL ERROR BIT SET ?
2014      BEQ        40$        ;ERROR IF NOT
2015      BIC         #+CTERCLS,R4 ;CLEAR ALL BUT TERMINATION CODE
2016      CMP        R4,#16     ;ALL THREE BITS MUST BE SET
2017      BNE        40$        ;ERROR IF NOT SET
2018      BR         45$        ;OK IF ALL ARE SET
2019      BIT         #BIT5,RO   ;IS FATAL ERROR BIT SET ?
2020      BEQ        45$        ;ERROR IF BIT IS SET WITH SSR
2021      BIT         #BIT2!BIT1,RO ;IS THIS A FUNCTION REJECT
2022      BNE        45$        ;BR, IF TSSR IS OK
2023      CLC         ;AMBIGUOUS CONTENTS
2024      BR         50$
2025      SEC         ;SHOW SUCCESS - NO AMBIGUITY
2026      RTS        PC        ;RETURN TO CALLER

```

```

2028      .SBTTL ENAIN,DSBINT - ENABLE/DISABLE INTERRUPTS
2029      ;
2030      ; DEFAULT DISPLAY INTERRUPT HANDLERS.
2031      ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
2032      ; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
2033      ;
2034      ;
2035      ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
2036      ;
2037      000200      IOKCKIN=BIT7      ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
2038      000001      IOKSTP=BIT0      ; EXPECT "STOP" INTERRUPT.
2039      ;
2040      ; INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
2041      016144      000      INTMASK:      .BYTE      0
2042      ; INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
2043      016145      000      INTFLAG:     .BYTE      0
2044      ;
2045      ; SAVED INTERRUPT VECTOR:
2046      016146      000000      INTVEC:    .WORD      0
2047      ; SAVE CPU PC
2048      016150      000000      INTCPIC:   .WORD      0
2049      ;
2050      ; SUBROUTINE TO ENABLE INTERRUPTS:
2051      016152      010046      ENAIN:      MOV        RO,-(SP)      ;SAVE RO
2052      016154      013700      002206      MOV        IVEC,RO      ;GET POINTER TO VECTORS
2053      016160      012720      016216      MOV        @INTR,(RO)+  ;SET UP INTERRUPT VECTOR
2054      016164      012720      000340      MOV        @PRI07,(RO)+
2055      016170      012600      MOV        (SP)+,RO      ;RESTORE RO
2056      016172      011646      MOV        (SP),-(SP)
2057      016174      012766      000000      000002      MOV        @0,2(SP)      ;SET CPU TO LEVEL 0
2058      016202      000002      RTI
2059      ;
2060      ; SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
2061      016204      011646      DSBINT:   MOV        (SP),-(SP)
2062      016206      012766      000340      000002      MOV        @PRI07,2(SP)
2063      016214      000002      RTI
2064      .SBTTL INTR - INTERRUPT HANDLERS
2065      ;
2066      016216      BGNSRV INTR      ;DEFINE INTERRUPT ENTRY
2067      016216      012737      000001      002222      INTR::   MOV        @1,INTRECV  ;SET FLAG TO SHOW INTERRUPT RECEIVED
2068      016224      105037      016145      CLR      INTFLAG      ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2069      016230      132737      000001      016144      BIT      @IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
2070      016236      001003      BNE      1$           ;BR IF YES
2071      016240      152737      000001      016145      BIS      @IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.
2072      ;
2073      ; SAVE REGISTERS, MSG BUFFER, ETC.
2074      016246      1$:
2075      016246      ENDSRV
2076      016246      L10026:
2077      016246      000002      RTI

```

```

2077          .SBTTL  WAITF  - WAIT FOR SUBSYSTEM READY
2078          ;
2079          ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
2080          ;
2081          ; INPUTS:
2082          ;
2083          ;       R5      ADDRESS OF FIRST DEVICE REGISTER
2084          ;
2085          ; OUTPUTS:
2086          ;
2087          ;       R0      CONTENTS OF LAST TSSR READ
2088          ;       CARRY   SET - READY BIT SET
2089          ;               CLR - TIMEOUT WAITING FOR READY
2090          ;
2091 016250 000401 WAITF:: BR      1$          ;NOP WHEN SUPER FIXED
2092 016252 104422      BREAK          ; DO A SUPVSR BREAK FIRST.
2093 016254 012746 011000      TRAP    C$BRK
2094 016260 016500 000002      MOV     #11000,-(SP) ;25-APRIL-83 REV B - 1100 MSEC TIMER
2095 016264 105700      MOV     TSSR(R5),R0 ;READ THE TSSR REGISTER
2096          ; TSTB    R0 ;TEST FOR READY BIT SET
2097 016266 100420      BMI     3$          ; EXIT ON STOP FLAG.
2098 016270          DELAY    1 ; WAIT 100 USEC
2099          016270 012727 000001      MOV     #1,(PC)+
2100          016274 000000      .WORD    0
2101          016276 013727 002116      MOV     L$DLY,(PC)+
2102          016302 000000      .WORD    0
2103          016304 005367 177772      DEC     -6(PC)
2104          016310 001375          BNE     .-4
2105          016312 005367 177756      DEC     -22(PC)
2106          016316 001367          BNE     .-20
2107          016320 005316      DEC     (SP) ;REDUCE DELAY COUNT
2108          016322 001356      BNE     2$ ;RETRY UNTIL TIMER EXPIRES
2109          016324 000241      CLC          ; C = 0, CONTROLLER STILL RUNNING...
2110          016326 000401      BR      4$ ;...OR HUNG-UP AFTER 300 MSEC.
2111          016330 000261      SEC          ; C = 1, CONTROLLER IS STOPPED.
2112          016332 005326      DEC     (SP)+ ;RESTORE STACK WITHOUT CHANGING CARRY BIT
2113          016334 000207      RTS     PC

```

```

2107 .SBTTL CHKTSSR - CHECK TSSR FOR READY
2108 ;*
2109 ; THIS ROUTINE WAITS FOR READY IN THE TSSR
2110 ; AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
2111 ;
2112 ; INPUT:
2113 ;        R5        ADDRESS OF CSR REGISTERS
2114 ;
2115 ; OUTPUT:
2116 ;        R0        CONTENTS OF TSSR
2117 ;        CARRY     SET - OKAY
2118 ;                   CLR - NOT READY AMBIGUOUS, OR SC SET
2119 ;
2120 CHKTSSR:
2121        JSR        PC, WAITF                    ; WAIT FOR READY
2122        BCC        204                        ; BRANCH IF TIME OUT
2123        JSR        PC, CHKAMB                ; TSSR AMBIGUOUS?
2124        BCC        104                        ; BR IF YES
2125        BIT        @SC, R0                   ; SPECIAL CONDITION SET?
2126        BEQ        154                        ; BR IF NO
2127        BIT        @<SCE!BIE!RMR!NXM>, R0   ; ANY ERROR BITS SET?
2128        BEQ        154                        ; BR IF NO
2129        104:     CLC                         ; SET FAILURE
2130        BR        204                         ;
2131        154:     SEC                         ; SET SUCCESS
2132        204:     RTS        PC                ; RETURN TO CALLER
2133        .SBTTL     NXNM        - CHECK FOR NONEXISTENT MEMORY
2134 ;*
2135 ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
2136 ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
2137 ;                        "C" = 0, ALL ADDRESSES OK.
2138 ;
2139 ; CALL:    MOV ADR1, R1
2140 ;           MOV ADR2, R2
2141 ;           JSR PC, NXNM
2142 ;           RETURN                            ; TEST "C" AND PROCEED.
2143        000004    NXNM:    MOV        @24, B04                    ; SET BUSERR VECTOR.
2144        000006            MOV        @PRI04, B06
2145        005003            CLR        R3                        ; FLAG.
2146        005711        14:    TST        (R1)                   ; TEST THE ADDRESS(ES).
2147                                                ; IF ANY TRAP, CONTINUE AT 24.
2148        020102            CMP        R1, R2                   ; OTHERWISE, CONTINUE HERE.
2149        001407            BEQ        34                        ; BR IF FINISHED (NO NEXM'S).
2150        062701        000002    ADD        @2, R1                   ; SET NEXT ADDRESS...
2151        000772            BR        14                        ; ...AND CONTINUE.
2152        005103        24:    COM        R3                        ; GOT ONE, SET FLAG...
2153        012716        016440    MOV        @34, (SP)
2154        000002            RTI                                ; ...AND DISMISS INTERRUPT...
2155        016440        34:    CLRVEC    @4                        ; ...AND GIVE BACK THE VECTOR.
2156        016444            MOV        @4, R0
2157        005703            TRAP       C1CVEC
2158        001401            TST        R3                        ; DID WE CATCH ONE ??
2159        000261            BEQ        .+4                       ; NO, "C" = 0, SKIP NEXT.
2159        000207            SEC                                ; YES, "C" = 1, (R1) = NEXM ADDR.
2159                       RTS        PC

```



```

2161          .SBTTL TSTLOOP - CHECK ITERATION COUNT
2162          ;*
2163          ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
2164          ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
2165          ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
2166          ;
2167          ; CALL: LOOPTO ARG
2168          ;
2169          TSTLOOP::
2170          TST      NOITS          ; ITERATIONS INHIBITED?
2171          BNE      1$             ; YES.
2172          TST      QVP            ; NO.
2173          BMI      1$             ;LOOPS DISALLOWED IN QUICK PASS.
2174          DEC     LOOPCNT        ; BUMP LOOP COUNTER.
2175          BNE      2$
2176          1$:     CLC              ;LOOP DISALLOWED, OR DONE.
2177          BR      3$
2178          2$:     SEC              ;LOOP ENABLED.
2179          3$:     RTS             PC
2180
2181          .SBTTL TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
2182          ;*
2183          ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
2184          ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
2185          ; IN THE CURRENT RUN SEQUENCE.
2186          ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
2187          ;
2188          ;INPUT:
2189          ;
2190          ;      RO      POINTER TO TEST ID ASCIZ STRING
2191          ;
2192          ;OUTPUT:
2193          ;
2194          ;      R5      ADDRESS OF FIRST DEVICE REGISTER
2195          ;
2196          ;IMPLICIT OUTPUTS:
2197          ;
2198          ;      TSTCNT  UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
2199          ;
2200          ;SIDE EFFECTS:
2201          ;
2202          ;      INTERRUPT LEVEL IS RASIED TO LEVEL OF
2203          ;      THE DEVICE UNDER TEST
2204          ;
2205          ;-
2206
2207          TSTSETUP::
2208          MOV      RO, -(SP)      ;SAVE THE TEST ID MESSAGE
2209          CLR      SIFLAG         ; CLEAR "SOFT INIT" FLAG
2210          CLR      ERRK           ; CLEAR LOCAL ERROR COUNTER.
2211          CLR      EXTA          ; CLEAR ERROR EXTENSION FLAG.
2212          CLR      INTMASK       ; CLEAR INTERRUPT MASK (CHECK ERROR)
2213          MOV      UNITN, RO      ; GET THE UNIT NUMBER,
2214          ASL      RO             ; ... AND MAKE IT A WORD OFFSET.
2215          TST      NODEV         ; DID STARTUP FIND THE DEVICE?
2216          BEQ      4$            ; BR IF YES
2217          BPL      3$            ; BR IF NOT IDLE

```

```

2218 016550 052760 160000 003174      BIS      #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
2219 016556      ERRDF      1,NXR,NXRERR ; NO DEVICE HERE -- PRINT IT
      016556 104455      TRAP      C#ERDF
      016560 000001      .WORD      1
      016562 003736      .WORD      NXR
      016564 005734      .WORD      NXRERR
2220 016566 000407      BR        2#
2221 016570 052760 160001 003174 3# :  BIS      #160001,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
2222 016576      ERRDF      2,NOINIT ; DEVICE NOT IDLE
      016576 104455      TRAP      C#ERDF
      016600 000002      .WORD      2
      016602 004333      .WORD      NOINIT
      016604 000000      .WORD      0
2223 016606 012737 177777 003110 2# :  MOV      #-1,DUFLG ; DROP THE UNIT
2224 016614      DODU      UNITN
      016614 013700 002200      MOV      UNITN,R0
      016620 104451      TRAP      C#DODU
2225 016622      DOCLN
      016622 104444      TRAP      C#DCLN ; ABORT THE PASS
2226 016624 000423      BR        5#
2227
2228 016626      RFLAGS   R0 ; GET THE OPERATOR FLAGS.
      016626 104421      TRAP      C#RFLA
2229 016630 032700 001000      BIT      #PNT,R0 ; PRINT THE TEST NUMBERS?
2230 016634 001412      BEQ      1# ; BR IF NO
2231 016636 011600      MOV      (SP),R0 ; GET THE ID MESSAGE
2232 016640      PRINTF   #TNAM,R0 ; DISPLAY THE TEST ID
      016640 010046      MOV      R0,-(SP)
      016642 012746 016704      MOV      #TNAM,-(SP)
      016646 012746 000002      MOV      #2,-(SP)
      016652 010600      MOV      SP,R0
      016654 104417      TRAP      C#PNTF
      016656 062706 000006      ADD      #6,SP
2233 016662 005237 002212      1# :  INC      TSTCNT ; BUMP TEST COUNTER.
2234 016666      SETPRI   IPRI ; PRIORITY THAT OF DEVICE
      016666 013700 002210      MOV      IPRI,R0
      016672 104441      TRAP      C#SPRI
2235 016674 005726      5# :  TST      (SP) ; FIX UP THE STACK
2236 016676 013705 002204      MOV      CSRADDR,R5 ; ADDRESS OF TSV REGISTERS ON UNIBUS
2237 016702 000207      RTS      PC
2238 016704 045 123 045 TNAM: .ASCIZ '#S#T#A Test'
2239      .EVEN
2240      .SBTTL TSTEND - PRINT ERRORS RECEIVED
2241
2242 ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
2243 ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
2244
2245 TSTEND: RFLAGS   R0
      016720 104421      TRAP      C#RFLA
2246 016722 030027 020000      BIT      R0,#IER
2247 016726 001412      BEQ      1# ; BR IF "IER" NOT SET.
2248 016730      PRINTF   #ESUM,ERRK ; PRINT ERROR COUNT.
      016730 013746 016756      MOV      ERRK,-(SP)
      016734 012746 016760      MOV      #ESUM,-(SP)
      016740 012746 000002      MOV      #2,-(SP)
      016744 010600      MOV      SP,R0
      016746 104417      TRAP      C#PNTF

```

TSV3 - GLOBAL AREAS MACRO M1113 14-JUN-84 15:15
TSTEND - PRINT ERRORS RECEIVED

SEQ 0082

2249	016750	062706	000006			ADD	#6,SP	
2250	016754	000207			18:	RTS	PC	
2251	016756	000000			ERRK:	0		; LOCAL ERROR COUNT.
2252	016760	045	101	040	ESUM:	.ASCIZ	/#A #D#A ERRORS/	
2253	016777	105	122	122	EMAXDU:	.ASCIZ	/ERROR LIMIT REACHED -- DROPPING UNIT/	
2254						.EVEN		

```

                .SBTTL INCERK - INCREMENT LOCAL ERROR COUNT
2256
2257
2258          ;*
2259          ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
                ;*
                ;-
                INCERK: INC      ERRK          ; INCREMENT LOCAL ERROR COUNT
                MOV      RO,-(SP)          ; SAVE RO
                MOV      UNITN,RO         ; GET UNIT NUMBER,
                ASL      RO                ; ... AND MAKE IT A WORD OFFSET,
                ADD      @ERTABL,RO        ; RO GETS ADDRESS OF ERROR TABLE ENTRY.
                INC      (RO)              ; INCREMENT THE DEVICE ERROR COUNT
                BIT      @7777,(RO)        ; DID WE OVERFLOW THE FIELD?
                BNE     1$                 ; BR IF NO.
                DEC      (RO)              ; YES -- BACK IT UP TO 7777.
2260 017044 005237 016756          1$: MOV      (SP)+,RO          ; RESTORE RO
2261 017050 010046
2262 017052 013700 002200          RTS      PC                ; RETURN TO CALLER.
2263 017056 006300
2264 017060 062700 003174
2265 017064 005210
2266 017066 032710 007777
2267 017072 001001
2268 017074 005310
2269 017076 012600
2270 017100 000207
2271
2272 017102 010046          CKEMAX: MOV      RO,-(SP)          ; SAVE RO
2273 017104 013700 002200          MOV      UNITN,RO         ; GET UNIT NUMBER
2274 017110 006300          ASL      RO                ; ... AND MAKE IT A WORD OFFSET
2275 017112 016000 003174          MOV      ERTABL(RO),RO    ; GET ERROR TABLE ENTRY
2276 017116 042700 170000          BIC      @170000,RO        ; EXTRACT ERROR COUNT FIELD
2277 017122 020037 002172          CMP      RO,GERRMAX       ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
2278 017126 103004          BHIS    1$                 ; BR IF YES
2279 017130 023737 016756 002170    CMP      ERRK,LERRMAX      ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
2280 017136 103417          BLO     2$                 ; BR IF NO
2281 017140          1$: RFLAGS   RO          ; GET OPERATOR FLAGS
                TRAP    C#RFLA
2282 017142 032700 000040          BIT      @IDU,RO          ; IS DROPPING INHIBITED?
2283 017146 001013          BNE     2$                 ; BR IF YES.
2284 017150 012737 177777 003110    MOV      #-1,DUFLG         ; NO -- DROP THE UNIT
2285 017156          ERRDF   4,EMAXDU
                TRAP    C#ERDF
                .WORD   4
                .WORD   EMAXDU
                .WORD   0
2286 017166          DODU    UNITN
                MOV     UNITN,RO
                TRAP    C#DODU
                017166 013700 002200
                017172 104451
2287 017174          DOCLN
                TRAP    C#DCLN
                017174 104444          2$: MOV      (SP)+,RO          ; RESTORE RO
2288 017176 012600          RTS      PC                ; RETURN TO CALLER
2289 017200 000207
    
```

TSV3 - GLOBAL AREAS MACRO M1113 14-JUN-84 15:15
 CKDROP - CHECK IF UNIT SHOULD BE DROPPED

SEQ 0084

```

2291          .SBTTL CKDROP - CHECK IF UNIT SHOULD BE DROPPED
2292          ;*
2293          ; CHECK IF UNIT SHOULD BE DROPPED
2294          ;-
2295 017202 010046          CKDROP: MOV      RO, -(SP)
2296 017204          FORCERROR      1$,NOTSSR
2297 017214          RFLAGS      RO
2298 017216 104421          TRAP      C$RFLA
2299 017222 032700 000040  BIT      @IDU,RO
2300 017224 001010          BNE      1$
2301 017226 011600          MOV      (SP),RO
2302 017234 012737 177777 003110  MOV      @-1,DUFLG
2303 017234 013700 002200  DODU     UNITN
2304 017242 104451          MOV      UNITN,RO
2305 017244 012600          TRAP     C$DODU
2306 017246 000207          DOCLN          ;ABORT THE PASS
2307          TRAP     C$DCLN
2308          MOV      (SP)+,RO
2309          RTS      PC
2310          .SBTTL CONFIG - DETERMINE CONFIGURATION OF SYSTEM
2311          ;
2312          ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
2313          ;
2314          ; CONFIG:
2315          JSR      PC,SOFINIT
2316          RTS      PC
2317          .SBTTL KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT
2318          ;
2319          ; SUBROUTINE - ENABLE MEM MGT.
2320          ;
2321          KTON: TST      KTFLG          ; GOT KT?
2322          BEQ      1$                ; NO.
2323          MOV      @1,SRO            ; YES. ENABLE KT11.
2324          RTS      PC
2325          ;
2326          ; SUBROUTINE - DISABLE MEM MGT.
2327          ;
2328          KTOFF: TST      KTFLG          ; GOT KT11?
2329          BEQ      1$                ; NO.
2330          NOP
2331          MOV      @0,SRO            ; DISABLE KT.
2332          RTS      PC

```

```

2334                            .SBTTL SETMAP - SETUP PAR6 MAPPING
2335
2336                            ;*
2337                            ;
2338                            ; THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
2339                            ; AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
2340                            ; IS RETURNED BIASED TO PAR6.
2341                            ;
2342                            ; INPUTS:
2343                            ;
2344                            ;        R0        HIGH ORDER ADDRESS BITS
2345                            ;        R1        LOW ORDER ADDRESS BITS
2346                            ;
2347                            ; OUTPUTS:
2348                            ;
2349                            ;        R0        OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
2350                            ;        CARRY     SET IF SUCCESS
2351                            ;                     CLR IF ERROR
2352                            ;
2353                            ;-
2353 017316                     SETMAP:
2354 017316                     SAVREG                     ;SAVE R1-R4 UNTIL NEXT RETURN
2355 017322    005737    003130    TST            KTFLG                     ;SYSTEM HAVE ABOVE 28K?
2356 017326    001433                     BEQ            10$                     ;BR IF NO
2357 017330    010102                     MOV            R1,R2                     ;SAVE LOW ORDER BITS
2358                     000006                     .REPT            6
2359                            ASR            R0                     ;CONVERT WORD ADDRESS TO 32W BLOCKS
2360                            ROR            R1                     ;MAKE IT DOUBLE PRECISION
2361                            .ENDR
2362 017362    042701    000177    BIC            #177,R1                     ;ALINE FOR LOWER 4K BOUNDARY
2363 017366    020137    003130    CMP            R1,KTFLG                     ;HIGHER THAN EXISTING MEMORY?
2364 017372    103011                     BHIS           10$                     ;BR IF YES
2365 017374    010137    172354    MOV            R1,#KIPAR6                   ;SETUP MAPPING REGISTER PAR6
2366 017400    042702    160000    BIC            #160000,R2                   ;SETUP DISPLACEMENT IN PAGE
2367 017404    062702    140000    ADD            #140000,R2                   ;ADD IN PAR6 BIAS
2368 017410    010200                     MOV            R2,R0                     ;RETURN IN R0
2369 017412    000261                     SEC                                     ;SET SUCCESS
2370 017414    000401                     BR            15$                     ;
2371 017416    000241                     10$: CLC                                     ;SET FAILURE
2372 017420    000207                     15$: RTS            PC                     ;RETURN
2373                            .SBTTL FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN
2374                            ;*
2375                            ; FILL MEMORY WITH A BACKGROUND PATTERN
2376                            ;
2377                            ; INPUTS:
2378                            ;
2379                            ;        R0 = BACKGROUND PATTERN
2380                            ;        FREE    = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2381                            ;        KTFLG    = SET TO HIGHEST MEMORY LOCATION IF > 28K.
2382                            ;
2383                            ; OUTPUTS:
2384                            ;
2385                            ;        NONE
2386                            ;
2387                            ;-
2388 017422                     FILLMEM:
2389 017422                     SAVREG                     ;SAVE R1-R5 UNTIL NEXT RETURN
2390 017426    004737    017274    JSR            PC,KTOFF                   ;DISABLE KT.

```

TSV3 - GLOBAL AREAS MACRO M1113 14-JUN-84 15:15
 FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN

SEQ 0086

2391	017432	010003			MOV	R0,R3		;COPY TEST PATTERN
2392	017434	013701	003122		MOV	FREE,R1		;GET FIRST FREE LOCATION
2393	017440	013702	003124		MOV	FRESIZ,R2		;SIZE OF FREE SPACE BELOW 28K.
2394	017444	010321		104:	MOV	R3,(R1)+		;STORE A BACKGROUND WORD
2395	017446	005302			DEC	R2		;DONE ALL MEMORY IN FREE SPACE?
2396	017450	003375			BGT	104		;BR IF NO
2397	017452	005737	003130		TST	KTFLG		; GOT KT?
2398	017456	001477			BEQ	554		; NO. GET OUT.
2399	017460	004737	017256		JSR	PC,KTON		; YES. ENABLE KT.
2400	017464	005000			CLR	R0		;HIGH ORDER ADDRESS START
2401	017466	013701	003150		MOV	PST32W,R1		;GET >28K START ADDRESS (IN 32W BLOCKS)
2402		000006			.REPT	6		
2403					CLC			;CLEAR C BIT
2404					ROL	R1		;CONVERT BLOCKS TO WORDS
2405					ROL	R0		;MAKE IT DOUBLE PRECISION
2406					.ENDR			
2407	017536	004737	017316		JSR	PC,SEMAP		;SETUP PAR6 MAPPING REGISTER
2408	017542	010320		304:	MOV	R3,(R0)+		;STORE TEST PATTERN IN >28K ADDRESS
2409	017544	020027	160000		CMP	R0,#160000		;END OF PAR6 MAPPING AREA?
2410	017550	103774			BLO	304		;BR IF NO
2411	017552	162700	020000		SUB	#20000,R0		;BACKUP INTO PAR6 MAPPING BEGIN
2412	017556	062737	000200	172354	ADD	#200,#KIPAR6		;POINT TO NEXT 4K BLOCK >28K.
2413	017564	023737	172354	003130	CMP	#KIPAR6,KTFLG		;END OF MEMORY?
2414	017572	001427			BEQ	504		;BR IF YES
2415	017574	005737	003142		TST	T23A		;11/23A?
2416	017600	001407			BEQ	354		;NO KEEP GOING
2417	017602	013704	177572		MOV	SRO,R4		;GET SRO CONTENTS
2418	017606	042704	177761		BIC	#177761,R4		;CLEAR ALL BUT PAGE NUMBER
2419	017612	022704	000016		CMP	#16,R4		;SEE IF PAGE 7
2420	017616	001415			BEQ	504		;EXIT IF THERE
2421	017620	005737	003144	354:	TST	T23B		;11/23B?
2422	017624	001410			BEQ	454		;NO KEEP GOING
2423	017626	023727	172354	007600	CMP	#KIPAR6,#7600		;REACHED 18 BITS?
2424	017634	103001			BHIS	404		;YES
2425	017636	000403			BR	454		;NO KEEP GOING
2426	017640	012737	000020	172516	MOV	#20,SR3		;SET 22 BIT RELOCATION
2427	017646	000137	017542		JMP	304		;KEEP GOING ON ETC.
2428	017652	004737	017274		JSR	PC,KTOFF		; DISABLE KT.
2429	017656	000207			RTS	PC		

```

2431          .SBTTL  CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN
2432          ;*
2433          ; COMPARE MEMORY WITH A BACKGROUND PATTERN
2434          ;
2435          ; INPUTS:
2436          ;
2437          ;     RO = BACKGROUND PATTERN
2438          ;     FREE  = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2439          ;     KTFLG  = SET TO HIGHEST MEMORY LOCATION IF > 28K.
2440          ;
2441          ; OUTPUTS:
2442          ;
2443          ;     CARRY  - SET IF NO ERROR
2444          ;     CARRY  - CLR IF ERROR
2445          ;
2446          ; IMPLICIT OUTPUTS:
2447          ;
2448          ;     ERRHI  - ERROR HIGH ADDRESS
2449          ;     ERRLO  - ERROR LOW ADDRESS
2450          ;     EXPD   - EXPECTED DATA
2451          ;     RECV   - RECEIVED DATA
2452          ;-
2453          CMPMEM:
2454          SAVREG
2455          MOV     RO,R3          ;SAVE R1-R5 UNTIL NEXT RETURN
2456          JSR    PC,KTOFF       ;COPY TEST PATTERN
2457          MOV     FREE,R1       ;DISABLE KT.
2458          MOV     FRESIZ,R2     ;GET FIRST FREE LOCATION
2459          CMP     R3,(R1)       ;SIZE OF FREE SPACE BELOW 28K.
2460          BEQ    15$           ;FREE SPACE LOCATION EQUAL TO EXPD?
2461          MOV     R1,ERRLO      ;BR IF YES
2462          CLR     ERRHI         ;SAVE ADDRESS IN ERROR
2463          MOV     R3,EXPD       ;NO HIGH ADDRESS
2464          MOV     (R1),RECV     ;SAVE EXPD FOR ERROR REPORT
2465          BR     50$           ;SAVE RECV FOR ERROR REPORT
2466          TST   (R1)+         ;
2467          DEC     R2           ;POINT TO NEXT ADDRESS
2468          BGT    10$          ;DONE ALL MEMORY IN FREE SPACE?
2469          TST   KTFLG         ;BR IF NO
2470          BEQ    55$          ; GOT KT?
2471          JSR    PC,KTON       ; NO. GET OUT.
2472          CLR     RO           ; YES. ENABLE KT.
2473          MOV     PST32W,R1     ;HIGH ORDER ADDRESS START
2474          .REPT  6             ;GET >28K START ADDRESS (IN 32W BLOCKS)
2475          ROL    R1
2476          ROL    RO           ;CONVERT BLOCKS TO WORDS
2477          .ENDR              ;MAKE IT DOUBLE PRECISION
2478          BIC   #177,R1       ;ALINE 4K BOUNDARY
2479          MOV   RO,-(SP)       ;SAVE HIGH ORDER
2480          MOV   R1,-(SP)       ;SAVE LOW ORDER
2481          JSR  PC,SETMAP      ;SETUP PAR6 MAPPING REGISTER
2482          MOV   RO,R4         ;COPY ADDRESS BIASED TO PAR6
2483          MOV   (SP)+,R1      ;RESTORE LOW ORDER IN NON PAR6 FORMAT
2484          MOV   (SP)+,RO     ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
2485          CMP   R3,(R4)      ;ABOVE 28K LOCATION EQUAL EXPD?
2486          BEQ   32$          ;BR IF YES
2487          MOV   RO,ERRHI     ;SAVE HIGH ORDER IN ERROR
    
```



```

2488 020040 010137 002236      MOV      R1,ERRLO      ;SAVE LOW ORDER IN ERROR
2489 020044 010337 002230      MOV      R3,EXPD      ;SAVE EXPD FOR ERROR REPORT
2490 020050 011437 002232      MOV      (R4),RECV    ;SAVE RECV FOR ERROR REPORT
2491 020054 000421              BR        50$         ;
2492 020056 062701 000002      32$:    ADD      @2,R1    ;UPDATE NON PAR6 ADDRESS
2493 020062 005500              ADC      R0          ;MAKE IT DOUBLE PRECISION ADD
2494 020064 062704 000002      ADD      @2,R4        ;UPDATE PAR FORMAT ADDRESS
2495 020070 020427 160000      CMP      R4,@160000   ;END OF PAR6 MAPPING AREA?
2496 020074 103755              BLO     30$         ;BR IF NO
2497 020076 162704 020000      SUB      @20000,R4    ;BACKUP INTO PAR6 MAPPING BEGIN
2498 020102 062737 000200 172354  ADD      @200,@#KIPAR6 ;POINT TO NEXT 4K BLOCK >28K.
2499 020110 023737 172354 003130  CMP      @#KIPAR6,KTFLG ;END OF MEMORY?
2500 020116 101744              BLOS    30$         ;BR IF NO
2501 020120 004737 017274      50$:    JSR      PC,KTOFF  ;TURN OFF MEMORY MAPPING
2502 020124 000241              CLC                    ;SET FAILURE
2503 020126 000403              BR        60$         ;
2504 020130 004737 017274      55$:    JSR      PC,KTOFF  ;TURN OFF MEMORY MAPPING
2505 020134 000261              SEC                    ;SET SUCCESS
2506 020136 000207      60$:    RTS      PC
2507                          .SBTTL REGSAV - SAVE R1-R5 ON STACK
2508                          ;*
2509                          ;
2510                          ;ROUTINE TO
2511                          ;SAVE R1 THROUGH R5 ON THE STACK
2512                          ;
2513                          ;CALLING SEQUENCE:
2514                          ;
2515                          ;      JSR      R5,REGSAV
2516                          ;
2517                          ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
2518                          ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
2519                          ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
2520                          ;REGISTERS.
2521                          ;
2522                          ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
2523                          ;CALLED VIA A JSR PC INSTRUCTION
2524                          ;
2525                          ;-
2526
2527 020140      REGSAV:
2528 020140 010446      MOV      R4,-(SP)
2529 020142 010346      MOV      R3,-(SP)
2530 020144 010246      MOV      R2,-(SP)
2531 020146 010146      MOV      R1,-(SP)
2532 020150 010546      MOV      R5,-(SP)
2533 020152 016605 000012      MOV      10.(SP),R5
2534 020156 004736      JSR      PC,@(SP)+
2535 020160 012601      MOV      (SP)+,R1
2536 020162 012602      MOV      (SP)+,R2
2537 020164 012603      MOV      (SP)+,R3
2538 020166 012604      MOV      (SP)+,R4
2539 020170 012605      MOV      (SP)+,R5
2540 020172 000207      RTS      PC

```

```

2542                    .SBTTL GETPAT - GET 8 BIT PATTERN FROM OPERATOR
2543                    ;*
2544                    ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
2545                    ;
2546                    ;INPUTS:            NONE.
2547                    ;
2548                    ;OUTPUTS:
2549                    ;        RO        OCTAL NUMBER FROM THE OPERATOR
2550                    ;
2551                    ;CALLING SEQUENCE:
2552                    ;        JSR        PC,GETPAT
2553                    ;-
2554 020174             GETPAT::
2555 020174                    SAVREG                    ;SAVE THE GENERAL REGISTERS
2556 020200             1$:        GMANID    DATASC,PATDAT,0,377,0,377,NO
                            TRAP        C$GMAN
                            BR         10000$
                            .WORD     PATDAT
                            .WORD     T$CODE
                            .WORD     DATASC
                            .WORD     377
                            .WORD     T$LOLIM
                            .WORD     T$HILIM
                     10000$:
2557 020220                    BNCOMPLETE        1$        ;RETRY IF ERROR
                            BCC        1$
2558 020222             103367        020230        MOV        PATDAT,RO        ;DATA PATTERN FROM OPERATOR
2559 020226             000207                        RTS        PC                ;RETURN TO CALLER
2560
2561                    ;*
2562                    ;LOCAL DATA AREA
2563                    ;-
2564
2565 020230             000000                        PATDAT: .WORD    0                ;TEMPORARY STORAGE FOR DATA
2566 020232             105            116        124        DATASC: .ASCIZ  'ENTER DATA PATTERN'
2567                    .EVEN

```

```

2569      .SBTTL  GETSEL  - ISSUE MENU AND GET OPERATOR RESPONSE
2570      ;*
2571      ;ROUTINE TO ISSUE A MENU AND GET THE OPERATOR'S RESPONSE.
2572      ;
2573      ;INPUTS:
2574      ;      RO      ADDRESS OF ASCIZ STRING OF MENU
2575      ;      R1      MAXIMUM ALLOWABLE OPERATOR RESPONSE
2576      ;
2577      ;OUTPUTS:
2578      ;      RO      NUMBER OF THE OPERATOR'S SELECTION
2579      ;-
2580      GETSEL::
2581      SAVREG                                ;SAVE GENERAL REGISTERS
2582      MOV      R0,R2                        ;SAVE THE MENU ADDRESS
2583      MOV      R2,R3                        ;START OF MENU STRING
2584      TST      (R3)                        ;END OF ASCII ?
2585      BEQ      3$                          ;BRANCH IF ALL LINES DISPLAYED
2586      PRINTF  #SELASC,(R3)+                ;DISPLAY THE MENU
2587      MOV      (R3)+,-(SP)
2588      MOV      #SELASC,-(SP)
2589      MOV      #2,-(SP)
2590      MOV      SP,R0
2591      TRAP     C#PNTF
2592      ADD      #6,SP
2593      BR       2$
2594      3$:    GMANID  MENASC,MENRES,D,-1,0,-1,NO
2595      TRAP     C#GMAN
2596      BR       10001$
2597      .WORD   MENRES
2598      .WORD   T#CODE
2599      .WORD   MENASC
2600      .WORD   -1
2601      .WORD   T#LOLIM
2602      .WORD   T#HILIM
2603      10001$:
2604      BNCOMPLETE  1$                        ;RETRY IF ERROR
2605      BCC      1$
2606      MOV      MENRES,R0                    ;GET THE OPERATOR'S REPLY
2607      CMP      R0,R1                        ;COMPARE TO MAXIMUM ALLOWED
2608      BLOS     5$                          ;BRANCH IF OK
2609      PRINTF  #MENERR                       ;DISPLAY ERROR MESSAGE
2610      MOV      #MENERR,-(SP)
2611      MOV      #1,-(SP)
2612      MOV      SP,R0
2613      TRAP     C#PNTF
2614      ADD      #4,SP
2615      BR       1$                          ;RETRY
2616      5$:    RTS      PC                    ;RETURN TO CALLER
2617      045    MENERR:  .ASCIZ  '#N#A *** Menu Selection Too Large ***'
2618      045    SELASC:  .ASCIZ  '#N#T'
2619      164    MENASC:  .ASCIZ  'Enter Menu Selection: '
2620      .EVEN
2621      MENRES:  .WORD   0

```

```

2602          .SBTTL  CHKMAN - CHECK MANUAL INTERVENTION LEGALITY
2603          ;*
2604          ;
2605          ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
2606          ;
2607          ;INPUT:
2608          ;
2609          ;     NONE.
2610          ;
2611          ;OUTPUT:
2612          ;
2613          ;     CARRY  0      MANUAL INTERVENTION NOT ALLOWED
2614          ;           1      MANUAL INTERVENTION IS OK
2615          ;
2616          ;SIDE EFFECTS:
2617          ;
2618          ;     A MESSAGE IS DISPLAYED WARNING THAT TEST IS
2619          ;     NOT EXECUTED IF MANUAL INTERVENTION IS NOT
2620          ;     ALLOWED.
2621          ;
2622          ;-
2623
2624          CHKMAN::
2625          SAVREG          ;SAVE THE REGISTERS
2626          MANUAL          ;SEE IF MANUAL INTERVENTION OK
2627          TRAP  C#MANI
2628          BCOMPLETE 1#    ;BRANCH IF ALLOWED
2629          BCS  1#
2630          PRINTF #NOMAN  ;PRINT THE WARNING MESSAGE
2631          MOV  #NOMAN,-(SP)
2632          MOV  #1,-(SP)
2633          MOV  SP,R0
2634          TRAP  C#PNTF
2635          ADD  #4,SP
2636          CLC          ;CLEAR CARRY FOR ERROR
2637          1#: RTS  PC    ;RETURN
2638
2639          .ASCIZ  '#N#A *** Manual Intervention not Allowed - Test Aborted ***'
2640          .even
    
```

```

                .SBTTL  ENVIRN  -  SETUP  FREE  DIAGNOSTIC  SPACE
2635
2636                ;
2637                ;  SUBROUTINE  TO  SET-UP  VARIOUS  ENVIRONMENTAL  PARAMETERS.
2638                ;
2639  020630  ENVIRN:  MEMORY  R0
                TRAP  C0MEM
2640  020632  104431  003122  MOV  R0,FREE  ;  GET  1ST  FREE  ADDRESS...
2641  020636  062737  000002  003122  ADD  @2,FREE
2642  020644  011037  003124  MOV  (R0),FRESIZ  ;...AND  WORD  COUNT.
2643  020650  162737  000004  003124  SUB  @4,FRESIZ
2644  020656  013702  002012  MOV  L$UNIT,R2  ;  GET  NUMBER  OF  UNITS
2645  020662  162737  000007  003124  100:  SUB  @7,FRESIZ  ;  TAKE  AWAY  7  WORDS  PER  UNIT
2646  020670  005302  DEC  R2
2647  020672  001373  BNE  100
2648  020674  013700  003122  MOV  FREE,R0  ;GET  FIRST  FREE  ADDRESS
2649  020700  063700  003124  ADD  FRESIZ,R0  ;POINT  TO  LAST  FREE  ADDRESS
2650  020704  162700  000002  SUB  @2,R0  ;BACKUP  1  WORD
2651  020710  010037  003126  MOV  R0,FREEM  ;STORE  LAST  FREE  ADDRESS
2652  020714  000240  NOP  ;*****
2653  020716  012701  177520  MOV  @BDVPCR,R1  ;GET  BDV11  PCR  ADDRESS
2654  020722  010102  MOV  R1,R2  ;COPY  TO  R2
2655  020724  062702  000002  ADD  @2,R2  ;SET  THE  RANGE
2656  020730  004737  016376  JSR  PC,XN0M  ;SEE  IF  WE  HAVE  ONE
2657  020734  103001  BCC  150  ;OK  TO  SET  FLAGS
2658  020736  000423  BR  400  ;RETURN  WITH  FLAGS  CLEAR
2659  020740  013701  177520  150:  MOV  BDVPCR,R1  ;SAVE  PCR  CONTENTS
2660  020744  062701  000001  ADD  @1,R1  ;ADD  ONE  TO  IT
2661  020750  012702  177520  MOV  @BDVPCR,R2  ;GET  BDV11  PCR  ADDRESS
2662  020754  005212  INC  (R2)  ;TRY  TO  WRITE  TO  IT
2663  020756  013703  177520  MOV  BDVPCR,R3  ;GET  RESULTS
2664  020762  020103  CMP  R1,R3  ;DID  IT  CHANGE?
2665  020764  001006  BNE  200  ;NO,  MUST  BE  11/238
2666  020766  005237  003142  INC  T23A  ;SET  THE  FLAG
2667  020772  C42737  170000  002120  BIC  @170000,L$HIME  ;SUPERVISOR  COULD  BE  WRONG
2668                ;
2669                ;  PRINTF  @M8186  ;BR  400  FOR  RELEASE
2670  021000  000402  BR  400  ;TELL  THE  SYSTEM  TYPE
2671  021002  005237  003144  200:  INC  T23B  ;RETURN
2672                ;  NOP  ;SET  THE  FLAG
2673                ;  PRINTF  @M8189  ;BR  400  FOR  RELEASE
2674  021006  400:  ;TELL  THE  SYSTEM  TYPE
2675  021006  000207  RTS  PC  ;RETURN
  
```

```

2677                                     .SBTTL  KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS
2678                                     ;*
2679                                     ;
2680                                     ;ROUTINE TO INIT KT-11
2681                                     ;
2682                                     ;-
2683
2684 021010                               KTINIT:
2685 021010 005037 003130                 CLR    KTFLG                ; INIT >28K MEMORY FLAG
2686 021014 005037 003132                 CLR    KTENABLE           ; INIT TEST >28K FLAG
2687 021020 023727 002120 001577         CMP    L#HIME,#1577       ; GOT ENOUGH MEMORY (>28K)?
2688 021026 101444                         BLOS   9#                 ; NO.
2689 021030 013700 000004                 MOV    @ERRVEC,R0        ; SAVE OLD ERR VEC PTR.
2690 021034 012737 021126 000004         MOV    #2#,@ERRVEC      ; SET ERR VEC PTR.
2691 021042 005737 177572                 TST    @SRO              ; GOT KT11?
2692 021046 000240                         NOP                       ; (TRAP IF NO).
2693 021050 013737 002120 003130         MOV    L#HIME,KTFLG     ; YES. SET KT FLAG.
2694 021056 042737 000177 003130         BIC    #177,KTFLG       ;
2695 021064 010037 000004                 MOV    R0,@ERRVEC       ; RESTORE OLD ERR VEC PTR.
2696 021070 005000                         CLR    R0                ; R0 = AR DATA.
2697 021072 012701 172340                 MOV    @KIPAR0,R1       ; R1 = KI REGS PTR.
2698 021076 012761 077406 177740 1# :   MOV    #77406,-40(R1)    ; SET DESCRIPTOR REG.
2699 021104 010021                         MOV    R0,(R1)          ; SET KIPAR REG.
2700 021106 062700 000200                 ADD    #200,R0          ; BUMP AR DATA BY "4K".
2701 021112 020027 002000                 CMP    R0,#2000         ; AT "I/O"?
2702 021116 001367                         BNE    1#               ; NO.
2703 021120 012741 177600                 MOV    #177600,-(R1)    ; YES. SET KTPAR7 FOR I/O.
2704 021124 000405                         BR     9#               ;
2705
2706 021126 012716 021134                 2# :   MOV    #6#,(SP)    ; SET UP RETURN
2707 021132 000002                         RTI                       ; RTI TO NEXT LOCATION
2708
2709 021134 010037 000004                 6# :   MOV    R0,@ERRVEC  ; RESTORE OLD ERR VEC PTR.
2710
2711 021140 000207                 9# :   RTS    PC

```

```

2713      ;*
2714      ;       SUBROUTINE TO SET EXTENDED FEATURES SWITCH
2715      ;
2716      ;       Requires that SOFINIT and WRTCHR have been done previous to call.
2717      ;
2718      ;
2719      ; INPUTS:
2720      ;       R5       CURRENT UNIT NUMBER
2721      ; OUTPUTS:
2722      ;       The Extended Features Switch is set.
2723      ;
2724      ;-
2725
2726      INVERT::
2727
2728      021142 005737 002224      TST      EXTFEA      ; IS SWITCH SET?
2729      021146 001020          BNE      1$          ; YES,EXIT STAGE RIGHT!(or the next one outa town!)
2730      021150 012737 100206 021214  MOV      @100206,CMDPKT ; WRT SUB-SYS MEM CMD
2731      021156 012737 021224 021216  MOV      @WSMBK,CMDPKT+2 ; MSG BUF ADDR
2732      021164 012737 000006 021222  MOV      @6,CMDPKT+6    ; BYTE COUNT
2733      021172 012737 100010 021224  MOV      @100010,WSMBK ; INVERT THE SWITCH
2734      021200 012704 021214      MOV      @CMDPKT,R4    ; SET CMDPKT INTO R4
2735      021204 004737 010662      JSR      PC,WRTCHR    ; DO IT
2736      021210 000207      1$:      RTS      PC              ; RETURN
2737
2738      ;       COMMAND PACKET.
2739
2740      021214      =      <..+3>&177774 ;MUST BE ON MOD 4 BOUNDRY.
2741
2742      021214 000000      CMDPKT:: 0      ;1ST WORD IS TS05 COMMAND.
2743      021216 000000          0      ;2ND WORD IS THE BUFFER LOW ADDRESS.
2744      021220 000000          0      ;3RD WORD IS THE BUFFER HIGH ADDRESS.
2745      021222 000000          0      ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
2746
2747      ;       WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
2748
2749      021224 000000      WSMBK:: 0      ;1ST WORD:: SEL 0
2750      021226 000000          0      ;2ND WORD:: SEL 2
2751      021230 000000          0      ;3RD WORD:: SEL 4
2752      .EVEN
2753
2754      ;*
2755      ;       SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
2756      ;
2757      ;
2758      ; INPUTS:
2759      ; OUTPUTS:
2760      ;       The NXMFLG is set if we can test.
2761      ;       The NXMLO and NXMHI addresses are setup.
2762      ;-
2763      MEMCK::
2764
2765      021232      SAVREG      ;SAVE THE REGISTERS
2766      021236 005037 003134      CLR      NXMFLG      ;CLEAR THE FLAG
2767      021242 005037 003136      CLR      NXMLO      ;CLEAR THE TEST ADDRESS LO
2768      021246 005037 003140      CLR      NXMHI      ;CLEAR THE TEST ADDRESS HI
2769      021252 005737 003144      TST      T238      ;IS IT A 11/238?

```

```

2770 021256 001407                    BEQ        1#                    ;NO
2771 021260 023727 002120 007777    CMP        L#HIME,#7777            ; GREATER THAN 128K
2772 021266 103406                    BLO        2#                    ; NO
2773 021270 004737 021406            JSR        PC,NXMTST        ;SETUP THE ADDRESS
2774 021274 000427                    BR         13#                ;SET THE FLAG AND EXIT
2775 021276 005737 003142            1#:        TST        T23A                ;IS IT A 11/23A?
2776 021302 001413                    BEQ        4#                    ;NO
2777 021304 023727 002120 005777    2#:        CMP        L#HIME,#5777        ;GREATER THAN 96K
2778 021312 101023                    BHI        14#                ;YES,23A/23B WITH 128K MEMORY
2779 021314 023727 002120 003777    CMP        L#HIME,#3777        ;GREATER THAN 64K BUT LESS THAN 92K?
2780 021322 103403                    BLO        4#                    ;NO, CHECK 24K
2781 021324 004737 021406            JSR        PC,NXMTST        ;SETUP THE ADDRESS
2782 021330 000411                    BR         13#                ;SET THE FLAG AND EXIT
2783 021332 023727 002120 001577    4#:        CMP        L#HIME,#1577        ;GREATER THAN 24K BUT LESS THAN 64K?
2784 021340 103410                    BLO        14#                ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
2785 021342 004737 021406            JSR        PC,NXMTST        ;SETUP THE ADDRESS
2786 021346 062737 000077 003140    ADD        #77,NXMHI        ;FOOL THE 11/02 & 11/03
2787 021354 005237 003134            13#:        INC        NXMFLG        ;SET THE FLAG
2788 021360 000411                    BR         15#                ;EXIT
2789 021362 000410                    14#:        BR         15#                ;NOP FOR PRINTOUT
2790 021364                            PRINTF    @NOMEM            ;TELL THEM & EXIT ***NO PRINT*****
         021364 012746 005456        MOV        @NOMEM,-(SP)
         021370 012746 000001        MOV        #1,-(SP)
         021374 010600                MOV        SP,R0
         021376 104417                TRAP      C#PNTF
         021400 062706 000004        ADD        #4,SP
2791 021404 000207                    15#:        RTS        PC                ;RETURN
2792
2793                                    ;*
2794                                    ;        SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
2795                                    ;
2796                                    ;OUTPUTS: NXMLO,NXMHI            ;SETUP WITH NXM ADDRESS
2797                                    ;
2798                                    ;-
2799
2800 021406 013701 002120            NXMTST:  MOV        L#HIME,R1            ;GET TOP OF MEMORY
2801 021412 062701 000200                    ADD        #200,R1        ;MAKE IT I/O BLOCK OR OTHER NXM
2802 021416 042701 000177                    BIC        #177,R1
2803 021422 010102                    MOV        R1,R2            ;RESAVE RESULTS
2804                                    .REPT      6
2805                                    ASL        R1                ;PUT IN PLACE FOR XFER
2806                                    .ENDR
2807 021440 010137 003136                    MOV        R1,NXMLO        ;SAVE TEST ADDRESS LOW
2808                                    .REPT     10.
2809                                    ASR        R2                ;PUT IN PLACE FOR XFER
2810                                    .ENDR
2811 021470 042702 177700                    BIC        #177700,R2      ;DON'T WANT ILA!
2812 021474 010237 003140                    MOV        R2,NXMHI        ;SAVE TEST ADDRESS HIGH
2813 021500 000207                    RTS        PC                ;RETURN
2814
2815 021502                            ENDMOD

```


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

SEQ 0096

```
7                                     .TITLE  TSV4 - MISCELLANEOUS SECTIONS
8
9 021502                               BGNMOD  TSV4
   021502                               TSV4::
10
16
17
18
19                                     .SBTTL  PROTECTION TABLE
20 021502                               BGNPROT
   021502                               L$PROT::
21 021502 177777 177777 177777         .WORD  -1, -1, -1, -1           ;NO DEVICE PROTECTION REQUIRED.
22 021512                               ENDPROT
```

```

24                                     .SBTTL INITIALIZE SECTION
25
26                                     ;**
27                                     ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
28                                     ;AT THE BEGINNING OF EACH PASS.
29
30                                     ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
31                                     ;IF "CONTINUE", NOTHING IS REQUIRED.
32
33                                     ;--
34                                     ;*
35                                     ;INSERT TEMPORARY JUMP TO ODT
36                                     ;-
37 021512                               BGNINIT
    021512                               L#INIT::
38 021512 005037 002224                 40#: CLR     EXTFEA
39 021516 005037 003134                 CLR     NXMFLG
40 021522 012737 006356 002176         MOV     @EPRT1,EPRTSW           ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
41 021530 005037 003152                 CLR     SIFLAG               ;CLEAR "SOFT INIT" FLAG
42 021534 005037 003132                 CLR     KTENABLE             ;CLEAR TEST ABOVE 28K FLAG
43 021540 005037 002300                 CLR     RAMSIZ               ;CLEAR RAM SIZE FOR RAMERR ROUTINE
44 021544                               READEF  @EF.CONTINUE
    021544 012700 000036                 MOV     @EF.CONTINUE,RO
    021550 104447                               TRAP   C#REFG
45 021552                               BNCOMPLETE 1#
    021552 103023                               BCC    1#
46 021554 023737 002200 002012         CMP     UNITN,L#UNIT           ;UNIT IN RANGE?
47 021562 103070                               BHIS   4#                     ;BR IF NO.
48 021564 005737 003110                 TST    DUFLG                 ;DROPPED UNIT?
49 021570 100472                               BMI    NXTU                   ;BR IF YES
50 021572 013701 002200                 MOV     UNITN,R1
51 021576 006301                               ASL    R1
52 021600 005761 003174                 TST    ERTABL(R1)
53 021604 001516                               BEQ    SETU
54 021606 032761 040000 003174         BIT    @BIT14,ERTABL(R1)     ;DROPPED?
55 021614 001060                               BNE    NXTU
56 021616                               EXIT    INIT                   ;DO NOTHING IF "CONTINUE".
    021616 104432                               TRAP   C#EXIT
    021620 000416                               .WORD  L10030-.
57 021622                               1#: READEF  @EF.NEW
    021622 012700 000035                 MOV     @EF.NEW,RO
    021626 104447                               TRAP   C#REFG
58 021630                               BNCOMPLETE NXTU               ;TAKE NEXT UNIT IF NOT NEW PASS.
    021630 103052                               BCC    NXTU
59 021632                               READEF  @EF.START
    021632 012700 000040                 MOV     @EF.START,RO
    021636 104447                               TRAP   C#REFG
60 021640                               BCOMPLETE 2#
    021640 103404                               BCS    2#
61 021642                               READEF  @EF.RESTART
    021642 012700 000037                 MOV     @EF.RESTART,RO
    021646 104447                               TRAP   C#REFG
62 021650                               BNCOMPLETE 31#
    021650 103031                               BCC    31#
63 021652                               2#: BRESET
    021652 104433                               TRAP   C#RESET           ;1ST PASS, BUS-INIT...
    ;BUS RESET.

```

```

65 021654 005037 002212          CLR      TSTCNT          ;NUMBER OF TESTS RUN IN PASS
66 021660 005037 002220          CLR      FATFLG         ;CLEAR FATAL ERROR COUNT
67 021664 005037 003142          CLR      T23A          ;CLEAR 11/23A FLAG
68 021670 005037 003144          CLR      T23B          ;CLEAR 11/23B FLAG
69                               ;      MOV      #340,-(SP)
70                               ;      MOV      #20,-(SP)          ;RETURN TO DEBUGGER
71                               ;      JMP      0.ODT          ;;ENTER THE DEBUGGER
72 021674 005037 003376          CLR      SKIPT          ;CLEAR THE SUBTEST "SKIPPER"
73 021700                               20$:
74 021700 012737 177777 002202    MOV      #-1,QVP        ;...QUICK VERIFY...
75 021706 004737 020630          JSR      PC,ENVIRN      ;SET ENVIRONMENT.
76 021712 004737 021010          JSR      PC,KTINIT     ;INITIALIZE KT MEMORY MANAGEMENT
77 021716 012700 003174          MOV      @ERTABL,RO
78 021722 005020 003374          30$: CLR      (RO)+      ;CLEAR THE ERROR TABLE
79 021724 020027 003374          CMP      RO,@ERTABE
80 021730 103774          BLO     30$
81 021732 000404          BR      4$
82 021734 005037 002202          31$: CLR      QVP
83 021740 000137 022010          JMP      PASRPT        ;GO REPORT THE STATUS
84
85 021744                               4$:
86 021744 012737 177777 002200    NEWPAS: MOV     #-1,UNITN ;INIT UNIT NUMBER...
87 021752 005037 002216          CLR     DEVCNT        ;CLEAR COUNT OF DEVICES RUNNING
88 021756                               NXTU:
89 021756 104422          TRAP   C#BRK
90 021760 005237 002200          INC     UNITN
91 021764 023737 002200 002012    CMP     UNITN,L#UNIT   ;...AND SET NEXT UNIT NUMBER.
92 021772 103423          BLO     SETU
93 022002 000401          MOV     #-1,DUFLG
94 022004 000401          BR      11$
95 022004 104444          DOCLN
96 022006 000240          TRAP   C#DCLN
97 022010 000240          NOP
98 022010 023727 002012 000001    11$: PASRPT:
99 022016 101752          CMP     L#UNIT,#1     ;HOW MANY UNITS SELECTED?
100 022020 005737 002216          BLOS   NEWPAS        ;BR IF ONLY 1
101 022024 001747          TST    DEVCNT        ;ARE ANY STILL RUNNING?
102 022026 104421          BEQ    NEWPAS        ;BR IF NO
103 022030 032700 000100          RFLAGS RO
104 022034 001343          TRAP   C#RFLA
105 022036 104424          BIT    @ISR,RO
106 022040 000741          BNE    NEWPAS        ;SHOULD WE PRINT STATISTICS
107 022042 000741          BR     NEWPAS        ;BR IF NO
108
109 022042 000741          DORPT
110 022042 013700 002200          TRAP   C#DRPT
111 022046 104442          BR     NEWPAS
112
113 022042 000741          SETU:
114 022042 013700 002200          GPHARD UNITN,RO      ;GET UNIT N P-TABLE POINTER.
115 022046 104442          MOV    UNITN,RO
116 022050 103342          TRAP   C#GPHRD
117 022052 005037 003110          BNCOMPLETE NXTU     ;BR IF UNIT NOT AVAILABLE.
118 022054 005237 002216          BCC    NXTU
119 022056 012001          CLR    DUFLG         ;CLEAR "DROPPED" FLAG.
120 022058 010137 002204          INC    DEVCNT
121 022060 010137 002204          MOV    (RO)+,R1
122 022062 010137 002204          MOV    R1,CSRADDR   ;GET 1ST REGISTER ADDRESS.
123 022064 010137 002204          ;ADDRESS OF REGISTERS OF UNIT UNDER TEST

```

```

115
116 022070 012001          MOV      (R0)+,R1          ;GET VECTOR ADDRESS.
117                      ;MOV      (R0),R2          ;GET INTERRUPT PRIORITY
118                      ;MOV      R2,IPRI        ;SET INTERRUPT PRIORITY.
119 022072 010137 002206   MOV      R1,IVEC          ;SET INTERRUPT VECTOR POINTER...
120 022076 012721 016216   MOV      @INTR,(R1)+     ;...VECTOR...
121 022102 013721 002210   MOV      IPRI,(R1)+     ;...AND PRIORITY.
122
123 022106                1$:
124                      ;      TST      QVP          ;1ST PASS ??
125                      ;      BEQ      5$          ;NO, SKIP THE PASS 1 STUFF.
126
127                      ;
128                      ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
129                      ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
130                      ;
131 022106 013701 002200   MOV      UNITN,R1
132 022112 006301          ASL      R1
133 022114 052761 100000 003174  BIS      @BIT15,ERTABL(R1) ;SAY DEVICE RUNNING
134 022122 005037 005770   CLR      EXTA           ;CLEAR ERROR EXTENSION FLAG.
135 022126 023727 002012 000001  CMP      L$UNIT,#1      ;ARE WE TESTING MULTIPLE UNITS?
136 022134 101416          BLOS    10$            ;BR IF NO.
137 022136                RFLAGS    RO          ;YES -- GET OPERATOR FLAGS.
138 022136 104421          TRAP    C$RFLA
139 022140 032700 001000   BIT      @PNT,RO        ;SHOULD WE PRINT UNIT #?
140 022144 001412          BEQ     10$            ;BR IF NOT.
141 022146                PRINTF    @PUNIT,UNITN ;PRINT THE UNIT #
142 022146 013746 002200   MOV      UNITN,-(SP)
143 022152 012746 022240   MOV      @PUNIT,-(SP)
144 022156 012746 000002   MOV      @2,-(SP)
145 022162 010600          MOV      SP,RO
146 022164 104417          TRAP    C$PNTF
147 022166 062706 000006   ADD     @6,SP
148 022172                10$:
149 022172 005037 003112   CLR      NODEV
150 022176 013701 002204   MOV      CSRADDR,R1    ;ADDRESS OF FIRST REGISTER
151 022202 010102          MOV      R1,R2         ;START OF REGISTERS
152 022204 062702 000002   ADD     @TSSR,R2       ;ADDRESS OF TSSR REGISTER
153 022210 004737 016376   JSR     PC,XNXM        ;TEST BOTH CONTROLLER REGISTERS...
154 022214 103005          BCC     2$            ;...AND BR IF ALL OK.
155 022216 010137 003112   MOV      R1,NODEV     ;FLAG DEVICE AS NON-EXISTENT
156 022222 012737 177777 003110  MOV      @-1,DUFLG    ;DROP THIS UNIT.
157 022230                2$:
158                      ;
159                      ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
160                      ;
161                      5$:
162 022230                SETPRI    @PRI00          ;ENABLE INTERRUPTS.
163 022230 012700 000000   MOV      @PRI00,RO
164 022234 104441          TRAP    C$SPRI
165 022236                ENDINIT
166 022236                L10030:
167 022236 104411          TRAP    C$INIT
168
169 022240 045 116 045 PUNIT: .ASCIZ /#N#N#A***** TESTING UNIT #D2#A *****/
170 .EVEN

```

.SBTTL ADD AND DROP UNITS SECTIONS

```

160
161
162
163
164
165
166
167 022306
    022306
168 022306 010001
169 022310 006301
170 022312 052761 100000 003174
171 022320 042761 040000 003174
172 022326
    022326 010046
    022330 012746 022354
    022334 012746 000002
    022340 010600
    022342 104417
    022344 062706 000006
173 022350
    022350 000167
    022352 000026
174 022354 045 116 045 1$:
175
176
177 022402
    022402
    022402 104452
178
179
180
181
182
183
184
185
186
187
188
189 022404
    022404
190 022404 012737 177777 003110
191 022412 010001
192 022414 006301
193 022416 052761 140000 003174
194 022424 000240 000240 000240
195 022432
    022432 010046
    022434 012746 022460
    022440 012746 000002
    022444 010600
    022446 104417
    022450 062706 000006
196 022454
    022454 000167
    022456 000030

```

```

; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
; OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
;--
BGNAU
L$AU::
MOV R0,R1 ; GET UNIT TO BE ADDED (R0)
ASL R1 ; MAKE IT A WORD INDEX
BIS #100000,ERTABL(R1) ; SET THE "ACTIVE" BIT
BIC #40000,ERTABL(R1) ; CLEAR THE "DROPPED" BIT
PRINTF #1$,R0
MOV RO,-(SP)
MOV #1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP
EXIT AU
.WORD J$JMP
.WORD L10031-2-.
.ASCIZ /#N$A UNIT #D$A ADDED/
.EVEN

ENDAU ; UNUSED.
L10031:
TRAP C$AU

; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
; TO BE REMOVED FROM THE TEST LIST.
;
; SUPVSR DOES THE "DROPPING". THIS IS JUST TO TELL THE MAN,
; "DROPPED" UNITS ARE RE-SELECTED ON OPERATOR "STA" OR "ADD"
; COMMAND, OTHERWISE REMAIN INACTIVE. THE "DISPLAY" COMMAND
; WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
; WHICH ARE STILL ACTIVE.
; UPON ENTRY, R0 CONTAINS THE UNIT TO BE DROPPED.
BGNDU
L$DU::
MOV #-1,DUFLG
MOV R0,R1
ASL R1
BIS #140000,ERTABL(R1) ; SAY DROPPED
240,240,240 ; ??????????
PRINTF #1$,R0
MOV RO,-(SP)
MOV #1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP
EXIT DU
.WORD J$JMP
.WORD L10032-2-.

```

TSV4 - MISCELLANEOUS SECTIONS
ADD AND DROP UNITS SECTIONS

MACRO M1113 14-JUN-84 15:15

SEQ 0101

```

197 022460      045      116      045 1$: .ASCIZ /%N%A UNIT %D%A DROPPED/
198                                     .EVEN
199 022510                                     ENDDU
    022510                                     L10032:
    022510 104453                                     TRAP C$DU
200                                     ;**
201                                     ; AUTO-DROP CODE SECTION.
202                                     ;--
203 022512                                     BGNAUTO
    022512                                     L$AUTO::
204 022512 013705 002204                                     MOV CSRADDR,R5 ;POINT TO DEVICE REGISTER
205 022516 012703 000550                                     MOV #360.,R3 ;ENOUGH TIME FOR 2400' REEL TO REWIND
206 022522 004737 016250 10$: JSR PC,WAITF ;WAIT FOR SSR TO SET
207 022526 103420                                     BCS 20$ ;LEAVE WHEN SSR IS SET
208 022530 DELAY 250. ;WAIT FOR .25 SECONDS
    022530 012727 000372                                     MOV #250.,(PC)+
    022534 000000                                     .WORD 0
    022536 013727 002116                                     MOV L$DLY,(PC)+
    022542 000000                                     .WORD 0
    022544 005367 177772                                     DEC -6(PC)
    022550 001375                                     BNE .-4
    022552 005367 177756                                     DEC -22(PC)
    022556 001367                                     BNE .-20
209 022560 005303                                     DEC R3 ;BUMP COUNTER DOWN
210 022562 001357                                     BNE 10$ ;KEEP GOING
211 022564 004737 017202                                     JSR PC,CKDROP ;TRY AND DROP UNIT
212 022570                                     20$:
213 022570                                     ENDAUTO ; UNUSED.
    022570                                     L10033:
    022570 104461                                     TRAP C$AUTO

```

```

215
216
217
218
219
220
221
222 022572
    022572
223 022572 013705 002204
224 022576 005737 003110
225 022602 100405
226
227
228 022604 012765 000000 000002
229 022612 004737 016250
230 022616
231 022616
    022616
    022616 104412
232
233
234
235
236 022620
    022620
237 022620
    022620 012746 023062
    022624 012746 000001
    022630 010600
    022632 104416
    022634 062706 000004
238 022640 010246
239 022642 010346
240 022644 010446
241 022646 012704 003174
242 022652 005003
243 022654 011402
244 022656 001467
245 022660 100066
246 022662 032702 040000
247 022666 001015
248 022670 042702 170000
249 022674
    022674 010246
    022676 010346
    022700 012746 023117
    022704 012746 000003
    022710 010600
    022712 104416
    022714 062706 000010
250 022720 000446
251 022722 020227 160000
252 022726 001012
253 022730
    022730 010346
    022732 012746 023167
    
```

```

.SBTTL CLEAN-UP AND REPORT CODING SECTIONS

; **
; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
; EXECUTED AT THE END OF EACH PASS (OR SUB-PASS).
; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
; --
      BGNCLN
L$CLEAN::
      MOV     CSRADDR,R5           ;POINT TO DEVICE REGISTER
      TST     DUFLG               ;"DROPPED" FLAG IS SET ON...
      BMI     1$                 ;...AND GROSS CONTROLLER FAULT...
                                   ;...DON'T TRY TO XCT CLEANUP CODE.
      MOV     #0,TSSR(R5)         ;DO SOFT INIT
      JSR     PC,WAITF
1$:
2$:
L10034:
      ENDCLN
      TRAP    C$CLEAN

; **
; THE REPORT CODING SECTION CONTAINS THE
; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
; --
      BGNRPT
L$RPT::
      PRINTS #DEVSUM
      MOV     #DEVSUM,-(SP)
      MOV     #1,-(SP)
      MOV     SP,R0
      TRAP    C$PNTS
      ADD     #4,SP
      MOV     R2,-(SP)
      MOV     R3,-(SP)
      MOV     R4,-(SP)
      MOV     #ERTABL,R4         ; GET START OF ERROR TABLE.
      CLR     R3                 ; CLEAR UNIT NUMBER
1$:
      MOV     (R4),R2           ; GET ERROR TABLE ENTRY & TEST IT.
      BEQ     4$                 ; ZERO IF UNIT NOT RUN
      BPL     4$
      BIT     #BIT14,R2         ; WAS UNIT DROPPED?
      BNE     2$                 ; BR IF YES
      BIC     #C7777,R2        ; GET ERROR COUNT FIELD
      PRINTS #DEVONL,R3,R2     ; PRINT
      MOV     R2,-(SP)
      MOV     R3,-(SP)
      MOV     #DEVONL,-(SP)
      MOV     #3,-(SP)
      MOV     SP,R0
      TRAP    C$PNTS
      ADD     #10,SP
      BR      4$
2$:
      CMP     R2,#160000        ; WAS UNIT NON-EXISTENT?
      BNE     3$                 ; BR IF NO
      PRINTS #DEVNXR,R3
      MOV     R3,-(SP)
      MOV     #DEVNXR,-(SP)
    
```

```

022736 012746 000002      MOV      #2,-(SP)
022742 010600      MOV      SP,R0
022744 104416      TRAP     C#PNTS
022746 062706 000006      ADD      #6,SP
254 022752 000431      BR       4#
255 022754 020227 160001      3#:     CMP      R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
256 022760 001012      BNE     30#                ; BR IF NO.
257 022762      PRINTS  #DEVNRD,R3
022762 010346      MOV      R3,-(SP)
022764 012746 023251      MOV      #DEVNRD,-(SP)
022770 012746 000002      MOV      #2,-(SP)
022774 010600      MOV      SP,R0
022776 104416      TRAP     C#PNTS
023000 062706 000006      ADD      #6,SP
258 023004 000414      BR       4#
259 023006 042702 170000      30#:    BIC      #+C7777,R2
260 023012      PRINTS  #DEVDR0,R3,R2
023012 010246      MOV      R2,-(SP)
023014 010346      MOV      R3,-(SP)
023016 012746 023332      MOV      #DEVDR0,-(SP)
023022 012746 000003      MOV      #3,-(SP)
023026 010600      MOV      SP,R0
023030 104416      TRAP     C#PNTS
023032 062706 000010      ADD      #10,SP
261 023036 062704 000002      4#:     ADD      #2,R4
262 023042 005203      INC      R3
263 023044 020427 003374      CMP      R4,#ERTABE
264 023050 103701      BLO     1#
265 023052 012604      MOV      (SP)+,R4
266 023054 012603      MOV      (SP)+,R3
267 023056 012602      MOV      (SP)+,R2
268 023060      ENDRPT      ; UNUSED.
023060      L10035:
023060 104425      TRAP     C#RPT
269
270 023062      045      116      045  DEVSUM: .ASCIZ /#N#ADEVICE STATUS SUMMARY:#N/
271 023117      045      101      040  DEVONL: .ASCIZ /#A UNIT #D3#A ONLINE, ERRORS = #D#N/
272 023167      045      101      040  DEVN XR: .ASCIZ /#A UNIT #D3#A DROPPED, NON-EXISTENT REGISTER#N/
273 023251      045      101      040  DEVNRD: .ASCIZ /#A UNIT #D3#A DROPPED, NOT READY AT STARTUP#N/
274 023332      045      101      040  DEVDR0: .ASCIZ /#A UNIT #D3#A DROPPED, ERRORS = #D#N/
275      .EVEN
276
277 023402      ENDMOD
278

```



```

1          .TITLE  TSV5A - HARDWARE TESTS
2
9
10 023402      BGNMOD  TSV5
    023402      TSV5::
16
24          .SBTTL  TEST  1:  BUS RESET TEST
25
26
27
28          :      THIS TEST VERIFIES THAT THE M7196 MODULE'S DEVICE REGISTERS ARE
29          :      ACCESSIBLE ON THE BUS (SUBTEST 1) AND THEN CHECKS THAT THE
30          :      BUILT-IN INITIALIZATION SELF-TEST MICRODIAGNOSTIC DID NOT FIND
31          :      ANY BASIC PROBLEMS IN THE MODULE.  AREAS OF LOGIC TESTED BY THE
32          :      SELF-TEST SEQUENCE ARE AS FOLLOWS:  ROM AND PIPELINE REGISTER,
33          :      SEQUENCER, INTERNAL BUSES, 2901 MICROPROCESSOR, AND, RAM.  THIS
34          :      TEST INITIALIZES THE CONTROLLER BY ISSUING THE BUS INIT SIGNAL
35          :      VIA A RESET INSTRUCTION, OR BY WRITING INTO THE TSSR REGISTER,
36          :      WAITS A PERIOD OF TIME (TO ALLOW THE CONTROLLER'S INITIALIZATION
37          :      MICRODIAGNOSTIC SEQUENCE TO BE COMPLETED), AND THEN CHECKS THE
38          :      CONTENTS OF THE TSSR REGISTER.  SUCCESSFUL INITIALIZATION IS
39          :      INDICATED BY SUBSYSTEM READY (SSR) AND NEED BUFFER ADDRESS (NBA)
40          :      BITS BEING SET (1) AND ALL OTHER BITS (EXCEPT A17 AND A16 AND
41          :      OFL, WHICH ARE IGNORED FOR THIS TEST) BEING CLEAR (0).  IF THE
42          :      CONTENTS OF TSSR ARE NOT AS EXPECTED, AN ERROR REPORT IS ISSUED
43          :      LISTING THE EXPECTED DATA, ACTUAL DATA, AND THE DISCREPANCIES.
44          :      THE ERROR REPORT ANALYZES THE TSSR CONTENTS AND DISCERNs AND
45          :      REPORTS ONE OF THREE POSSIBILITIES:
46
47
48          :
49          :      1.  TSSR CONTENTS ARE AMBIGUOUS (ANY OF BITS 11-14 ARE SET,
50          :      OR STATES OF SSR AND SC BITS DO NOT CORRESPOND TO THE
51          :      APPARENT ERROR CODE IN BITS 0-5):  INDICATES THAT THE
52          :      TSSR CONTENT CANNOT BE TRUSTED.  INDICATES A
53          :      CATASTROPHIC CONTROLLER MALFUNCTION.  THIS IS A FATAL
54          :      ERROR (EXECUTION IS ABORTED).  FIELD ACTION WOULD BE TO
55          :      REPLACE THE M7196.  IF THE M7196 ITSELF IS BEING
56          :      DEBUGGED, THE PROGRAM SHOULD BE RESTARTED WITH LOOP ON
57          :      ERROR ENABLED IN ORDER TO PROBE FOR THE PROBLEM.
58
59          :
60          :      2.  SSR = 0, SC = 0 AND THE ERROR CODE IN BITS 0-5 IS IN
61          :      THE RANGE 17-13:  THIS IS A FATAL ERROR.  THE ERROR
62          :      CODE IS DECODED AND THE APPROPRIATE DESCRIPTION GIVEN.
63          :      INDICATES THAT A SERIOUS PROBLEM EXISTS.
64
65          :
66          :
67          :
68          :
69          :
70          :
71          :
72          :
73          :
74          :
75          :
76          :
77          :
78          :
79          :
80          :
81          :
82          :
83          :
84          :
85          :
86          :
87          :
88          :
89          :
90          :
91          :
92          :
93          :
94          :
95          :
96          :
97          :
98          :
99          :
100         :
101         :
102         :
103         :
104         :
105         :
106         :
107         :
108         :
109         :
110         :
111         :
112         :
113         :
114         :
115         :
116         :
117         :
118         :
119         :
120         :
121         :
122         :
123         :
124         :
125         :
126         :
127         :
128         :
129         :
130         :
131         :
132         :
133         :
134         :
135         :
136         :
137         :
138         :
139         :
140         :
141         :
142         :
143         :
144         :
145         :
146         :
147         :
148         :
149         :
150         :
151         :
152         :
153         :
154         :
155         :
156         :
157         :
158         :
159         :
160         :
161         :
162         :
163         :
164         :
165         :
166         :
167         :
168         :
169         :
170         :
171         :
172         :
173         :
174         :
175         :
176         :
177         :
178         :
179         :
180         :
181         :
182         :
183         :
184         :
185         :
186         :
187         :
188         :
189         :
190         :
191         :
192         :
193         :
194         :
195         :
196         :
197         :
198         :
199         :
200         :
201         :
202         :
203         :
204         :
205         :
206         :
207         :
208         :
209         :
210         :
211         :
212         :
213         :
214         :
215         :
216         :
217         :
218         :
219         :
220         :
221         :
222         :
223         :
224         :
225         :
226         :
227         :
228         :
229         :
230         :
231         :
232         :
233         :
234         :
235         :
236         :
237         :
238         :
239         :
240         :
241         :
242         :
243         :
244         :
245         :
246         :
247         :
248         :
249         :
250         :
251         :
252         :
253         :
254         :
255         :
256         :
257         :
258         :
259         :
260         :
261         :
262         :
263         :
264         :
265         :
266         :
267         :
268         :
269         :
270         :
271         :
272         :
273         :
274         :
275         :
276         :
277         :
278         :
279         :
280         :
281         :
282         :
283         :
284         :
285         :
286         :
287         :
288         :
289         :
290         :
291         :
292         :
293         :
294         :
295         :
296         :
297         :
298         :
299         :
300         :
301         :
302         :
303         :
304         :
305         :
306         :
307         :
308         :
309         :
310         :
311         :
312         :
313         :
314         :
315         :
316         :
317         :
318         :
319         :
320         :
321         :
322         :
323         :
324         :
325         :
326         :
327         :
328         :
329         :
330         :
331         :
332         :
333         :
334         :
335         :
336         :
337         :
338         :
339         :
340         :
341         :
342         :
343         :
344         :
345         :
346         :
347         :
348         :
349         :
350         :
351         :
352         :
353         :
354         :
355         :
356         :
357         :
358         :
359         :
360         :
361         :
362         :
363         :
364         :
365         :
366         :
367         :
368         :
369         :
370         :
371         :
372         :
373         :
374         :
375         :
376         :
377         :
378         :
379         :
380         :
381         :
382         :
383         :
384         :
385         :
386         :
387         :
388         :
389         :
390         :
391         :
392         :
393         :
394         :
395         :
396         :
397         :
398         :
399         :
400         :
401         :
402         :
403         :
404         :
405         :
406         :
407         :
408         :
409         :
410         :
411         :
412         :
413         :
414         :
415         :
416         :
417         :
418         :
419         :
420         :
421         :
422         :
423         :
424         :
425         :
426         :
427         :
428         :
429         :
430         :
431         :
432         :
433         :
434         :
435         :
436         :
437         :
438         :
439         :
440         :
441         :
442         :
443         :
444         :
445         :
446         :
447         :
448         :
449         :
450         :
451         :
452         :
453         :
454         :
455         :
456         :
457         :
458         :
459         :
460         :
461         :
462         :
463         :
464         :
465         :
466         :
467         :
468         :
469         :
470         :
471         :
472         :
473         :
474         :
475         :
476         :
477         :
478         :
479         :
480         :
481         :
482         :
483         :
484         :
485         :
486         :
487         :
488         :
489         :
490         :
491         :
492         :
493         :
494         :
495         :
496         :
497         :
498         :
499         :
500         :
501         :
502         :
503         :
504         :
505         :
506         :
507         :
508         :
509         :
510         :
511         :
512         :
513         :
514         :
515         :
516         :
517         :
518         :
519         :
520         :
521         :
522         :
523         :
524         :
525         :
526         :
527         :
528         :
529         :
530         :
531         :
532         :
533         :
534         :
535         :
536         :
537         :
538         :
539         :
540         :
541         :
542         :
543         :
544         :
545         :
546         :
547         :
548         :
549         :
550         :
551         :
552         :
553         :
554         :
555         :
556         :
557         :
558         :
559         :
560         :
561         :
562         :
563         :
564         :
565         :
566         :
567         :
568         :
569         :
570         :
571         :
572         :
573         :
574         :
575         :
576         :
577         :
578         :
579         :
580         :
581         :
582         :
583         :
584         :
585         :
586         :
587         :
588         :
589         :
590         :
591         :
592         :
593         :
594         :
595         :
596         :
597         :
598         :
599         :
600         :
601         :
602         :
603         :
604         :
605         :
606         :
607         :
608         :
609         :
610         :
611         :
612         :
613         :
614         :
615         :
616         :
617         :
618         :
619         :
620         :
621         :
622         :
623         :
624         :
625         :
626         :
627         :
628         :
629         :
630         :
631         :
632         :
633         :
634         :
635         :
636         :
637         :
638         :
639         :
640         :
641         :
642         :
643         :
644         :
645         :
646         :
647         :
648         :
649         :
650         :
651         :
652         :
653         :
654         :
655         :
656         :
657         :
658         :
659         :
660         :
661         :
662         :
663         :
664         :
665         :
666         :
667         :
668         :
669         :
670         :
671         :
672         :
673         :
674         :
675         :
676         :
677         :
678         :
679         :
680         :
681         :
682         :
683         :
684         :
685         :
686         :
687         :
688         :
689         :
690         :
691         :
692         :
693         :
694         :
695         :
696         :
697         :
698         :
699         :
700         :
701         :
702         :
703         :
704         :
705         :
706         :
707         :
708         :
709         :
710         :
711         :
712         :
713         :
714         :
715         :
716         :
717         :
718         :
719         :
720         :
721         :
722         :
723         :
724         :
725         :
726         :
727         :
728         :
729         :
730         :
731         :
732         :
733         :
734         :
735         :
736         :
737         :
738         :
739         :
740         :
741         :
742         :
743         :
744         :
745         :
746         :
747         :
748         :
749         :
750         :
751         :
752         :
753         :
754         :
755         :
756         :
757         :
758         :
759         :
760         :
761         :
762         :
763         :
764         :
765         :
766         :
767         :
768         :
769         :
770         :
771         :
772         :
773         :
774         :
775         :
776         :
777         :
778         :
779         :
780         :
781         :
782         :
783         :
784         :
785         :
786         :
787         :
788         :
789         :
790         :
791         :
792         :
793         :
794         :
795         :
796         :
797         :
798         :
799         :
800         :
801         :
802         :
803         :
804         :
805         :
806         :
807         :
808         :
809         :
810         :
811         :
812         :
813         :
814         :
815         :
816         :
817         :
818         :
819         :
820         :
821         :
822         :
823         :
824         :
825         :
826         :
827         :
828         :
829         :
830         :
831         :
832         :
833         :
834         :
835         :
836         :
837         :
838         :
839         :
840         :
841         :
842         :
843         :
844         :
845         :
846         :
847         :
848         :
849         :
850         :
851         :
852         :
853         :
854         :
855         :
856         :
857         :
858         :
859         :
860         :
861         :
862         :
863         :
864         :
865         :
866         :
867         :
868         :
869         :
870         :
871         :
872         :
873         :
874         :
875         :
876         :
877         :
878         :
879         :
880         :
881         :
882         :
883         :
884         :
885         :
886         :
887         :
888         :
889         :
890         :
891         :
892         :
893         :
894         :
895         :
896         :
897         :
898         :
899         :
900         :
901         :
902         :
903         :
904         :
905         :
906         :
907         :
908         :
909         :
910         :
911         :
912         :
913         :
914         :
915         :
916         :
917         :
918         :
919         :
920         :
921         :
922         :
923         :
924         :
925         :
926         :
927         :
928         :
929         :
930         :
931         :
932         :
933         :
934         :
935         :
936         :
937         :
938         :
939         :
940         :
941         :
942         :
943         :
944         :
945         :
946         :
947         :
948         :
949         :
950         :
951         :
952         :
953         :
954         :
955         :
956         :
957         :
958         :
959         :
960         :
961         :
962         :
963         :
964         :
965         :
966         :
967         :
968         :
969         :
970         :
971         :
972         :
973         :
974         :
975         :
976         :
977         :
978         :
979         :
980         :
981         :
982         :
983         :
984         :
985         :
986         :
987         :
988         :
989         :
990         :
991         :
992         :
993         :
994         :
995         :
996         :
997         :
998         :
999         :
1000        :
1001        :
1002        :
1003        :
1004        :
1005        :
1006        :
1007        :
1008        :
1009        :
1010        :
1011        :
1012        :
1013        :
1014        :
1015        :
1016        :
1017        :
1018        :
1019        :
1020        :
1021        :
1022        :
1023        :
1024        :
1025        :
1026        :
1027        :
1028        :
1029        :
1030        :
1031        :
1032        :
1033        :
1034        :
1035        :
1036        :
1037        :
1038        :
1039        :
1040        :
1041        :
1042        :
1043        :
1044        :
1045        :
1046        :
1047        :
1048        :
1049        :
1050        :
1051        :
1052        :
1053        :
1054        :
1055        :
1056        :
1057        :
1058        :
1059        :
1060        :
1061        :
1062        :
1063        :
1064        :
1065        :
1066        :
1067        :
1068        :
1069        :
1070        :
1071        :
1072        :
1073        :
1074        :
1075        :
1076        :
1077        :
1078        :
1079        :
1080        :
1081        :
1082        :
1083        :
1084        :
1085        :
1086        :
1087        :
1088        :
1089        :
1090        :
1091        :
1092        :
1093        :
1094        :
1095        :
1096        :
1097        :
1098        :
1099        :
1100        :
1101        :
1102        :
1103        :
1104        :
1105        :
1106        :
1107        :
1108        :
1109        :
1110        :
1111        :
1112        :
1113        :
1114        :
1115        :
1116        :
1117        :
1118        :
1119        :
1120        :
1121        :
1122        :
1123        :
1124        :
1125        :
1126        :
1127        :
1128        :
1129        :
1130        :
1131        :
1132        :
1133        :
1134        :
1135        :
1136        :
1137        :
1138        :
1139        :
1140        :
1141        :
1142        :
1143        :
1144        :
1145        :
1146        :
1147        :
1148        :
1149        :
1150        :
1151        :
1152        :
1153        :
1154        :
1155        :
1156        :
1157        :
1158        :
1159        :
1160        :
1161        :
1162        :
1163        :
1164        :
1165        :
1166        :
1167        :
1168        :
1169        :
1170        :
1171        :
1172        :
1173        :
1174        :
1175        :
1176        :
1177        :
1178        :
1179        :
1180        :
1181        :
1182        :
1183        :
1184        :
1185        :
1186        :
1187        :
1188        :
1189        :
1190        :
1191        :
1192        :
1193        :
1194        :
1195        :
1196        :
1197        :
1198        :
1199        :
1200        :
1201        :
1202        :
1203        :
1204        :
1205        :
1206        :
1207        :
1208        :
1209        :
1210        :
1211        :
1212        :
1213        :
1214        :
1215        :
1216        :
1217        :
1218        :
1219        :
1220        :
1221        :
1222        :
1223        :
1224        :
1225        :
1226        :
1227        :
1228        :
1229        :
1230        :
1231        :
1232        :
1233        :
1234        :
1235        :
1236        :
1237        :
1238        :
1239        :
1240        :
1241        :
1242        :
1243        :
1244        :
1245        :
1246        :
1247        :
1248        :
1249        :
1250        :
1251        :
1252        :
1253        :
1254        :
1255        :
1256        :
1257        :
1258        :
1259        :
1260        :
1261        :
1262        :
1263        :
1264        :
1265        :
1266        :
1267        :
1268        :
1269        :
1270        :
1271        :
1272        :
1273        :
1274        :
1275        :
1276        :
1277        :
1278        :
1279        :
1280        :
1281        :
1282        :
1283        :
1284        :
1285        :
1286        :
1287        :
1288        :
1289        :
1290        :
1291        :
1292        :
1293        :
1294        :
1295        :
1296        :
1297        :
1298        :
1299        :
1300        :
1301        :
1302        :
1303        :
1304        :
1305        :
1306        :
1307        :
1308        :
1309        :
1310        :
1311        :
1312        :
1313        :
1314        :
1315        :
1316        :
1317        :
1318        :
1319        :
1320        :
1321        :
1322        :
1323        :
1324        :
1325        :
1326        :
1327        :
1328        :
1329        :
1330        :
1331        :
1332        :
1333        :
1334        :
1335        :
1336        :
1337        :
1338        :
1339        :
1340        :
1341        :
1342        :
1343        :
1344        :
1345        :
1346        :
1347        :
1348        :
1349        :
1350        :
1351        :
1352        :
1353        :
1354        :
1355        :
1356        :
1357        :
1358        :
1359        :
1360        :
1361        :
1362        :
1363        :
1364        :
1365        :
1366        :
1367        :
1368        :
1369        :
1370        :
1371        :
1372        :
1373        :
1374        :
1375        :
1376        :
1377        :
1378        :
1379        :
1380        :
1381        :
1382        :
1383        :
1384        :
1385        :
1386        :
1387        :
1388        :
1389        :
1390        :
1391        :
1392        :
1393        :
1394        :
1395        :
1396        :
1397        :
1398        :
1399        :
1400        :
1401        :
1402        :
1403        :
1404        :
1405        :
1406        :
1407        :
1408        :
1409        :
1410        :
1411        :
1412        :
1413        :
1414        :
1415        :
1416        :
1417        :
1418        :
1419        :
1420        :
1421        :
1422        :
1423        :
1424        :
1425        :
1426        :
1427        :
1428        :
1429        :
1430        :
1431        :
1432        :
1433        :
1434        :
1435        :
1436        :
1437        :
1438        :
1439        :
1440        :
1441        :
1442        :
1443        :
1444        :
1445        :
1446        :
1447        :
1448        :
1449        :
1450        :
1451        :
1452        :
1453        :
1454        :
1455        :
1456        :
1457        :
1458        :
1459        :
1460        :
1461        :
1462        :
1463        :
1464        :
1465        :
1466        :
1467        :
1468        :
1469        :
1470        :
1471        :
1472        :
1473        :
1474        :
1475        :
1476        :
1477        :
1478        :
1479        :
1480        :
1481        :
1482        :
1483        :
1484        :
1485        :
1486        :
1487        :
1488        :
1489        :
1490        :
1491        :
1492        :
1493        :
1494        :
1495        :
1496        :
1497        :
1498        :
1499        :
1500        :
1501        :
1502        :
1503        :
1504        :
1505        :
1506        :
1507        :
1508        :
1509        :
1510        :
1511        :
1512        :
1513        :
1514        :
1515        :
1516        :
1517        :
1518        :
1519        :
1520        :
1521        :
1522        :
1523        :
1524        :
1525        :
1526        :
1527        :
1528        :
1529        :
1530        :
1531        :
1532        :
1533        :
1534        :
1535        :
1536        :
1537        :
1538        :
1539        :
1540        :
1541        :
1542        :
1543        :
1544        :
1545        :
1546        :
1547        :
1548        :
1549        :
1550        :
1551        :
1552        :
1553        :
1554        :
1555        :
1556        :
1557        :
1558        :
1559        :
1560        :
1561        :
1562        :
1563        :
1564        :
1565        :
1566        :
1567        :
1568        :
1569        :
1570        :
1571        :
1572        :
1573        :
1574        :
1575        :
1576        :
1577        :
1578        :
1579        :
1580        :
1581        :
1582        :
1583        :
1584        :
1585        :
1586        :
1587        :
1588        :
1589        :
1590        :
1591        :
1592        :
1593        :
1594        :
1595        :
1596        :
1597        :
1598        :
1599        :
1600        :
1601        :
1602        :
1603        :
1604        :
1605        :
1606        :
1607        :
1608        :
1609        :
1610        :
1611        :
1612        :
1613        :
1614        :
1615        :
1616        :
1617        :
1618        :
1619        :
1620        :
1621        :
1622        :
1623        :
1624        :
1625        :
1626        :
1627        :
1628        :
1629        :
1630        :
1631        :
1632        :
1633        :
1634        :
1635        :
1636        :
1637        :
1638        :
1639        :
1640        :
1641        :
1642        :
1643        :
1644        :
1645        :
1646        :
1647        :
1648        :
1649        :
1650        :
1651        :
1652        :
1653        :
1654        :
1655        :
1656        :
1657        :
1658        :
1659        :
1660        :
1661        :
1662        :
1663        :
1664        :
1665        :
1666        :
1667        :
1668        :
1669        :
1670        :
1671        :
1672        :
1673        :
1674        :
1675        :
1676        :
1677        :
1678        :
1679        :
1680        :
1681        :
1682        :
1683        :
1684        :
1685        :
1686        :
1687        :
1688        :
1689        :
1690        :
1691        :
1692        :
1693        :
1694        :
1695        :
1696        :
1697        :
1698        :
1699        :
1700        :
1701        :
1702        :
1703        :
1704        :
1705        :
1706        :
1707        :
1708        :
1709        :
1710        :
1711        :
1712        :
1713        :
1714        :
1715        :
1716        :
1717        :
1718        :
1719        :
1720        :
1721        :
1722        :
1723        :
1724        :
1725        :
1726        :
1727        :
1728        :
1729        :
1730        :
1731        :
1732        :
1733        :
1734        :
1735        :
1736        :
1737        :
1738        :
1739        :
1740        :
1741        :
1742        :
1743        :
1744        :
1745        :
1746        :
1747        :
1748        :
1749        :
1750        :
1751        :
1752        :
1753        :
1754        :
1755        :
1756        :
1757        :
1758        :
1759        :
1760        :
1761        :
1762        :
1763        :
1764        :
1765        :
1766        :
1767        :
1768        :
1769        :
1770        :
1771        :
1772        :
1773        :
1774        :
1775        :
1776        :
1777        :
1778        :
1779        :
1780        :
1781        :
1782        :
1783        :
1784        :
1785        :
1786        :
1787        :
1788        :
1789        :
1790        :
1791        :
1792        :
1793        :
1794        :
1795        :
1796        :
1797        :
1798        :
1799        :
1800        :
1801        :
1802        :
1803        :
1804        :
1805        :
1806        :
1807        :
1808        :
1809        :
1810        :
1811        :
1812        :
1813        :
1814        :
1815        :
1816        :
1817        :
1818        :
1819        :
1820        :
1821        :
1822        :
1823        :
1824        :
1
```

TSVSA - HARDWARE TESTS MACRO M1113 14-JUN-84 15:15
 TEST 1: BUS RESET TEST

SEQ 0105

```

76
77 023424          BRESET          ;ISSUE A BUS RESET
    023424 104433          TRAP      C0RESET
78 023426 004737 016250      JSR      PC, WAITF          ;WAIT FOR READY
79 023432 016501 000002      MOV      TSSR(R5), R1      ;GET THE CONTENTS OF TSSR
80 023436 010102          MOV      R1, R2          ;CONTENTS OF TSSR
81 023440 042702 176277      BIC      @+C<HIADDR!OFL>, R2 ;THESE BITS MAY BE SET
82 023444 052702 002200      BIS      @SSR!NBA, R2      ;READY AND NEW DATA SHOULD BE SET
83 023450 020102          CMP      R1, R2          ;COMPARE EXPECTED TO RECEIVED
84 023452 001405          BEQ      101          ;BRANCH IF COMPARE
88 023454          ERRDF      ERRNO, SFHERR, SFFMSG ;REPORT A FATAL ERROR
    023454 104455          TRAP      C1ERDF
    023456 000145          .WORD    101
    023460 003703          .WORD    SFHERR
    023462 012102          .WORD    SFFMSG
89 023464 005203          INC      R3          ;SET THE FATAL ERROR FLAG
90 023466          101:          ;
91 023466          ENDSUB      ;////////// END SUBTEST //////////
    023466          L10037:      TRAP      C0ESUB
    023466 104403          ;
92
93 023470 005703          TST      R3          ;DID WE HAVE FATAL ERROR ?
94 023472 001402          BEQ      201          ;BRANCH IF NOT
95 023474 004737 017202      JSR      PC, CKDROP      ;GO DROP THIS UNIT, IF ALLOWED
96 023500 005003          201:      CLR      R3          ;RESET FATAL ERROR FLAG
97
98
99 023502          BGNSUB      ;////////// BEGIN SUBTEST //////////
    023502          T1.2:          TRAP      C0BSUB
    023502 104402          ;
100
101 023504 005065 000002      CLR      TSSR(R5)      ;WRITE TO ISSUE A SOFT RESET
102 023510 004737 016250      JSR      PC, WAITF      ;WAIT FOR READY TO SET
103 023514 016501 000002      MOV      TSSR(R5), R1  ;GET REGISTER TSSR DATA
104 023520 010102          MOV      R1, R2          ;CONTENTS OF TSSR
105 023522 042702 176277      BIC      @+C<HIADDR!OFL>, R2 ;THESE BITS MAY BE SET
106 023526 052702 002200      BIS      @SSR!NBA, R2  ;READY AND NEW DATA SHOULD BE SET
107 023532 020102          CMP      R1, R2          ;COMPARE EXPECTED TO RECEIVED
108 023534 001405          BEQ      101          ;BRANCH IF COMPARE
112 023536          ERRDF      ERRNO, SFIERR, SFFMSG ;REPORT A FATAL ERROR
    023536 104455          TRAP      C1ERDF
    023540 000146          .WORD    102
    023542 003650          .WORD    SFIERR
    023544 012102          .WORD    SFFMSG
113 023546 005203          INC      R3          ;SET THE ERROR FLAG
114 023550          101:          ;
115 023550          ENDSUB      ;////////// END SUBTEST //////////
    023550          L10040:      TRAP      C0ESUB
    023550 104403          ;
116
117
118 023552 005703          TST      R3          ;FATAL ERROR DETECTED ?
119 023554 001402          BEQ      201          ;BRANCH IF NOT
120 023556 004737 017202      JSR      PC, CKDROP      ;SEE IF TIME TO DROP UNIT
121 023562 004737 016456          201:      JSR      PC, TSTLOOP      ;SHOULD WE DO ITERATIONS ?
122 023566 103002          BCC      401          ;BRANCH IF NOT
123 023570 000137 023420      JMP      T1LOOP        ;LOOP UNTIL COUNT EXPIRED

```



```

180 023650 001412          BEQ      10$          ;BRANCH IF NOT
181 023652 005003          CLR      R3           ;DON'T NEED INIT NEXT TIME THRU
182 023654 004737 015774   JSR      PC,SOFINIT   ;DO SOFT INIT OF CONTROLLER
183 023660 103406          BCS      10$          ;BR IF SOFT INIT = OK
187 023662 010001          MOV      R0,R1        ;SAVE CONTENTS OF TSSR
188 023664          ERRDF   ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
      023664 104455          TRAP    C$ERDF
      023666 000311          .WORD  201
      023670 003650          .WORD  SFIERR
      023672 012034          .WORD  SFIMSG
189 023674 005203          INC      R3           ;FORCE SOFT INIT ON NEXT PASS
190 023676 005037 002220   10$:   CLR      FATFLG    ;CLEAR FATAL ERROR FLAG
191
192 023702          BGNSEG
      023702 104404          TRAP    C$BSEG
193
194 023704 110465 000001   MOVB    R4,TSDBH(R5) ;SET MAINT MODE + WRITE DATA
195 023710 004737 016250   JSR      PC,WAITF     ;WAIT FOR SSR TO SET
196 023714 103411          BCS      15$          ;BR IF CARRY SET (GOOD RETURN)
197 023716 010001          MOV      R0,R1        ;SAVE CONTENTS OF TSSR
198 023720 010402          MOV      R4,R2        ;DATA THAT WAS WRITTEN
202 023722          ERRDF   ERRNO,T2SSR,EXPREC ;DEVICE FATAL SSR FAILED TO SET
      023722 104455          TRAP    C$ERDF
      023724 000312          .WORD  202
      023726 024216          .WORD  T2SSR
      023730 015474          .WORD  EXPREC
203 023732 005203          INC      R3           ;FORCE SOFT INIT ON NEXT PASS
204 023734 005237 002220   15$:   INC      FATFLG    ;SET FATAL ERROR FLAG
205 023740          CKLOOP
      023740 104406          TRAP    C$CLP1
206 023742 005737 002220   TST     FATFLG       ;WAS FATAL ERROR RECEIVED ?
207 023746 001402          BEQ      20$          ;BRANCH IF NOT
208 023750 004737 017202   JSR      PC,CKDROP    ;SEE IF TIME TO DROP UNIT
209 023754 010402          MOV      R4,R2        ;DATA PATTERN WRITTEN
210 023756 042702 177774   20$:   BIC     #1C<BIT0!BIT1>,R2 ;CLEAR ALL BUT LOW 2 BITS
211 023762 000302          SWAB    R2           ;BITS 8 AND 9 HAVE LOW DATA BITS
212 023764 052702 002200   BIS     #SSR!NBA,R2   ;THESE BITS MUST BE SET ALSO
213 023770 016501 000002   MOV      TSSR(R5),R1 ;GET THE CONTENTS OF TSSR
214 023774 032701 000100   BIT     #OFL,R1       ;IS OFF-LINE BIT SET ?
215 024000 001402          BEQ      25$          ;BRANCH IF NOT OFF-LINE
216 024002 052702 000100   BIS     #OFL,R2       ;SET OFF-LINE IN EXPECTED DATA
217 024006 020201          CMP     R2,R1         ;DOES EXPECTED MATCH RECEIVED ?
218 024010 001405          BEQ      30$          ;OKAY IF MATCH
222 024012          ERRHRD  ERRNO,T2TSSR,EXPREC ;TSSR WASN'T CORRECT
      024012 104456          TRAP    C$ERHRD
      024014 000313          .WORD  203
      024016 024151          .WORD  T2TSSR
      024020 015474          .WORD  EXPREC
223 024022 005203          INC      R3           ;FORCE SOFT INIT ON NEXT PASS
224 024024          CKLOOP
      024024 104406          TRAP    C$CLP1
225 024026 016501 000000   MOV      TSBA(R5),R1 ;GET TSBA REGISTER CONTENTS
226 024032 005002          CLR      R2           ;
227 024034 150402          BISB   R4,R2         ;DATA PATTERN WRITTEN
228 024036 000302          SWAB   R2           ;MOVE INTO TOP BYTE
229 024040 150402          BISB   R4,R2         ;BOTH HALVES SHOULD BE SAME
230 024042 020102          CMP     R1,R2        ;COMPARE EXPECTED TO RECEIVED

```



```

385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421 025012          BGNTST
    025012
422
423 025012          BGNSUB
    025012          ;//////////////// BEGIN SUBTEST //////////////////
    025012 104402    T4.1:          TRAP      C#BSUB
424
429 025014 012700 026322      MOV      #TST4ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
430 025020 004737 016510      JSR      PC,TSTSETUP   ;DO INITIAL TEST SETUP
431 025024 012737 000005 002214  MOV      #5,LOOPCNT    ;PERFORM 5 ITERATIONS
432 025032          T4LOOP:
433 025032 004737 015774      JSR      PC,SOFINIT    ;DO INITIALIZE ON CONTROLLER
434 025036 103405          BCS      20$,          ;BR IF INIT WAS OK
438 025040 010001          MOV      R0,R1         ;CONTENTS OF TSSR REGISTER
439 025042          ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
    025042 104455          TRAP      C#ERDF
    025044 000621          .WORD    401
    025046 003650          .WORD    SFIERR
    025050 012034          .WORD    SFIMSG
440 025052 005004          20$: CLR      R4         ;SET RAM ADDRESS AT ZERO
441 025054 004737 016336      JSP     PC,CHKTSSR    ;WAIT FOR READY, NON-AMBIGUOUS

```

TEST 4 , SUBTEST 1: -

THIS SUBTEST VERIFIES EACH RAM LOCATION BY FIRST PLACING THE M7196 INTO MAINTENANCE MODE BY WRITING INTO THE LOW BYTE OF TSDB AND THEN PERFORMING THE FOLLOWING SEQUENCE FOR EACH ADDRESS 0-7777 (OCTAL):

1. THE ADDRESS TO BE TESTED IS LOADED INTO THE TSDB (VIA A WORD WRITE).
2. THE ADDRESSED RAM LOCATION IS WRITTEN, THEN READ INTO THE LOW BYTE OF TSBA, BY WRITING A DATA BYTE INTO THE LOW BYTE OF TSDB.
3. THE LOW BYTE OF TSBA IS CHECKED TO SEE IF IT CONTAINS THE DATA PATTERN ORIGINALLY WRITTEN; A DISCREPANCY IS REPORTED AS AN ERROR.
4. THE ADDRESS OF THE LOCATION BEING TESTED IS AGAIN WRITTEN INTO TSDB (WORD WRITE), TO CAUSE THE LOCATION UNDER TEST TO AGAIN BE READ INTO THE LOW BYTE OF TSBA. THE LOW BYTE OF TSBA IS AGAIN CHECKED AND DISCREPANCIES REPORTED.
5. THE HIGH BYTE OF TSBA IS CHECKED; IT SHOULD CONTAIN THE SUM OF THE HIGH AND LOW BYTES LAST WRITTEN INTO TSDB AS A WORD. A DISCREPANCY IS REPORTED AS A 2901 PROBLEM.
6. THE CONTENT OF TSSR IS CHECKED; SETTING OF THE SC BIT IS IGNORED. OTHER DISCREPANCIES IN TSSR ARE REPORTED.


```

537
538 025372 005304          35$: DEC    R4          ;SET BACK TO 7777
539 025374 005002          CLR    R2          ;SET TO ALL ZEROS
540 025376 004737 016336  40$: JSR    PC,CHKTSSR ;WAIT FOR READY (SSR) TO SET
541 025402 010465 000000  MOV    R4,TSDB(R5) ;LOAD UP THE ADDRESS FOR RAM
542 025406 004737 016336  JSR    PC,CHKTSSR ;WAIT FOR READY (SSR) TO SET
543 025412 016501 000000  MOV    TSBA(R5),R1 ;READ THE RAM CONTENTS BACK
544 025416 005002          CLR    R2          ;LOOKING FOR 000000 (EXPECTED)
545 025420 120102          CMPB   R1,R2       ;BOTH SHOULD BE 00000000 BINARY
546 025422 001404          BEQ    43$        ;BR, IF DATA IS GOOD
550 025424          ERRHRD ERRNO,TSBAM3,EXPREC ;CHARACTERISTICS DATA NOT CORRECT
      025424 104456          TRAP    C$ERHRD
      025426 000627          .WORD   407
      025430 026242          .WORD   TSBAM3
      025432 015474          .WORD   EXPREC
551 025434 012702 000377  43$: MOV    #000377,R2 ;SET ALL ONES WORD
552 025440 010465 000000  MOV    R4,TSDB(R5) ;LOAD UP RAM ADDRESS POINTER
553 025444 004737 016336  JSR    PC,CHKTSSR ;WAIT FOR READY, NON-AMBIGUOUS
554 025450 110265 000000  MOVB   R2,TSDB(R5) ;WRITE DATA INTO RAM
555 025454 004737 016336  JSR    PC,CHKTSSR ;WAIT FOR READY, NON-AMBIGUOUS
556 025460 016501 000000  MOV    TSBA(R5),R1 ;READ RAM CONTENTS BACK
557 025464 120102          CMPB   R1,R2       ;CHECK WITH DATA WRITTEN
558 025466 001404          BEQ    45$        ;BR IF OK, DATA IN = DATA OUT
562 025470          ERRHRD ERRNO,TSBAM2,EXPREC ;WRITTEN DATA NOT = TO READ
      025470 104456          TRAP    C$ERHRD
      025472 000630          .WORD   408
      025474 026150          .WORD   TSBAM2
      025476 015474          .WORD   EXPREC
563 025500          45$: CKLOOP ;SCOPE LOOP
      025500 104406          TRAP    C$CLP1
564 025502 004737 016336  JSR    PC,CHKTSSR ;WAIT FOR READY, NON-AMBIGUOUS
565 025506 010465 000000  MOV    R4,TSDB(R5) ;WORD WRITE TO SET UP ADDRESS
566 025512 004737 016336  JSR    PC,CHKTSSR ;WAIT FOR READY, NON-AMBIGUOUS
567 025516 116501 000001  MOVB   TSBAH(R5),R1 ;HIGH BYTE READ OF TSBA
568 025522 010403          MOV    R4,R3      ;DATA PATTERN WRITTEN
569 025524 000303          SWAB   R3         ;HIGH TO LOW
570 025526 060403          ADD    R4,R3      ;TOTAL OF BYTES IN LOW BYTE
571 025530 120103          CMPB   R1,R3      ;SUM OF BYTES WRITTEN TO TSDB = TSBAH
572 025532 001404          BEQ    50$        ;BR IF OK, THEY SHOULD BE
576 025534          ERRHRD ERRNO,M2901,EXPREC ;2901 PROBLEM ADDER
      025534 104456          TRAP    C$ERHRD
      025536 000631          .WORD   409
      025540 026070          .WORD   M2901
      025542 015474          .WORD   EXPREC
577 025544          50$: CKLOOP ;SCOPE LOOP
      025544 104406          TRAP    C$CLP1
578 025546 005304          DEC    R4         ;DROP RAM ADDRESS POINTER
579 025550 002312          BGE    40$        ;NOT AT LOC. ZERO YET
580
581 025552          ENDSUB ;////////////////////// END SUBTEST ////////////////////////
      025552          L10045:
      025552 104403          TRAP    C$ESUB
582
583
584 025554          BGNSUB ;////////////////////// BEGIN SUBTEST ////////////////////////
      025554          T4.3:
      025554 104402          TRAP    C$BSUB

```


026344 104401

TRAP C#ETST

682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719

.SBTTL TEST 5: SECOND INITIALIZATION TEST

```

: THIS TEST VERIFIES THE SAME ELEMENTS AS DID INITIALIZATION TEST
: #1 AND ALSO CHECKS THAT CERTAIN PARTS OF RAM IS CLEARED TO ZERO
: AND THAT 2901 REGISTERS 10 AND 11 ARE ALSO CLEARED TO ZERO.
: THIS IS A CONFIDENCE CHECK OF A PART OF THE SELF-TEST SEQUENCE
: (I.E., THAT IT IS REALLY BEING EXECUTED). FOR EACH OF TWO
: SUBTESTS (ONE FOR INITIALIZING VIA A BUS INIT, THE OTHER FOR
: INITIALIZING BY WRITING INTO THE TSSR), THE FOLLOWING SEQUENCE
: IS PERFORMED:
    
```

1. EACH RAM LOCATION AND 2901 REGISTERS 10 AND 11 ARE SET TO ALL 1'S BY USING WRITES INTO THE TSDB REGISTER (LOW BYTE AND MAINTENANCE MODE WORD WRITES).
2. THE CONTROLLER IS INITIALIZED AND THE VARIOUS CHECKS ON THE TSSR DESCRIBED IN INITIALIZATION TEST #1 ARE PERFORMED.
3. #1'S (377 OCTAL) ARE WRITTEN INTO THE LOW BYTE OF TSDB, WHICH SHOULD CAUSE RAM LOCATION 0 TO BE WRITTEN TO ALL 1'S SINCE 2901 REGISTERS 10 AND 11, SPECIFYING THE RAM ADDRESS, SHOULD BE 0. RAM LOCATION 0 IS VERIFIED BY WRITING A WORD OF ZEROS INTO THE TSDB. THE RESULTING LOW BYTE OF TSBA SHOULD CONTAIN ALL 1'S.
4. THE ENTIRE RAM IS SCANNED. LOCATION 0 SHOULD CONTAIN ALL 1'S AND THE REMAINING LOCATIONS, EXCEPT FOR THE MESSAGE BUFFER IMAGE AREA, SHOULD CONTAIN 0. DISCREPANCIES ARE REPORTED. AN ERROR AT THIS POINT IS MOST LIKELY DUE TO A ROM, PIPELINE OR SEQUENCER PROBLEM OR A TIMING PROBLEM.

```

720 026346          BGNTST
      026346
725 026346 012700 027320      MOV    #TST5ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
726 026352 004737 016510      JSR    PC,TSTSETUP    ;DO INITIAL TEST SETUP
727 026356 012737 000024 002214  MOV    #20.,LOOPCNT  ;PERFORM 20 ITERATIONS
728 026364          T5LOOP:
729 026364 005037 002220      CLR    FATFLG        ;CLEAR THE FATAL ERROR FLAG
730
731 026370          BGNSUB      ;//////////////// BEGIN SUBTEST //////////////////
      026370
      026370 104402          T5.1:
732                                     TRAP    C#BSUB
733 026372 004737 015774      JSR    PC,SOFINIT    ;DO A SOFT TO START
734 026376 103404          BCS    10#          ;BRANCH IF O.K.
738 026400          ERRDF     ERRNO,SFIERR,SFIMSG ;REPORT ERROR AND DROP DRIVE
      026400 104455          TRAP    C#ERDF
      026402 000765          .WORD   501
      026404 003650          .WORD   SFIERR
    
```


TSVSA - HARDWARE TESTS MACRO M1113 14-JUN-84 15:15
 TEST 5: SECOND INITIALIZATION TEST

SEQ 0119

783	026636	005002		CLR	R2		;MEMORY EXPECTED SHOULD BE 000000		
784	026640	004737	016336	JSR	PC,CHKTSSR		;WAIT FOR READY, NON-AMBIGUOUS		
785	026644	010465	000000	304:	MOV	R4,TSDB(R5)	;SELECT LOCATION SPECIFIED		
786	026650	004737	016336	JSR	PC,CHKTSSR		;WAIT FOR READY, NON-AMBIGUOUS		
787	026654	116501	000000	MOVB	TSBA(R5),R1		;READ LOC CONTENTS		
788	026660	120102		CMPB	R1,R2		;CHECK MEMORY FOR 000000		
789	026662	001406		BEQ	404		;BRANCH IF DATA OKAY		
790	026664			ERRDF	ERRNO,TSMEM,SFFMSG		;MEMORY NOT ZERO AFTER INIT.		
	026664	104455					TRAP	C1ERDF	
	026666	000766					.WORD	502	
	026670	027350					.WORD	TSMEM	
	026672	012102					.WORD	SFFMSG	
791	026674	005237	002220	404:	INC	FATFLG	;SET THE FATAL ERROR FLAG		
792	026700			CKLOOP					
793	026702	104406		ESCAPE	SUB		;EXIT ON FATAL ERROR	TRAP	C1CLP1
	026702	104410						TRAP	C1ESCAPE
	026704	000012					.WORD	L10050-	
794	026706	005204		INC	R4		;LOOK AT NEXT RAM LOC.		
795	026710	020427	000400	CMP	R4,#400		;AT TOP OF RAM ADDRESS SPACE		
796	026714	001353		BNE	304		;BRANCH TILL ALL MEMORY TESTED		
797									
798	026716			ENDSUB			;////////// END SUBTEST //////////		
	026716						L10050:		
	026716	104403					TRAP	C1ESUB	
799									
800	026720	005737	002220	TST	FATFLG		;IS FATAL ERROR FLAG SET ?		
801	026724	001404		BEQ	504		;BRANCH IF NOT		
802	026726	004737	017202	JSR	PC,CKDROP		;NO LOOP, TRY TO DROP DEVICE		
803	026732	005037	002220	CLR	FATFLG		;CLEAR THE FATAL ERROR FLAG		
804	026736			504:					
805									
806	026736			BGNSUB			;////////// BEGIN SUBTEST //////////		
	026736						TS.2:		
	026736	104402					TRAP	C1BSUB	
807									
808	026740	004737	015774	JSR	PC,SOFINIT		;DO A SOFT TO START		
809	026744	103404		BCS	104		;BRANCH IF O.K.		
813	026746			ERRDF	ERRNO,SFIERR,SFIMSG		;REPORT ERROR AND DROP DRIVE		
	026746	104455					TRAP	C1ERDF	
	026750	000767					.WORD	503	
	026752	003650					.WORD	SFIERR	
	026754	012034					.WORD	SFIMSG	
814	026756	012702	177777	104:	MOV	#-1,R2	;ALL ONE DATA PATTERN		
815	026762	005004		CLR	R4		;STARTING RAM ADDRESS		
816	026764	004737	016336	JSR	PC,CHKTSSR		;WAIT FOR READY, NON-AMBIGUOUS		
817	026770	105065	000000	154:	CLRB	TSDB(R5)	;SET MAINTENANCE MODE		
818	026774	004737	016336	JSR	PC,CHKTSSR		;WAIT FOR READY, NON-AMBIGUOUS		
819	027000	010465	000000	MOV	R4,TSDB(R5)		;SET THE NEXT RAM ADDRESS		
820	027004	004737	016336	JSR	PC,CHKTSSR		;WAIT FOR READY, NON-AMBIGUOUS		
821	027010	110265	000000	MOVB	R2,TSDB(R5)		;LOAD TEST DATA		
822	027014	005204		INC	R4		;NEXT ADDRESS TO TEST		
823	027016	020427	007777	CMP	R4,#7777		;COMPARE TO LAST ADDRESS		
824	027022	003762		BLE	154		;BRANCH TILL ALL DATA WRITTEN		
825	027024	005065	000002	CLR	TSSR(R5)		;ISSUE A SOFT RESET		
826	027030	004737	016336	JSR	PC,CHKTSSR		;WAIT FOR READY, NON-AMBIGUOUS		
827	027034	016501	000002	MOV	TSSR(R5),R1		;GET THE CONTENTS OF TSSR		

TSV5A - HARDWARE TESTS MACRO M1113 14-JUN-84 15:15
 TEST 5: SECOND INITIALIZATION TEST

SEQ 0121

```

027252 104410                                TRAP  C#ESCAPE
027254 000012                                .WORD L10051-.
869 027256 005204                            INC    R4                                ;LOOK AT NEXT RAM LOC.
870 027260 020427 000400                    CMP    R4,#400                          ;AT TOP OF RAM ADDRESS SPACE
871 027264 001353                            BNE    30#                               ;BRANCH TILL ALL MEMORY TESTED
872
873 027266                                ENDSUB                                ;////////// END SUBTEST ////////////
027266                                L10051:                                TRAP  C#ESUB
027266 104403
874
875 027270 005737 002220                    TST    FATFLG                          ;IS FATAL ERROR FLAG SET ?
876 027274 001402                            BEQ    50#                               ;BRANCH IF NOT
877 027276 004737 017202                    JSR    PC,CKDROP                       ;NO LOOP, TRY TO DROP DEVICE
878 027302 004737 016456                    50#:  JSR    PC,TSTLOOP                  ;SHOULD WE DO ITERATIONS ?
879 027306 103002                            BCC    60#                               ;BRANCH IF NOT
880 027310 000137 026364                    JMP    T5LOOP                          ;LOOP UNTIL COUNT EXPIRED
881 027314                                60#:  EXIT    TST                       ;ALL DONE THIS TEST
027314 104432                                TRAP  C#EXIT
027316 000132                                .WORD L10047-.
882
883
884 ;*
885 ;LOCAL TEXT MESSAGES FOR TEST
886 ;-
887 027320 105 170 164 TSTSID: .ASCIZ 'Extended Initialization'
888 027350 111 156 143 TSMEM: .ASCIZ 'Incorrect RAM Data After Init'
889 027406 111 156 143 TSADDR: .ASCIZ 'Incorrect RAM Address After Init'
890 .EVEN
891 027450                                ENDTST
027450                                L10047:                                TRAP  C#ETST
027450 104401
892
893
894 .SBTTL TEST 6: COMMAND REJECT
895
896
897 ;
898 ; THIS TEST VERIFIES THAT ALL COMMANDS OTHER THAN WRITE
899 ; CHARACTERISTICS ARE REJECTED DUE TO THE NEED BUFFER ADDRESS
900 ; (NBA) BIT BEING SET IN TSSR, AND THAT THE TSBA AND TSSR
901 ; REGISTERS ARE LEFT IN THE PROPER STATE AFTER EACH COMMAND IS
902 ; REJECTED. THIS TEST CHECKS MICROPROCESSOR SEQUENCING, BASIC
903 ; COMMAND DECODING AND DATI DMA HANDLING. THIS TEST CONTAINS TWO
904 ; SUBTESTS: SUBTEST 1 SEQUENCES THROUGH ALL COMMAND WORDS (OTHER
905 ; THAN WRITE CHARACTERISTICS) WITH THE INTERRUPT ENABLE (IE) BIT
906 ; CLEAR AND VERIFIES THAT AN INTERRUPT IS NOT GENERATED BY THE
907 ; REJECTED COMMAND; SUBTEST 2 PERFORMS SIMILARLY TO SUBTEST 1 BUT
908 ; SETS THE IE BIT IN EACH COMMAND WORD AND VERIFIES THAT AN
909 ; INTERRUPT IS GENERATED WHEN THE COMMAND IS REJECTED. SUBTEST 1
910 ; SETS UP THE INTERRUPT SERVICE ROUTINE TO FLAG UNEXPECTED
911 ; INTERRUPTS. THE COMMAND WORD IN THE COMMAND BUFFER IS
912 ; INITIALIZED TO 100000 (OCTAL) AND THE REMAINING THREE WORDS IN
913 ; THE COMMAND BUFFER ARE SET TO KNOWN UNIQUE PATTERNS. THEN THE
914 ; FOLLOWING SEQUENCE IS PERFORMED:
915 ;
916 ; 1. INITIALIZE THE CONTROLLER BY WRITING INTO THE TSSR;
917 ; PROPER INITIAL CONDITIONS ARE VERIFIED.

```


Address	Offset	Label	Instruction	Comments	Trap	Code
976	027524		ERRDF	ERRNO, SFIERR, SFIMSG		
	027524	104455			TRAP	C#ERDF
	027526	001131			.WORD	601
	027530	003650			.WORD	SFIERR
	027532	012034			.WORD	SFIMSG
977	027534	005037	10#: CLR	FATFLG		
	027534	005037	CLR	INTRECV		
978	027540	004737	10#: JSR	PC, CHKTSSR		
	027540	004737	JSR	PC, CHKTSSR		
979	027544	004737	10#: BIC	#BIT7, (R4)		
	027544	004737	BIC	#BIT7, (R4)		
980	027550	042714	10#: MOV	R4, TSDB(R5)		
	027550	042714	MOV	R4, TSDB(R5)		
981	027554	010465	10#: JSR	PC, WAITF		
	027554	010465	JSR	PC, WAITF		
982	027560	004737	10#: BCS	15#		
	027560	004737	BCS	15#		
983	027564	103407	10#: MOV	R0, R1		
	027564	103407	MOV	R0, R1		
984	027566	010001	10#: ERRDF	ERRNO, T6SSR, PKTSSR		
	027566	010001	ERRDF	ERRNO, T6SSR, PKTSSR		
	027570	104455			TRAP	C#ERDF
	027572	001132			.WORD	602
	027574	030415			.WORD	T6SSR
	027576	012046			.WORD	PKTSSR
989	027600	005237	15#: INC	FATFLG		
	027600	005237	INC	FATFLG		
990	027604	104406	15#: CKLOOP			
	027604	104406	CKLOOP			
991	027606	104410	15#: ESCAPE	SUB		
	027606	104410	ESCAPE	SUB		
	027610	000170			TRAP	C#CLP1
	027612	005737			.WORD	C#ESCAPE
	027616	001404			.WORD	L10053-
992	027612	005737	15#: TST	INTRECV		
	027612	005737	TST	INTRECV		
993	027616	001404	15#: BEQ	22#		
	027616	001404	BEQ	22#		
997	027620	104456	15#: ERRHRD	ERRNO, T6INT, PKTSSR		
	027620	104456	ERRHRD	ERRNO, T6INT, PKTSSR		
	027622	001133			TRAP	C#ERHRD
	027624	030473			.WORD	603
	027626	012046			.WORD	T6INT
	027626	012046			.WORD	PKTSSR
998	027630	012702	22#: MOV	#SC!NBA!SSR!TSREJ, R2		
	027630	012702	MOV	#SC!NBA!SSR!TSREJ, R2		
999	027634	004737	22#: JSR	PC, CHKTSSR		
	027634	004737	JSR	PC, CHKTSSR		
1000	027640	016501	22#: MOV	TSSR(R5), R1		
	027640	016501	MOV	TSSR(R5), R1		
1001	027644	032701	22#: BIT	#OFL, R1		
	027644	032701	BIT	#OFL, R1		
1002	027650	001402	22#: BEQ	25#		
	027650	001402	BEQ	25#		
1003	027652	052702	22#: BIS	#OFL, R2		
	027652	052702	BIS	#OFL, R2		
1004	027656	020201	25#: CMP	R2, R1		
	027656	020201	CMP	R2, R1		
1005	027660	001404	25#: BEQ	30#		
	027660	001404	BEQ	30#		
1009	027662	104456	25#: ERRHRD	ERRNO, T6NBA, PKTSSR		
	027662	104456	ERRHRD	ERRNO, T6NBA, PKTSSR		
	027664	001134			TRAP	C#ERHRD
	027666	030370			.WORD	604
	027670	012046			.WORD	T6NBA
	027670	012046			.WORD	PKTSSR
1010	027672	104406	30#: CKLOOP			
	027672	104406	CKLOOP			
	027672	104406			TRAP	C#CLP1
1011	027674	004737	30#: JSR	PC, CHKTSSR		
	027674	004737	JSR	PC, CHKTSSR		
1012	027700	016501	30#: MOV	TSBA(R5), R1		
	027700	016501	MOV	TSBA(R5), R1		
1013	027704	010402	30#: MOV	R4, R2		
	027704	010402	MOV	R4, R2		
1014	027706	062702	30#: ADD	#10, R2		
	027706	062702	ADD	#10, R2		
1015	027712	020102	30#: CMP	R1, R2		
	027712	020102	CMP	R1, R2		
1016	027714	001404	30#: BEQ	35#		
	027714	001404	BEQ	35#		
1020	027716	104456	30#: ERRHRD	ERRNO, T6TSBA, EXPREC		
	027716	104456	ERRHRD	ERRNO, T6TSBA, EXPREC		
	027720	001135			TRAP	C#ERHRD
	027722	030631			.WORD	605
	027724	015474			.WORD	T6TSBA
	027724	015474			.WORD	EXPREC


```

1113 030274 042700 177740          BIC    #177740,R0          ;GET BITS 0-4
1114 030300 020027 000004          CMP    R0,#4             ;DON'T TEST WRITE CHARACTERISTICS
1115 030304 001002                   BNE    45$               ;BRANCH IF NOT WRITE CHARACTERISTICS
1116 030306 062703 000002          ADD    #2,R3             ;BY-PASS WRITE CHARACTERISTICS
1117 030312 020327 003060          45$:  CMP    R3,#TBLEND   ;HAVE WE COMPLETED DATA TABLE ?
1118 030316 103002                   BHIS   50$               ;BRANCH IF ALL TESTED
1119 030320 000137 030034          JMP    5$                ;TEST WITH NEXT DATA
1120
1121 030324 030324 104403          50$:  ENDSUB             ;////////////////// END SUBTEST ////////////////////
                                L10054:
1122 030326 005737 002220          TST    FATFLG            ;ANY FATAL ERRORS ?
1123 030332 001402                   BEQ    60$               ;BRANCH IF NOT
1124 030334 004737 017202          JSR    PC,CKDROP         ;TRY TO DROP THE UNIT
1125 030340 004737 016456          60$:  JSR    PC,TSTLOOP   ;SHOULD WE DO ITERATIONS ?
1126 030344 103002                   BCC    62$               ;BRANCH IF NOT
1127 030346 000137 027470          JMP    T6LOOP            ;LOOP UNTIL COUNT EXPIRED
1128 030352 030352 104432          62$:  EXIT    TST         ;ALL DONE THIS TEST
                                TRAP    C$EXIT
                                .WORD   L10052-.
1129
1130
1131
1132
1133

```

```

1135 030356
1137 030360
1138 030360 000000
1139 030362 052525
1140 030364 125252
1141 030366 052525
1142
1143
1144
1145
1146
1147

```

```

;+
;LOCAL STORAGE FOR THIS TEST
;-
                                .BLKB   10-<.-TSV2&7>
16PACKET:
                                .WORD   0
                                .WORD   052525
                                .WORD   125252
                                .WORD   052525

```

```

;COMMAND PACKET FOR TEST
;WILL CONTAIN VARIABLE COMMANDS

```

```

1148 030370 103 157 155 T6NBA:
1149 030415 103 157 156 T6SSR:
1150 030473 125 156 145 T6INT:
1151 030551 105 170 160 T6NINT:
1152 030631 111 156 143 T6TSBA:
1153 030703 103 157 155 TST6ID:
1154
1155 030722
1156 030722 104401
1157
1158
1159
1160
1161
1162
1163
1164
1165

```

```

;+
;LOCAL TEXT MESSAGES FOR TEST
;-
                                .ASCIZ  'Command Not Rejected'
                                .ASCIZ  'Contents of TSSR Incorrect After Write Packet'
                                .ASCIZ  'Unexpected Interrupt Received On Write Packet'
                                .ASCIZ  'Expected Interrupt Not Received On Write Packet'
                                .ASCIZ  'Incorrect TSBA Address After Packet Write'
                                .ASCIZ  'Command Reject'
                                .EVEN
                                ENDTST

```

```

L10052:
                                TRAP    C$ETST

```

.SBTTL TEST 7: WRITE CHARACTERISTICS

```

;
; THIS TEST VERIFIES BASIC OPERATION OF THE WRITE CHARACTERISTICS
; COMMAND. IT VERIFIES THAT THE COMMAND BLOCK AND CHARACTERISTICS
; DATA BLOCK ARE FETCHED PROPERLY FROM CPU MEMORY, THE NEED BUFFER
; ADDRESS (NBA) BIT IN TSSR IS HANDLED PROPERLY, AND THAT A PROPER
; MESSAGE PACKET IS STORED, WHERE APPROPRIATE. THIS TEST DOES NOT
; CHECK THAT THE VARIOUS FUNCTIONS ENABLED BY CHARACTERISTIC MODE
;

```



```

1687 033066 000010          .WORD 8.          ;STARTING VALUE OF BLOCK SIZE
1688
1689 033070          T7DATA:          ;CHARACTERISTICS DATA BLOCK
1690 033070 033106          .WORD T7BFR      ;ADDRESS OF MESSAGE BUFFER
1691 033072 000000          .WORD 0
1692 033074 000016          .WORD 14.        ;LENGTH OF MESSAGE BUFFER
1693 033076 000000          .WORD 0
1694 033100 000000          T7SP: .WORD 0      ;EXTFEA EXTRA WORD
1695
1696 033102 000000 000000          .WORD 0,0        ;SPACE
1697 033106          T7BFR: .BLKW 8.    ;MESSAGE BUFFER
1698
1699
1700
1701          ;*
1702          ;TEST DATA FOR SUBTEST TWO
1703          ;
1704          ;DATA HAS FORMAT:
1705          ;
1706          ;          1ST WORD          OFFSET TO TEST WORD IN PACKET
1707          ;          2ND WORD          BITS TO SET FOR TEST
1708          ;
1709          ;-
1710          T72DATA:
1711 033126          .WORD 0,BIT5!BIT6!BIT9!BIT10!BIT11!BIT12!BIT13
1712 033132 000000 037140          .WORD 2,BIT0
1713 033136 000004 100100          .WORD 4,BIT6!BIT15
1714          T72DONE=.
1715
1716
1717          ;*
1718          ;LOCAL TEXT MESSAGES FOR TEST
1719          ;-
1720
1721 033142          116          102          101 T72NBA: .ASCIZ 'NBA Not Set On Rejected WRITE CHARACTERISTICS'
1722 033220          127          122          111 T7NBA: .ASCIZ 'WRITE CHARACTERISTICS Command Not Accepted'
1723 033273          127          122          111 T72REJ: .ASCIZ 'WRITE CHARACTERISTICS Not Rejected With Non-Zero Unused Fields'
1724 033372          127          122          111 T73REJ: .ASCIZ 'WRITE CHARACTERISTICS Not Rejected With Invalid Data Count'
1725 033465          127          122          111 T74REJ: .ASCIZ 'WRITE CHARACTERISTICS Not Rejected With Invalid Block Address'
1726 033563          127          122          111 T75REJ: .ASCIZ 'WRITE CHARACTERISTICS Not Rejected With Invalid Buffer Length'
1727 033661          103          157          156 T7SSR: .ASCIZ 'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
1728 033750          105          170          160 T7NINT: .ASCIZ 'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
1729 034041          125          156          145 T7INT: .ASCIZ 'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
1730 034130          111          156          143 T7TSBA: .ASCIZ 'Incorrect TSBA Address After WRITE CHARACTERISTICS'
1731 034213          127          162          151 TST7ID: .ASCIZ 'Write Characteristics'
1732          .EVEN
1733
1734
1735          ;*
1736          ;
1737          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
1738          ;
1739          ;-
1740
1741 034242          T7REST:
1742 034242          SAVREG          ;SAVE THE REGISTERS
1743 034246 012701 033060          MOV #T7PACKET,R1 ;START OF THE PACKET

```

```

1744 034252 012721 100004      MOV      #100004,(R1)+    ;WRITE CHARACTERISTICS WITH ACK
1745 034256 012721 033070      MOV      #T7DATA,(R1)+  ;ADDRESS OF CHAR DATA BLOCK
1746 034262 005021              CLR      (R1)+          ;EXTENDED ADDRESS
1747 034264 012721 000010      MOV      #8,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
1748 034270 012721 033106      MOV      #T7BFR,(R1)+  ;ADDRESS OF MESSAGE BUFFER
1749 034274 005021              CLR      (R1)+
1750 034276 012721 000020      MOV      #16,(R1)+     ;LENGTH OF MESSAGE BUFFER
1751 034302 005021              CLR      (R1)+
1752 034304 005011              CLR      (R1)+
1753 034306 000207              RTS       PC            ;RETURN
1754 034310              ENDTST
                                L10055:
                                TRAP    C$ETST
                                034310 104401
    
```

.SBTTL TEST 8: VOLUME CHECK

```

1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
    
```

THIS TEST VERIFIES THAT THE VOLUME CHECK (VCK) BIT, A FLAG HELD WITHIN THE M7196 AND APPEARING IN XSTO, IS SET BY INITIALIZE AND CLEARED BY EXECUTING A WRITE CHARACTERISTICS COMMAND WITH THE CVC BIT SET. IT IS ALSO VERIFIED THAT A WRITE CHARACTERISTICS COMMAND WITH THE CVC BIT CLEAR DOES NOT AFFECT THE STATE OF THE VOLUME CHECK BIT. THE ACTUAL FUNCTION OF VOLUME CHECK, THAT OF PREVENTING OR ALLOWING A TAPE MOTION COMMAND DEPENDING UPON WHETHER VOLUME CHECK IS SET OR CLEAR, IS NOT CHECKED BY THIS TEST; THIS FUNCTIONALITY IS CHECKED IN THE INDIVIDUAL TESTS OF TAPE MOTION COMMANDS.

THE TEST PROCEEDS AS FOLLOWS:

1. THE CONTROLLER IS INITIALIZED BY WRITING INTO THE TSSR.
2. A WRITE CHARACTERISTICS COMMAND IS ISSUED (WITH CVC=0) AND XSTO IN THE RETURNED MESSAGE BUFFER IS EXAMINED; THE VCK BIT SHOULD BE CLEAR (0).
3. THE PREVIOUS STEP IS REPEATED TO VERIFY THAT VCK DOES NOT CHANGE (REMAINS AT 0).
4. A WRITE CHARACTERISTICS COMMAND IS ISSUED WITH CVC=1 AND THE VCK BIT IN XSTO IN THE MESSAGE BUFFER IS EXAMINED; THE VCK BIT SHOULD BE CLEAR (0).
5. A WRITE CHARACTERISTICS COMMAND IS ISSUED WITH CVC=0 AND THE VCK BIT IN XSTO IN THE MESSAGE BUFFER IS EXAMINED; THE VCK BIT SHOULD REMAIN CLEAR (0).

```

1791 034312              BGNTST
                                T8::
1796 034312 012700 035177      MOV      #TST8ID,R0    ;ASCII MESSAGE TO IDENTIFY TEST
1797 034316 004737 016510      JSR      PC,TSTSETUP   ;DO INITIAL TEST SETUP
1798 034322 012737 000024 002214  MOV      #20,LOOPCNT   ;PERFORM 20 ITERATIONS
1799 034330              T8LOOP:
1800
1801 034330 012704 034720      MOV      #T8PACKET,R4 ;PACKET FOR WRITE CHARACTERISTICS
    
```

TSVSA - HARDWARE TESTS MACRO M1113 14-JUN-84 15:15
 TEST 8: VOLUME CHECK

SEQ 0140

1802	034334	004737	015774	5#:	JSR	PC,SOFINIT	;DO SOFT INIT OF CONTROLLER
1803	034340	103405			BCS	10#	;BR IF SOFT INIT = OK
1807	034342	010001			MOV	R0,R1	;SAVE CONTENTS OF TSSR
1808	034344				ERRDF	ERRNO,SFIERR,SFIMSG	;DEVICE FATAL ERROR DURING INIT
	034344	104455					TRAP C#ERDF
	034346	001441					.WORD 801
	034350	003650					.WORD SFIERR
	034352	012034					.WORD SFIMSG
1809	034354	042714	040000	10#:	BIC	#BIT14,(R4)	;CLEAR THE CVC BIT
1810	034360	010465	000000		MOV	R4,TSDB(R5)	;SET THE PACKET ADDRESS FOR WRITE CHAR
1811	034364	004737	J16336		JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
1812	034370	103405			BCS	15#	;BR IF CARRY SET (GOOD RETURN)
1813	034372	010001			MOV	R0,R1	;SAVE CONTENTS OF TSSR
1817	034374				ERRDF	ERRNO,T8SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	034374	104455					TRAP C#ERDF
	034376	001442					.WORD 802
	034400	035110					.WORD T8SSR
	034402	012046					.WORD PKTSSR
1818	034404			15#:	CKLOOP		;LOOP ON ERROR, IF FLAG SET
	034404	104406					TRAP C#CLP1
1819	034406				ESCAPE	TST	;EXIT IF FATAL ERROR
	034406	104410					TRAP C#ESCAPE
	034410	000604					.WORD L10064--
1820	034412	012702	034742		MOV	#T8BFR,R2	;ADDRESS OF THE MESSAGE BUFFER
1821	034416	032762	000020	000006	BIT	#XSOVCK,XSTO(R2)	;IS VOLUME CHECK SET IN XSTO ?
1822	034424	001406			BEQ	20#	;OKAY IF VOLUME CHECK IS CLEAR
1826	034426	016501	000002		MOV	TSSR(R5),R1	;CONTENTS OF TSSR FOR ERROR REPORT
1827	034432				ERRHRD	ERRNO,T8NVCK,PKTMES	;VOLUME CHECK NOT CLEAR
	034432	104456					TRAP C#ERHRD
	034434	001443					.WORD 803
	034436	035017					.WORD T8NVCK
	034440	012110					.WORD PKTMES
1828	034442			20#:	CKLOOP		;LOOP ON ERROR ?
	034442	104406					TRAP C#CLP1
1829	034444	010465	000000		MOV	R4,TSDB(R5)	;SET THE PACKET ADDRESS FOR WRITE CHAR
1830	034450	004737	016336		JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET
1831	034454	103405			BCS	25#	;BR IF CARRY SET (GOOD RETURN)
1832	034456	010001			MOV	R0,R1	;SAVE CONTENTS OF TSSR
1836	034460				ERRDF	ERRNO,T8SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	034460	104455					TRAP C#ERDF
	034462	001444					.WORD 804
	034464	035110					.WORD T8SSR
	034466	012046					.WORD PKTSSR
1837	034470			25#:	CKLOOP		;LOOP ON ERROR, IF FLAG SET
	034470	104406					TRAP C#CLP1
1838	034472				ESCAPE	TST	;EXIT IF FATAL ERROR
	034472	104410					TRAP C#ESCAPE
	034474	000520					.WORD L10064--
1839	034476	032762	000020	000006	BIT	#XSOVCK,XSTO(R2)	;IS VOLUME CHECK SET IN XSTO ?
1840	034504	001406			BEQ	30#	;OKAY IF VOLUME CHECK IS SET
1844	034506	016501	000002		MOV	TSSR(R5),R1	;CONTENTS OF TSSR FOR ERROR REPORT
1845	034512				ERRHRD	ERRNO,T8NVCK,PKTMES	;VOLUME CHECK NOT SET
	034512	104456					TRAP C#ERHRD
	034514	001445					.WORD 805
	034516	035017					.WORD T8NVCK
	034520	012110					.WORD PKTMES
1846	034522			30#:	CKLOOP		;LOOP ON ERROR ?


```

1889
1890
1891
1892
1893
1895 034712
1897 034720
1898 034720 100004
1899 034722 034730
1900 034724 000000
1901 034726 000010
1902
1903 034730
1904 034730 034742
1905 034732 000000
1906 034734 000020
1907 034736 000000 000000
1908
1909 034742
1910
1911
1912
1913
1914
1915
1916 034762 126 157 154 T8VCK: .ASCIZ 'Volume Check Bit Not Cleared'
1917 035017 126 157 154 T8NVCK: .ASCIZ 'Volume Check Bit (VCK) Not Clear After Initialize (XST0)'
1918 035110 103 157 156 T8SSR: .ASCIZ 'Contents of TSSR Incorrect After Write Characteristics'
1919 035177 126 157 154 T8T8ID: .ASCIZ 'Volume Check'
1920
1921 035214
035214
035214 104401
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942 035216
035216
1947 035216 005037 002224
1948 035222 012700 040241

```

```

;+
;LOCAL STORAGE FOR THIS TEST
;-
      .BLKB 10-<.-TSV2&7>
T8PACKET:
      .WORD 100004
      .WORD T8DATA
      .WORD 0
      .WORD 10
;COMMAND PACKET FOR TEST
;WRITE CHARACTERISTICS COMMAND
;ADDRESS OF CHARACTERISTICS BLOCK
;STARTING VALUE OF COUNTER
T8DATA:
      .WORD T8BFR
      .WORD 0
      .WORD 16.
      .WORD 0.0
;CHARACTERISTICS DATA BLOCK
;ADDRESS OF MESSAGE BUFFER
;LENGTH OF MESSAGE BUFFER
T8BFR: .BLKW 8.
;MESSAGE BUFFER

;+
;LOCAL TEXT MESSAGES FOR TEST
;-
T8VCK: .ASCIZ 'Volume Check Bit Not Cleared'
T8NVCK: .ASCIZ 'Volume Check Bit (VCK) Not Clear After Initialize (XST0)'
T8SSR: .ASCIZ 'Contents of TSSR Incorrect After Write Characteristics'
T8T8ID: .ASCIZ 'Volume Check'
      .EVEN
      ENDTST
L10064: TRAP C$ETST

.SBTTL TEST 9: COMPLETION INTERRUPT

;
; THIS TEST VERIFIES THAT AN INTERRUPT IS GENERATED AT THE
; COMPLETION OF THE WRITE CHARACTERISTICS COMMAND IF THE INTERRUPT
; ENABLE (IE) BIT IN THE COMMAND HEADER WORD IS SET. THIS TEST
; CHECKS THE FUNCTIONING OF THE INTERRUPT LOGIC AND BASIC
; PROCESSING OF THE IE BIT.
;
; THE SEQUENCES OF TEST 7 ARE REPEATED, EXCEPT THAT THE INTERRUPT
; SERVICE ROUTINE IS SET UP TO EXPECT INTERRUPTS AND EACH WRITE
; CHARACTERISTICS COMMAND IS ISSUED WITH THE IE BIT SET (1). IT
; IS VERIFIED, WHERE APPROPRIATE, THAT THE IE STATUS BIT IN XST0
; OF ANY MESSAGE PACKET IS SET AND THAT A COMPLETION INTERRUPT IS
; GENERATED. FINALLY, A SEQUENCE OF TWO COMMANDS ARE ISSUED, THE
; FIRST WITH IE=1 AND THE SECOND WITH IE=0. IT IS VERIFIED THAT
; NO INTERRUPT IS GENERATED AFTER THE SECOND COMMAND AND THAT THE
; IE BIT IN XST0 IS 0.
;
BGNTST
      CLR EXTFEA
      MOV #TST9ID,R0
T9::
;CLEAR EXTENDED FEATURES SWITCH
;ASCII MESSAGE TO IDENTIFY TEST

```



```

2364 037024 004737 016336      JSR    PC,CHKTSSR      ;WAIT FOR SSR TO SET
2365 037030 103405             BCS    15$             ;BR IF CARRY SET (GOOD RETURN)
2366 037032 010001             MOV    R0,R1          ;SAVE CONTENTS OF TSSR
2370 037034             ERRDF  ERRNO,T9SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP    C$ERDF
                                .WORD   926
                                .WORD   T9SSR
                                .WORD   PKTSSR
2371 037044             15$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
2372 037046             ESCAPE SUB           ;BY-PASS SUBTEST IF FATAL ERROR
                                TRAP    C$CLP1
                                .WORD   C$ESCAPE
                                .WORD   L10074-.
2373 037052 005737 002222      TST    INTRECV        ;DID AN INTERRUPT OCCUR ?
2374 037056 001004             BNE    22$           ;BRANCH IF YES
2378 037060             ERRHRD ERRNO,T9INT,PKTSSR
                                TRAP    C$ERHRD
                                .WORD   927
                                .WORD   T9INT
                                .WORD   PKTSSR
2379 037070             22$:  CKLOOP          ;LOOP ON ERROR ?
                                TRAP    C$CLP1
2380
2381 037072 005037 002222      CLR    INTRECV        ;CLEAR INTERRUPT RECEIVED FLAG
2382 037076 042714 000200      BIC    @BIT7,(R4)    ;DISABLE INTERRUPTS
2383 037102 010465 000000      MOV    R4,TSDB(R5)  ;SET THE PACKET ADDRESS
2384 037106 004737 016336      JSR    PC,CHKTSSR      ;WAIT FOR SSR TO SET
2385 037112 103405             BCS    25$           ;BR IF CARRY SET (GOOD RETURN)
2386 037114 010001             MOV    R0,R1          ;SAVE CONTENTS OF TSSR
2390 037116             ERRDF  ERRNO,T9SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
                                TRAP    C$ERDF
                                .WORD   928
                                .WORD   T9SSR
                                .WORD   PKTSSR
2391 037126             25$:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET
2392 037130             ESCAPE SUB           ;BY-PASS SUBTEST IF FATAL ERROR
                                TRAP    C$CLP1
                                .WORD   C$ESCAPE
                                .WORD   L10074-.
2393 037134 005737 002222      TST    INTRECV        ;DID AN INTERRUPT OCCUR ?
2394 037140 001404             BEQ    30$           ;BRANCH IF NOT
2398 037142             ERRHRD ERRNO,T9INT,PKTSSR
                                TRAP    C$ERHRD
                                .WORD   929
                                .WORD   T9INT
                                .WORD   PKTSSR
2399 037152             30$:  ENDSUB          ;////////////////// END SUBTEST ////////////////////
2400 037152             .WORD   L10074:
2401 037154             TRAP    C$ESUB
2402 037154             104403
2403 037154             EXIT    TST          ;ALL DONE THIS TEST
2404 037156             TRAP    C$EXIT
2405             .WORD   L10065-.

;+
;LOCAL STORAGE FOR THIS TEST

```

```

2406      ; -
2407
2409 037160      .BLKB  10-<.-TSV2&7>
2411 037170      T9PACKET:      ;COMMAND PACKET FOR TEST
2412 037170 100204      .WORD  100204      ;WRITE CHAR COMMAND, WITH IE, ACK
2413 037172 037200      .WORD  T9DATA      ;ADDRESS OF CHARACTERISTICS BLOCK
2414 037174 000000      .WORD  0
2415 037176 000010      .WORD  8.      ;STARTING VALUE OF BLOCK SIZE
2416
2417 037200      T9DATA:      ;CHARACTERISTICS DATA BLOCK
2418 037200 037212      .WORD  T98FR      ;ADDRESS OF MESSAGE BUFFER
2419 037202 000000      .WORD  0
2420 037204 000016      .WORD  14.      ;LENGTH OF MESSAGE BUFFER
2421 037206 000000 000000      .WORD  0,0
2422
2423 037212      T98FR:  .BLKW  8.      ;MESSAGE BUFFER
2424
2425      ;*
2426      ;
2427      ;TEST DATA FOR SUBTEST TWO
2428      ;
2429      ;DATA HAS FORMAT:
2430      ;
2431      ;      1ST WORD      OFFSET TO TEST WORD IN PACKET
2432      ;      2ND WORD      BITS TO SET FOR TEST
2433      ;
2434      ; -
2435
2436 037232      T92DATA:
2437 037232 000000 037140      .WORD  0,BIT5!BIT6!BIT9!BIT10!BIT11!BIT12!BIT13
2438 037236 000002 000001      .WORD  2,BIT0
2439 037242 000004 100100      .WORD  4,BIT6!BIT15
2440
2441      T92DONE=.
2442
2443      ;*
2444      ;LOCAL TEXT MESSAGES FOR TEST
2445      ; -
2446
2447 037246      127      122      111  T9NBA:  .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
2448 037321      127      122      111  T92REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Non-Zero Unused Fields'
2449 037420      127      122      111  T93REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Data Count'
2450 037513      127      122      111  T94REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Block Address'
2451 037611      127      122      111  T95REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Buffer Length'
2452 037707      103      157      156  T9SSR:  .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
2453 037776      105      170      160  T9NINT: .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
2454 040067      125      156      145  T9INT:  .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
2455 040156      111      156      143  T9TSBA: .ASCIZ  'Incorrect TSBA Address After WRITE CHARACTERISTICS'
2456 040241      103      157      155  T9T9ID: .ASCIZ  'Completion Interrupt'
2457      .EVEN
2458
2459
2460      ;*
2461      ;
2462      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
2463      ;
2464      ; -

```


TSVSA - HARDWARE TESTS MACRO M1113 14-JUN-84 15:15
 TEST 10: BASIC PACKET PROTOCOL

SEQ 0165

```

2989 042544 000000          .WORD 0
2990 042546 000016          .WORD 14.          ;LENGTH OF MESSAGE BUFFER
2991 042550 000000 000000  .WORD 0,0
2992
2993 042554          T10BUFR: .BLKW 8.          ;MESSAGE BUFFER
2994
2995          ;+
2996          ;LOCAL TEXT MESSAGES FOR TEST
2997          ;-
2998
2999
3000 042574          115      145      163  T10MBF: .ASCIZ 'Message Buffer Modified after MESSAGE BUFFER RELEASE Command'
3001 042671          116      102      101  T10NBA: .ASCIZ 'NBA Not Clear After WRITE CHARACTERISTICS Command'
3002 042753          116      102      101  T10NNBA: .ASCIZ 'NBA Set After MESSAGE BUFFER RELEASE Command'
3003 043030          103      157      156  T10SSR: .ASCIZ 'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
3004 043117          105      170      160  T10INT: .ASCIZ 'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
3005 043210          125      156      145  T10INT: .ASCIZ 'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
3006 043277          102      141      163  TST10ID: .ASCIZ 'Basic Packet Protocol'
3007          .EVEN
3008
3009
3010
3011          ;+
3012          ;
3013          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
3014          ;
3015          ;-
3016
3017 043326          T10RST:
3018 043326          SAVREG          ;SAVE THE REGISTERS
3019 043332 012701 042470  MOV      #T10PACKET,R1          ;START OF THE PACKET
3020 043336 012721 100204  MOV      #100204,(R1)+          ;WRITE CHARACTERISTICS WITH ACK, IE
3021 043342 012721 042500  MOV      #T10DATA,(R1)+          ;ADDRESS OF CHAR DATA BLOCK
3022 043346 005021          CLR      (R1)+          ;EXTENDED ADDRESS
3023 043350 012721 000010  MOV      #8.,(R1)+          ;SIZE OF DATA BLOCK IN BYTES
3024 043354 012721 042512  MOV      #T10BFR,(R1)+          ;ADDRESS OF MESSAGE BUFFER
3025 043360 005021          CLR      (R1)+
3026 043362 012721 000016  MOV      #14.,(R1)+          ;LENGTH OF MESSAGE BUFFER
3027 043366 005021          CLR      (R1)+
3028 043370 005011          CLR      (R1)
3029 043372 005037 042512  CLR      T10BFR          ;CLEAR 1ST LOC IN MESSAGE BUFFER
3030 043376 000207          RTS      PC          ;RETURN
3031          ;+
3032          ;
3033          ;ROUTINE TO RESTORE COMMAND PACKET #2 TO START-UP (DEFAULT) VALUES
3034          ;
3035          ;-
3036
3037 043400          T10RT2:
3038 043400          SAVREG          ;SAVE THE REGISTERS
3039 043404 012701 042532  MOV      #T10PKT,R1          ;START OF THE PACKET
3040 043410 012721 100204  MOV      #100204,(R1)+          ;WRITE CHARACTERISTICS WITH ACK, IE
3041 043414 012721 042542  MOV      #T10DTA,(R1)+          ;ADDRESS OF CHAR DATA BLOCK
3042 043420 005021          CLR      (R1)+          ;EXTENDED ADDRESS
3043 043422 012721 000010  MOV      #8.,(R1)+          ;SIZE OF DATA BLOCK IN BYTES
3044 043426 012721 042554  MOV      #T10BUFR,(R1)+          ;ADDRESS OF MESSAGE BUFFER
3045 043432 005021          CLR      (R1)+

```


3237	044202			ERRDF	ERRNO,SFIERR,SFIMSG	:DEVICE FATAL ERROR DURING INIT		
	044202	104455					TRAP	C#ERDF
	044204	002131					.WORD	1113
	044206	003650					.WORD	SFIERR
	044210	012034					.WORD	SFIMSG
3238	044212			36:				
3239	044212	012704	044710		MOV @T11PK2,R4	:WRITE CHARACTERISTICS PACKET		
3240	044216	004737	010662		JSR PC,WRTCHR	:ISSUE WRITE CHARACTERISTICS		
3241	044222	103404			BCS 48	:BR, IF COMMAND ISSUED OK		
3245	044224				ERRHRD ERRNO,WRTMSG,SFIMSG	:WRITE CHARACTERISTICSC FAILED		
	044224	104456					TRAP	C#ERHRD
	044226	002132					.WORD	1114
	044230	005054					.WORD	WRTMSG
	044232	012034					.WORD	SFIMSG
3246	044234			48:				
3247	044234	004737	045514		JSR PC,T11REST	:SET UP PACKET FOR COMMAND		
3248	044240	012704	044640		MOV @T11PACKET,R4	:GET THE ADDRESS OF COMMAND PACKET		
3249	044244			56:				
3250	044244	005037	002222	108:	CLR INTRECV	:CLEAR INTERRUPT RECEIVED FLAG		
3251	044250	010465	000000		MOV R4,TSDB(R5)	:SET THE PACKET ADDRESS		
3252	044254	004737	016336		JSR PC,CHKTSSR	:WAIT FOR SSR TO SET		
3253	044260	103405			BCS 158	:BR IF CARRY SET (GOOD RETURN)		
3254	044262	010001			MOV R0,R1	:SAVE CONTENTS OF TSSR		
3258	044264				ERRDF ERRNO,T11SR2,PKTSSR	:DEVICE FATAL SSR FAILED TO SET		
	044264	104455					TRAP	C#ERDF
	044266	002133					.WORD	1115
	044270	045260					.WORD	T11SR2
	044272	012046					.WORD	PKTSSR
3259	044274			158:	CKLOOP	:LOOP ON ERROR, IF FLAG SET		
	044274	104406					TRAP	C#CLP1
3260	044276				ESCAPE SEG	:BY-PASS SUBTEST IF FATAL ERROR		
	044276	104410					TRAP	C#ESCAPE
	044300	000074					.WORD	100008-
3261	044302	005737	002222		TST INTRECV	:DID AN INTERRUPT OCCUR ?		
3262	044306	001004			BNE 228	:BRANCH IF YES		
3266	044310				ERRHRD ERRNO,T11NINT,PKTSSR			
	044310	104456					TRAP	C#ERHRD
	044312	002134					.WORD	1116
	044314	045334					.WORD	T11NINT
	044316	012046					.WORD	PKTSSR
3267	044320	016501	000002	228:	MOV TSSR(R5),R1	:GET THE CONTENTS OF TSSR		
3268	044324	012702	000200		MOV @SSR,R2	:EXPECTED CONTENTS OF TSSR		
3269	044330	032701	000100		BIT @OFL,R1	:IS OFF-LINE BIT SET ?		
3270	044334	001402			BEQ 258	:BRANCH IF NOT OFF-LINE		
3271	044336	052702	000100		BIS @OFL,R2	:SET OFF-LINE IN EXPECTED DATA		
3272	044342	020201		258:	CMP R2,R1	:DOES EXPECTED MATCH RECEIVED ?		
3273	044344	001404			BEQ 308	:OKAY IF MATCH		
3277	044346				ERRHRD ERRNO,T113REJ,PKTSSR	:COMMAND NOT ACCEPTED		
	044346	104456					TRAP	C#ERHRD
	044350	002135					.WORD	1117
	044352	045073					.WORD	T113REJ
	044354	012046					.WORD	PKTSSR
3278	044356			308:				
3279	044356	004737	011114	358:	JSR PC,CKRAM	:CHECK RAM TO MEMORY		
3280	044362	103405			BCS 598	:RAM OK GO ON		
3284	044364				ERRHRD ERRNO,PKTRAM,RAMERR	:THEY DON'T MATCH		
	044364	104456					TRAP	C#ERHRD

TSVSA - HARDWARE TESTS MACRO M1113 14-JUN-84 15:15
 TEST 11: NON-TAPE MOTION COMMANDS

SEQ 0173

```

3374 044642 044650          .WORD  T11DATA          ;ADDRESS OF CHARACTERISTICS BLOCK
3375 044644 000000          .WORD  0
3376 044646 000010          .WORD  8.          ;STARTING VALUE OF BLOCK SIZE
3377
3378 044650          T11DATA:          ;CHARACTERISTICS DATA BLOCK
3379 044650 044662          .WORD  T11BFR          ;ADDRESS OF MESSAGE BUFFER
3380 044652 000000          .WORD  0
3381 044654 000016          .WORD  14.          ;LENGTH OF MESSAGE BUFFER
3382 044656 000000 000000          .WORD  0,0
3383
3384 044662          T11BFR: .BLKW  8.          ;MESSAGE BUFFER
3385
3386
3388 044702          .BLKB  10-<.-TSV2&7>
3390 044710          T11PK2:          ;COMMAND PACKET FOR TEST
3391 044710 100204          .WORD  100204          ;WRITE CHAR COMMAND, WITH IE, ACK
3392 044712 044720          .WORD  T11DTA          ;ADDRESS OF CHARACTERISTICS BLOCK
3393 044714 000000          .WORD  0
3394 044716 000010          .WORD  8.          ;STARTING VALUE OF BLOCK SIZE
3395
3396 044720          T11DTA:          ;CHARACTERISTICS DATA BLOCK
3397 044720 044732          .WORD  T11BF2          ;ADDRESS OF MESSAGE BUFFER
3398 044722 000000          .WORD  0
3399 044724 000016          .WORD  14.          ;LENGTH OF MESSAGE BUFFER
3400 044726 000000 000000          .WORD  0,0
3401
3402 044732          T11BF2: .BLKW  8.          ;MESSAGE BUFFER
3403
3404
3405
3406
3407          ;*
3408          ;LOCAL TEXT MESSAGES FOR TEST
3409          ;-
3410 044752          111          116          111  T11NBA: .ASCIZ  'INITIALIZE Command Not Accepted'
3411 045012          111          116          111  T112REJ: .ASCIZ  'INITIALIZE Not Rejected With Non-Zero Mode Field'
3412 045073          107          105          124  T113REJ: .ASCIZ  'GET STATUS Not Accepted'
3413 045123          107          105          124  T114REJ: .ASCIZ  'GET STATUS Not Rejected With Non-Zero Mode Field'
3414 045204          103          157          156  T11SSR: .ASCIZ  'Contents of TSSR Incorrect After INITIALIZE'
3415 045260          103          157          156  T11SR2: .ASCIZ  'Contents of TSSR Incorrect After GET STATUS'
3416 045334          105          170          160  T11NINT: .ASCIZ  'Expected Interrupt Not Received On INITIALIZE'
3417 045412          111          156          143  T11TSBA: .ASCIZ  'Incorrect TSBA Address After INITIALIZE'
3418 045462          116          157          156  TST11ID: .ASCIZ  'Non-Tape Motion Commands'
3419          .EVEN
3420
3421
3422          ;*
3423          ;
3424          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
3425          ;INITIALIZE COMMAND
3426          ;
3427          ;-
3428
3429 045514          T11REST:
3430 045514          SAVREG          ;SAVE THE REGISTERS
3431 045520 012701 044640          MOV          #T11PACKET,R1          ;START OF THE PACKET
3432 045524 012721 100213          MOV          #100213,(R1)*          ;INITIALIZE WITH ACK, IE

```

TSV5A - HARDWARE TESTS MACRO M1113 14-JUN-84 15:15
 TEST 11: NON-TAPE MOTION COMMANDS

SEQ 0174

```

3433 045530 005021          CLR      (R1)+      ;ADDRESS OF CHAR DATA BLOCK
3434 045532 005021          CLR      (R1)+      ;EXTENDED ADDRESS
3435 045534 005021          CLR      (R1)+      ;SIZE OF DATA BLOCK IN BYTES
3436 045536 005021          CLR      (R1)+      ;ADDRESS OF MESSAGE BUFFER
3437 045540 005021          CLR      (R1)+
3438 045542 005021          CLR      (R1)+      ;LENGTH OF MESSAGE BUFFER
3439 045544 005021          CLR      (R1)+
3440 045546 005011          CLR      (R1)
3441 045550 005037 044662   CLR      T11BFR     ;CLEAR 1ST LOC IN MESSAGE BUFFER
3442 045554 000207          RTS        PC        ;RETURN
3443
3444
3445
3446
3447
3448
3449
3450 045556
3451 045556
3452 045562 012701 044640   SAVREG
3453 045566 012721 100217   MOV      @T11PACKET,R1 ;SAVE THE REGISTERS
3454 045572 005021          MOV      @100217,(R1)+ ;START OF THE PACKET
3455 045574 005021          CLR      (R1)+      ;GET STATUS WITH ACK, IE
3456 045576 005021          CLR      (R1)+      ;ADDRESS OF CHAR DATA BLOCK
3457 045600 005021          CLR      (R1)+      ;EXTENDED ADDRESS
3458 045602 005021          CLR      (R1)+      ;SIZE OF DATA BLOCK IN BYTES
3459 045604 005021          CLR      (R1)+      ;ADDRESS OF MESSAGE BUFFER
3460 045606 005021          CLR      (R1)+
3461 045610 005011          CLR      (R1)+      ;LENGTH OF MESSAGE BUFFER
3462 045612 005037 044662   CLR      T11BFR     ;CLEAR 1ST LOC IN MESSAGE BUFFER
3463 045616 000207          RTS        PC        ;RETURN
3464 045620
3465 045622 104401          ENDTST
                                L10102: TRAP C#ETST
                                ENDMOD

```

TSV6 - PARAMETER CODING MACRO M1113 14-JUN-84 15:15
 TEST 11: NON-TAPE MOTION COMMANDS

SEQ 0175

```

1          .TITLE  TSV6 - PARAMETER CODING
7
12
18
19 045622      BGNMOD  TSV6
   045622      TSV6::
20
21          .SBTTL  HARDWARE PARAMETER CODING SECTION
22
23          ;**
24          ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
25          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
26          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
27          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
28          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
29          ; WITH THE OPERATOR.
30          ;--
31 045622      BGNHRD
   045622      .WORD  L10107-L#HARD/2
   045624      L#HARD::
32
33 045624      GPRMA  HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
   045624      .WORD  T#CODE
   045626      .WORD  HPM1
   045630      .WORD  T#LLOLIM
   045632      .WORD  T#HILIM
34 045634      GPRMA  HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
   045634      .WORD  T#CODE
   045636      .WORD  HPM2
   045640      .WORD  T#LLOLIM
   045642      .WORD  T#HILIM
35          ;GPRMD  HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
36 045644      ENDHRD
   045644      .EVEN
37 045644      L10107:
   104      105      126  HPM1:  .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
38 045700      111      116      124  HPM2:  .ASCIZ  'INTERRUPT VECTOR '
39 045724      111      116      124  HPM3:  .ASCIZ  'INTERRUPT PRIORITY '
40          .EVEN

```



```

42          .SBTTL  SOFTWARE PARAMETER CODING SECTION
43
44          ;**
45          ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
46          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
47          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
48          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
49          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
50          ; WITH THE OPERATOR.
51          ;--
52 045754      BGNSFT
53 045754      .WORD L10110-L#SOFT/2
54 045756      000003
55          L#SOFT::
56          ;      GPRML  SPM1,0,-1,YES          ; GET TRANSPORT TEST FLAG.
57          ;      GPRML  SPM4,2,-1,YES         ; GET ITERATION CONTROL.
58          ;      .WORD  T#CODE
59          ;      .WORD  SPM4
60          ;      .WORD  -1
61          ;      GPRMD  SPM6,4,D,7777,0,7777,YES ; GET LOCAL ERROR LIMIT
62          ;      GPRMD  SPM7,6,D,7777,0,7777,YES ; GET GLOBAL ERROR LIMIT
63          ENDSFT
64          .EVEN
65          L10110:
66          58 045764      105      116      101  SPM1:  .ASCIZ  'ENABLE TRANSPORT TESTS '
67          59 046014      111      116      110  SPM4:  .ASCIZ  'INHIBIT ITERATIONS '
68          60 046044      120      105      122  SPM6:  .ASCIZ  'PER TEST ERROR LIMIT '
69          61 046074      120      105      122  SPM7:  .ASCIZ  'PER UNIT ERROR LIMIT '
70          62          .SBTTL  PATCH AREA
71          ;
72          ; FINALLY A GENEROUS PATCH AREA.
73          ;
74          ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
75          ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
76          ;
77          PATCH::
78          .BLKW  32.
79          .=.!377*1
80          LASTAD          ;SET LAST USED ADDRESS.
81          .EVEN
82          .WORD  0
83          .WORD  0
84          L#LAST::
85          ENDMOD
86          .END

```

SYMBOL TABLE

ADSSR	012126	G	C#AU	=	000G52	DEVDR0	023332	FRESIZ	003124	G	INTFLA	016145			
ADR	=	000020	C#AUTO	=	000061	DEVNRD	023251	FUSI	004115		INTMAS	016144			
AMBTSS	006633		C#BRK	=	000022	DEVNXR	023167	F#AU	=	000015	INTR	016216	G		
ASSEMB	=	000010	C#BSEG	=	000004	DEVONL	023117	F#AUTO	=	000020	INTREC	002222	G		
A1716	=	000003	C#BSUB	=	000002	DEVSUM	023062	F#BGN	=	000040	INTVEC	016146			
BADDAT	003154	G	C#CEFG	=	000045	DFPTBL	002154	F#CLEA	=	000007	INTX	004276			
BADSSR	015700	G	C#CLCK	=	000062	DIAGMC	=	000000	F#DU	=	000016	INVERT	021142	G	
BDVPCR	=	177520	C#CLEA	=	000012	DICEA	=	000001	F#END	=	000041	IOKCKI	=	000200	
BENBSW	002226	G	C#CLOS	=	000035	DSBINT	016204	F#HARD	=	000004	IOKSTP	=	000001		
BIE	=	040000	C#CLP1	=	000006	DUAD12	004641	F#HW	=	000013	IPRI	002210	G		
BIT0	=	000001	C#CVEC	=	000036	DUFLG	003110	F#INIT	=	000006	ISR	=	000100	G	
BIT00	=	000001	C#DCLN	=	000044	DUMMY	003060	F#JMP	=	000050	IVEC	002206	G		
BIT01	=	000002	C#DODU	=	000051	EF.CON	=	000036	F#MOD	=	000000	IXE	=	004000	G
BIT02	=	000004	C#DRPT	=	000024	EF.NEW	=	000035	F#MSG	=	000011	I#AU	=	000041	
BIT03	=	000010	C#DU	=	000053	EF.PWR	=	000034	F#PROT	=	000021	I#AUTO	=	000041	
BIT04	=	000020	C#EDIT	=	000003	EF.RES	=	000037	F#PWR	=	000017	I#CLN	=	000041	
BIT05	=	000040	C#ERDF	=	000055	EF.STA	=	000040	F#RPT	=	000012	I#DU	=	000041	
BIT06	=	000100	C#ERHR	=	000056	EMAXDU	016777	F#SEG	=	000003	I#HRD	=	000041		
BIT07	=	000200	C#ERRO	=	000060	EN	=	000000	F#SOFT	=	000005	I#INIT	=	000041	
BIT08	=	000400	C#ERSF	=	000051	ENAINT	016152	F#SRV	=	000010	I#MOD	=	000041		
BIT09	=	001000	C#ERSO	=	000057	ENVIRN	020630	F#SUB	=	000002	I#MSG	=	000041		
BIT1	=	000002	C#ESCA	=	000010	EPRTSW	002176	F#SW	=	000014	I#PROT	=	000040		
BIT10	=	002000	C#ESEG	=	000005	EPRT1	006356	F#TEST	=	000001	I#PTAB	=	000041		
BIT11	=	004000	C#ESUB	=	000003	EPRT2	006356	GDDAT	003156	G	I#PWR	=	000041		
BIT12	=	010000	C#ETST	=	000001	ERCM	011733	GERRMA	002172	G	I#RPT	=	000041		
BIT13	=	020000	C#EXIT	=	000032	ERRHI	002234	GETPAT	020174	G	I#SEG	=	000041		
BIT14	=	040000	C#GETB	=	000026	ERRK	016756	GETSEL	020256	G	I#SETU	=	000041		
BIT15	=	100000	C#GETW	=	000027	ERRLO	002236	G#CNT0	=	000200	I#SFT	=	000041		
BIT2	=	000004	C#GMAN	=	000043	ERRNO	=	002144	G#DELM	=	000372	I#SRV	=	000041	
BIT3	=	000010	C#GPHR	=	000042	ERRVEC	=	000004	G#DISP	=	000003	I#SUB	=	000041	
BIT4	=	000020	C#GPLO	=	000030	ERTABE	003374	G#EXCP	=	000400	I#TST	=	000041		
BIT5	=	000040	C#GPRI	=	000040	ERTABL	003174	G#HILI	=	000002	J#JMP	=	000167		
BIT6	=	000100	C#INIT	=	000011	ESUM	016760	G#LOLI	=	000001	KIPAR0	=	172340		
BIT7	=	000200	C#INLP	=	000020	EVL	=	000004	G#NO	=	000000	KIPAR1	=	172342	
BIT8	=	000400	C#MANI	=	000050	EXBCNT	=	000010	G#OFFS	=	000400	KIPAR2	=	172344	
BIT9	=	001000	C#MEM	=	000031	EXPBRE	015502	G#OFSI	=	000376	KIPAR3	=	172346		
BOE	=	000400	C#MSG	=	000023	EXPD	002230	G#PRMA	=	000001	KIPAR4	=	172350		
BRINIT	004455		C#OPEN	=	000034	EXPGOT	004531	G#PRMD	=	000002	KIPAR5	=	172352		
BSELO	=	000000	C#PNTB	=	000014	EXPGT2	004565	G#PRML	=	000000	KIPAR6	=	172354		
BSEL1	=	000001	C#PNTF	=	000017	EXPMSG	002320	G#RADA	=	000140	KIPAR7	=	172356		
CHKAMB	016044		C#PNTS	=	000016	EXPREC	015474	G#RADB	=	000000	KIPDR0	=	172300		
CHKMAN	020500	G	C#PNTX	=	000015	EXTA	005770	G#RADD	=	000040	KIPDR1	=	172302		
CHKTSS	016336		C#QIO	=	000377	EXTEND	005766	G#RADL	=	000120	KIPDR2	=	172304		
CKDROP	017202		C#RDBU	=	000007	EXTFEA	002224	G#RADO	=	000020	KIPDR3	=	172306		
CKEMAX	017102		C#REFG	=	000047	E#END	=	002100	G#XFER	=	000004	KIPDR4	=	172310	
CKMSG	011360	G	C#RESE	=	000033	E#LOAD	=	000035	G#YES	=	000010	KIPDR5	=	172312	
CKMSG2	011500	G	C#REVI	=	000003	FATERR	=	000060	HIADDR	=	001400	KIPDR6	=	172314	
CKRAM	011114	G	C#RFLA	=	000021	FATFLG	002220	HOE	=	100000	G	KIPDR7	=	172316	
CKRAM2	011224	G	C#RPT	=	000025	FERCM	011722	HPM1	045644		KTENAB	003132	G		
CHDPKT	021214	G	C#SEFG	=	000046	FIFEXP	012170	HPM2	045700		KTFLG	003130	G		
CHPMEM	017660		C#SPRI	=	000041	FIF1MS	012242	HPM3	045724		KTINIT	021010			
CONFIG	017250		C#SVEC	=	000037	FIF2MS	012311	IBE	=	010000	G	KTOFF	017274		
COUNT	002306	G	C#TPRI	=	000013	FILLME	017422	IDU	=	000040	G	KTON	017256		
CSRADD	002204	G	DATA	002310	G	FNOINT	004213	IER	=	020000	G	LERRMA	002170	G	
CTAB	003162	G	DATASC	020232		FORCER	002174	IFALT	004254		LISTAL	=	000001		
CTABE	003174	G	DEBUGM	011632		FREE	003122	INCERK	017044		LOE	=	040000	G	
CTABM	003162	G	DEVCNT	002216	G	FREEHI	003126	INTCPC	016150		LOOPCN	002214	G		

LOOPCO	013126	L10001	002174	L10073	036746	NXR	003736	PRI05	= 000240 G
LOOPFL	003160 G	L10002	005764	L10074	037152	NXRERR	005734 G	PRI06	= 000300 G
LOT	= 000010 G	L10003	012044	L10075	043452	NXRX	003775	PRI07	= 000340 G
L\$ACP	002110 G	L10004	012062	L10076	040742	NXTU	021756	PRMESS	014242
L\$APT	002036 G	L10005	012100	L10077	041334	OFL	= 000100	PRMNO	C02316 G
L\$AU	022306 G	L10006	012106	L10100	042224	ONEFIL	= 000000	PRMSGE	014552 G
L\$AUT	002070 G	L10007	012124	L10101	042452	O#APTS	= 000000	PRMSG0	014732
L\$AUTO	022512 G	L10010	012142	L10102	045620	O#AU	= 000001	PRMSG1	014777
L\$CCP	002106 G	L10011	012156	L10103	043720	O#BGNR	= 000001	PRMSG2	015035
L\$CLEA	022572 G	L10012	012240	L10104	044156	O#BGNS	= 000001	PROFSC	014420
L\$CO	002032 G	L10013	012410	L10105	044376	O#DU	= 000001	PR1ASC	014465
L\$DEPO	002011 G	L10014	013124	L10106	044622	O#ERRT	= 000000	PST32W	003150 G
L\$DESC	003406 G	L10015	013752	L10107	045644	O#GNSW	= 000001	PUNIT	022240
L\$DESP	002076 G	L10016	013774	L10110	045764	O#POIN	= 000001	PW.D11	= 000021
L\$DEVP	002060 G	L10017	015500	MEMADD	013754 G	O#SETU	= 000000	PW.D13	= 000022
L\$DISP	002124 G	L10020	015506	MEMCK	021232 G	PASRPT	022010	PW.D22	= 000020
L\$DLY	002116 G	L10021	015514	MENASC	020447	PATCH	046124 G	PW.NOP	= 000000
L\$DTP	002040 G	L10022	015526	MENERR	020374	PATDAT	020230	PW.NO1	= 000023
L\$DTYP	002034 G	L10023	015550	MENRES	020476	PC.ERA	= 002400	PW.RDE	= 000024
L\$DU	022404 G	L10024	015576	M#VEC	= 000250	PC.IER	= 002000	PW.RDR	= 000001
L\$DUT	002072 G	L10025	015736	MSA.FR	= 000006	PC.NOD	= 001000	PW.RDS	= 000005
L\$DVTY	003400 G	L10026	016246	MSA.NO	= 000000	PC.REL	= 000000	PW.RFI	= 000003
L\$EF	002052 G	L10030	022236	MSA.NR	= 000004	PC.REW	= 000400	PW.WCT	= 000006
L\$ENVI	002044 G	L10031	022402	MSA.VO	= 000002	PKBCNT	= 000006	PW.WFI	= 000004
L\$ETP	002102 G	L10032	022510	MSGEXP	012144 G	PKHI	= 000004	PW.WFM	= 000007
L\$EXP1	002046 G	L10033	022570	MSGLOO	013064 G	PKLOW	= 000002	PW.WMI	= 000010
L\$EXP4	002064 G	L10034	022616	MSGSTA	012350 G	PKTADD	007552	PW.WNP	= 000011
L\$EXP5	002066 G	L10035	023060	MSGSUB	013742 G	PKTFRM	007514	PW.WTR	= 000002
L\$HARD	045624 G	L10036	023620	MS.ATT	= 000006	PKTGET	012064 G	P.ACK	= 100000
L\$HIME	002120 G	L10037	023466	MS.EXT	= 000200	PKTMES	012110 G	P.CMD	= 000037
L\$HPCP	002016 G	L10040	023550	MS.RSD	= 000001	PKTRAM	004743 G	P.CONT	= 000012
L\$HPTP	002022 G	L10041	024316	MS.RSF	= 000020	PKTSSR	012046 G	P.CVC	= 040000
L\$HW	002154 G	L10042	025010	MS.RST	= 000010	PNT	= 001000 G	P.FMT	= 000140
L\$ICP	002104 G	L10043	026344	M2901	026070	PRAMPK	013776	P.FORM	= 000011
L\$INIT	021512 G	L10044	025252	M8186	005552	PRASC	014523	P.GETS	= 000017
L\$LADP	002026 G	L10045	025552	M8189	005643	PRBEXP	015470	P.IE	= 000200
L\$LAST	046404 G	L10046	026050	NBA	= 002000	PRBMSG	015336	P.INIT	= 000013
L\$LOAD	002100 G	L10047	027450	NEMPAS	021744	PRBREC	015472	P.MODE	= 007400
L\$LUN	002074 G	L10050	026716	NODEV	003112 G	PRBTOT	015423	P.OPP	= 020000
L\$MREV	002050 G	L10051	027266	NOINIT	004333	PRBYTE	015122 G	P.POSI	= 000010
L\$NAME	002000 G	L10052	030722	NOINTR	004217	PRI	= 002000 G	P.READ	= 000001
L\$PRIO	002042 G	L10053	030000	NOITS	002166 G	PRIADD	010156	P.SWB	= 010000
L\$PRT	002112 G	L10054	030324	NOMAN	020534	PRIAO	010226	P.WRIT	= 000005
L\$REPP	002062 G	L10055	034310	NOMEM	005456	PRIBXO	007610 G	P.WRTC	= 000004
L\$REV	002010 G	L10056	031254	NP.IR	= 000200	PRIEQU	010056	P.WRTS	= 000006
L\$RPT	022620 G	L10057	031540	NP.LOO	= 000040	PRIPKT	007366 G	QVP	002202 G
L\$SOFT	045756 G	L10060	032004	NP.OUT	= 000100	PRIRAM	010064	RAMASC	014156
L\$SPC	002056 G	L10061	032232	NP.WRP	= 000020	PRITAD	010272	RAMDAT	002240 G
L\$SPCP	002020 G	L10062	032476	NSI	004150	PRITSS	006022	RAMERR	015510 G
L\$SPTP	002024 G	L10063	033036	NSINIT	004405	PRITO	010354	RAMEXP	015530 G
L\$STA	002030 G	L10064	035214	NUL	004525	PRIT1	010417	RAMFOR	010114
L\$SW	002164 G	L10065	040340	NULCR	004526	PRIXOR	007740 G	RAMSIZ	002300 G
L\$TEST	002114 G	L10066	035456	NXM	= 004000	PRI00	= 000000 G	RAMTAD	015516 G
L\$TIML	002014 G	L10067	035716	NXMFLG	003134 G	PRI01	= 000040 G	RCVHIA	002302 G
L\$UNIT	002012 G	L10070	036122	NXMHI	003140 G	PRI02	= 000100 G	RCVLOA	002304 G
L10000	002162	L10071	036310	NXMLO	003136 G	PRI03	= 000140 G	RDERR	005204
		L10072	036514	NXMTST	021406	PRI04	= 000200 G	RECMG	002464 G

RECV	002232	G	S1.IEO=	010000	TST6ID	030703	T1.1	023422	T4	025012	G
REGSAV	020140		S1.IFM=	001000	TST7ID	034213	T1.2	023502	T4LOOP	025032	
RETERR	005370		S1.IHE=	000400	TST8ID	035177	T10	040342	T4.1	025012	
REWIND	011014	G	S1.IID=	004000	TST9ID	040241	T10BFR	042512	T4.2	025254	
RMCHBE=	000167		S1.IIR=	020000	TSV2	002000	T10BUF	042554	T4.3	025554	
RMCHEN=	000200		S1.I2R=	040000	TSV3	002174	T10DAT	042500	T5	026346	G
RMMSGB=	000215		S1.PAR=	100000	TSV4	021502	T10DTA	042542	T5ADDR	027406	
RMMSGC=	000234		S2.ATI=	000010	TSV5	023402	T10INT	043210	T5LOOP	026364	
RMPKTB=	000201		S2.BTI=	000004	TSV6	045622	T10LOO	040360	T5MEM	027350	
RMPKTE=	000210		S2.DIM=	000200	TTIBFR=	177562	T10MBF	042574	T5.1	026370	
RMR	=	010000	S2.ILW=	000100	TTICSR=	177560	T10NBA	042671	T5.2	026736	
RMPACK	011110		S2.INR=	000020	TTIVEC=	000060	T10NIN	043117	T6	027452	G
SC	=	100000	S2.OUT=	000040	T#ARGC=	000003	T10NNB	042753	T6INT	030473	
SCE	=	020000	S2.UND=	000003	T#CODE=	001130	T10PAC	042470	T6LOOP	027470	
SCHERR	005276		TBLEND=	003060	T#ERRN=	002144	T10PKT	042532	T6NBA	030370	
SCME	005011		TCOASC	006474	T#EXCP=	000000	T10RST	043326	T6NINT	030551	
SDELAY	010660		TCOCOD	006674	T#FLAG=	000040	T10RT2	043400	T6PACK	030360	
SELASC	020442		TEMP1	003114	T#GMAN=	000000	T10SSR	043030	T6SSR	030415	
SELDAT=	000004		TEMP2	003116	T#HILI=	000776	T10.1	040360	T6TSBA	030631	
SEL2	=	000002	TERCLS=	000016	T#LAST=	000001	T10.2	040744	T6.1	027470	
SETMAP	017316		TESTNO=	000013	T#LOLI=	000000	T10.3	041336	T6.2	030014	
SETU	022042		TEXASC	006433	T#LSYM=	010000	T10.4	042226	T7	030724	G
SFFMSG	012102	G	TFCASC	006535	T#LTND=	000013	T11	043454	T7BFR	033106	
SFHERR	003703		TIMEXP	015552	T#NEST=	177777	T11BFR	044662	T7DATA	033070	
SFIERR	003650		TIMSGO	015600	T#NSO =	000000	T11BF2	044732	T7INT	034041	
SFIMSG	012034	G	TINERR	012021	T#NS1 =	000005	T11DAT	044650	T7LOOP	030742	
SFPTBL	002164	G	TMPBFR	002630	T#NS2 =	000002	T11DTA	044720	T7NBA	033220	
SIFLAG	003152	G	TNAM	016704	T#NS3 =	000003	T11LOO	043472	T7NINT	033750	
SIMSG	011766		TRANST	002164	T#PTNU=	000000	T11NBA	044752	T7PACK	033060	
SKIPT	003376		TSBA	=	000000	G	T11NIN	045334	T7REST	034242	
SOFINI	015774	G	TSBAH	=	000001	G	T11PAC	044540	T7SP	033100	
SPACE	010464	G	TSBAM2	026160	T#SEGL=	177777	T11PK2	044710	T7SSR	033661	
SPM1	045764		TSBAM3	026242	T#SEKO=	010000	T11RES	045514	T7TSBA	034130	
SPM4	046014		TSDB	=	000000	G	T11RT2	045556	T7.1	030742	
SPM6	046044		TSDBH	=	000001	G	T11SR2	045260	T7.2	031270	
SPM7	046074		TSFCOD	007234	T#TEMP=	000000	T11SSR	045204	T7.3	031542	
SR0	=	177572	TSREJ	=	000006		T11TSB	045412	T7.4	032006	
SR1	=	177574	TSSDEF	006604	T#TEST=	000013	T1.1.1	043472	T7.5	032234	
SR2	=	177576	TSSR	=	000002	G	T11.2	043734	T7.6	032500	
SR3	=	172516	TSSRBI	003500	T#TSTS=	000001	T11.3	044160	T72DAT	033126	
SSR	=	000200	TSSRFO	006413	T#AU =	010031	T11.4	044400	T72DON=	033142	
STATCO	012412		TSSRH	=	000003	G	T112RE	045012	T72NBA	033142	
SVCGBL=	000000		TSSX	004016	T#CLE=	010034	T113RE	045073	T72REJ	033273	
SVCINS=	000000		TSTBLK	002750	T#DU =	010032	T114RE	045123	T73REJ	033372	
SVCSUB=	000001		TSTCNT	002212	T#HAR=	010107	T2	023622	T74REJ	033465	
SVCTAG=	000000		TSTEND	016720	T#HW =	010000	T2LOOP	023640	T75REJ	033563	
SVCTST=	000001		TSTFLA	002312	T#INI=	010030	T2SSR	024216	T8	034312	G
S#LSYM=	010000		TSTL00	016456	T#MSG=	010025	T2TSBA	024104	T8BFR	034742	
SO.IDB=	000010		TSTPTR	002314	T#PRO=	010027	T2TSSR	024151	T8DATA	034730	
SO.IFB=	000002		TSTSET	016510	T#RPT=	010035	T23A	003142	T8LOOP	034330	
SO.IFP=	000001		TST1ID	023600	T#SEG=	010000	T23B	003144	T8NVCK	035017	
SO.ILD=	000020		TST10I	043277	T#SOF=	010110	T3	024320	T8PACK	034720	
SO.ION=	000040		TST11I	045462	T#SRV=	010026	T3BFLG	003146	T8SSR	035110	
SO.IRD=	000100		TST2ID	024270	T#SUB=	010106	T3LOOP	024336	T8VCK	034762	
SO.IRW=	000004		TST3ID	024763	T#SW =	010001	T3SSR	024712	T9	035216	G
SO.ISP=	000200		TST4ID	026322	T#TES=	010102	T3TSBA	024602	T9BFR	037212	
S1.ICE=	002000		TST5ID	027320	T1	023402	T3TSSR	024646	T9DATA	037200	
					T1LOOP	023420					

TSV6 - PARAMETER CODING MACRO M1113 14-JUN-84 15:15
 SYMBOL TABLE

SEQ 0180

T9INT	040067	UAM	= 000200 G	WRTCHR	010662 G	XSOONL	= 000100	X2.EXT	= 000200
T9LOOP	035240	UNITN	= 002200 G	WRTERR	005111	XSOPEL	= 000010	X2.OPM	= 100000
T9NBA	037246	UNREC	= 000006	WRTMSG	005054	XSORLL	= 010000	X2.RCE	= 040000
T9NINT	037776	USI	= 004121	WSMBK	021224 G	XSORLS	= 040000	X2.REV	= 000077
T9PACK	037170	WAITF	= 016250 G	XFERAS	015740	XSOYMK	= 100000	X2.SPA	= 035400
T9REST	040266	WC.IFA	= 000200	XNXM	016376	XSOVCK	= 000020	X2.UNI	= 000007
T9SSR	037707	WC.IFE	= 000002	XORBFO	007672	XSOVLE	= 004000	X2.WCF	= 002000
T9TSBA	040156	WC.IGD	= 000001	XORFOR	010010	XSOVLK	= 000004	X3.DCK	= 000010
T9.1	035240	WC.IRE	= 000010	XST0	= 000006 G	XXCOMM	003120 G	X3.MBZ	= 000006
T9.2	035506	WC.IRW	= 000004	XST1	= 000010 G	X#ALWA	= 000000	X3.MDE	= 177400
T9.3	035720	WC.IOT	= 000100	XST2	= 000012 G	X#FALS	= 000040	X3.OPI	= 000100
T9.4	036124	WC.IIT	= 000040	XST3	= 000014 G	X#OFFS	= 000400	X3.REV	= 000040
T9.5	036312	WC.ISR	= 000020	XST4	= 000016 G	X#TRUE	= 000020	X3.RIB	= 000001
T9.6	036516	WF.IED	= 000010	XSOBOT	= 000002	X1.COR	= 020000	X3.SPA	= 000200
T9.7	036750	WF.IER	= 000004	XSOEOT	= 000001	X1.DLT	= 100000	X3.TRF	= 000020
T92DAT	037232	WF.IHI	= 000200	XSOIE	= 000040	X1.MBZ	= 017375	X4.HSP	= 100000
T92DON	037246	WF.IRE	= 000040	XSOILA	= 000400	X1.RBP	= 000400	X4.MBZ	= 017400
T92REJ	037321	WF.IWF	= 000020	XSOILC	= 001000	X1.SPA	= 040000	X4.RCE	= 040000
T93REJ	037420	WF.IWR	= 000100	XSOLET	= 020000	X1.UNC	= 000002	X4.TSM	= 020000
T94REJ	037513	WF.I3R	= 000002	XSOLOT	= 000200	X2.BUF	= 000100	X4.WRC	= 000377
T95REJ	037611	WF.I4R	= 000001	XSONEF	= 002000				

. ABS. 046404 000
 000000 001
 ABS 000000 002
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 28224 WORDS (111 PAGES)
 DYNAMIC MEMORY: 20614 WORDS (79 PAGES)
 ELAPSED TIME: 00:39:38
 CVTSAB, CVTSAB/-SP=SVC/ML, TSV1A, TSV22A, TSV3B, TSV4, TSV5A, TSV6