

TSV05

TSV05 CTRL PT 1
CVTSAC0

AH-T094C-MC

1 OF 1 SEP 1987

COPYRIGHT© 1982-87

digital

MADE IN USA

.REM_
IDENTIFICATION

PRODUCT ID: AC-T093C-MC
PRODUCT TITLE: CVTSACO TSV05 CTRL PART 1
DECO/DEPO: 1.0
DEPARTMENT: COMPUTER SPECIAL SYSTEMS/PGG
DATE: JUNE 4, 1987

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

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

COPYRIGHT (C) 1983, 1987 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

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 (HSAAS.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 HIERARCHY 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 TS05 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-C-0
****TSV05 LOGIC DIAGNOSTIC****
UNIT IS TSV05
>DR
```


2.2 SWITCHES

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

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

EXAMPLE OF SWITCH USAGE:

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

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

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

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

2.3 FLAGS

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

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

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

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

/FLAGS:LOE:IER:BOE

2.4 HARDWARE QUESTIONS

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

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

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

TSBA/TSDB = 172520, VECTOR = 224

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

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

UNIT 0

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

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

THE ADDRESS AND VECTOR QUESTIONS WILL BE ASKED FOR EACH OF THE NUMBER OF UNITS (CONTROLLERS) SPECIFIED IN THE "# UNITS?" QUESTION. LOGICAL UNIT NUMBERS ARE ASSIGNED IN ORDER, BEGINNING AT 0. UP TO FOUR UNITS CAN BE SELECTED FOR TESTING AS FOLLOWS:
UP TO 4 TSV05 CONTROLLERS PER 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 (D) ? 8<CR>

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

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

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

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

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

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

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

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

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

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

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

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

UNIT 1
CSR ADDRESS (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 HRD ERR 01610 ON UNIT 00 TST 016 SUB 002 PC: 040624
FIFO STATUS (IN WORD 9) INCORRECT AFTER WRITE FIFO

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

TAPE BUS SIGNALS IN WORD #9:
DATMIS<7> ILW<6> OUTRDY<5> INRDY<4>

MESSAGE BUFFER ADDRESS = 047352

MESSAGE BUFFER CONTENTS:

WORD #	EXPD:	RECV:	XOR:
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 HRD ERR 00159 ON UNIT 00 TST 001 SUB 005 PC: 026202
TSSR NOT CORRECT AFTER SPACE RECORDS COMMAND

TSSR = 100214

TSSR BITS SET: SC,SSR

TERMINATION CLASS CODE = UNRECOVERABLE ERROR

PACKET ADDRESS = 026420

PACKET WORD # = 140010

PACKET WORD # = 000010

PACKET WORD # = 000000

PACKET WORD # = 000024

ERROR MESSAGE EXAMPLE 3

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

CVTS HRD ERR 00121 ON UNIT 00 TST 001 SUB 002 PC: 023306
MOT BIT (XST0) NOT SET DURING REWIND (EXTENDED FEATURES MODE)
EXPD: 000312 RECV: 000112 XOR: 000200

4.0 PERFORMANCE AND PROGRESS REPORTS

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

SUCCESSFUL RUN EXAMPLE (LSI-11)

DR>STA/FLA:PNT:HOE

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 DC005 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.

REVISION C - APRIL 1987

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

J2

```

2      .TITLE   TSV2 - PROGRAM HEADER
3      .SBTTL   PROGRAM HEADER
4      .PSECT   ABS
5      000000
11     .MCALL    SVC
12     000000      SVC          ; INITIALIZE SUPERVISOR MACROS
13     .ENABLE   LC
14     .NLIST    BEX,CND
20     000000      .ENABL    ABS,AMA
21     002000      .-2000
22     002000      BGNMOD    TSV2
23     002000
24     TSV2::
25     ;**
26     ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
27     ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
28     ;--
29
30     002000      POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
31     002000      HEADER CVTSA,C,0,655.,0
      002000      L$NAME::      ;DIAGNOSTIC NAME
      002000      .ASCII /C/
      002001      103      .ASCII /V/
      002002      126      .ASCII /T/
      002003      124      .ASCII /S/
      002004      123      .ASCII /A/
      002005      101      .BYTE 0
      002006      000      .BYTE 0
      002007      000      .BYTE 0
      002010      L$REV::      ;REVISION LEVEL
      002010      103      .ASCII /C/
      002011      L$DEPO::      ;0
      002011      060      .ASCII /0/
      002012      L$UNIT::      ;NUMBER OF UNITS
      002012      000000      .WORD 0
      002014      L$TIML::      ;LONGEST TEST TIME
      002014      001217      .WORD 655.
      002016      L$HPCP::      ;POINTER TO H.W. QUES.
      002016      046004      .WORD L$HARD
      002020      L$SPCP::      ;POINTER TO S.W. QUES.
      002020      046136      .WORD L$SOFT
      002022      L$HPTP::      ;PTR. TO DEF. H.W. PTABLE
      002022      002154      .WORD L$HW
      002024      L$SPTP::      ;PTR. TO S.W. PTABLE
      002024      002164      .WORD L$SW
      002026      L$LADP::      ;DIAG. END ADDRESS
      002026      047004      .WORD L$LAST
      002030      L$STA::      ;RESERVED FOR APT STATS
      002030      000000      .WORD 0
      002032      L$CO::      .WORD 0
      002032      000000      .WORD 0
      002034      L$DTYP::      ;DIAGNOSTIC TYPE
      002034      000000      .WORD 0
      002036      L$APT::      ;APT EXPANSION
      002036      000000      .WORD 0
      002040      L$DTP::      ;PTR. TO DISPATCH TABLE

```


PROGRAM HEADER

002040	002124			
002042		L\$PRIO::	.WORD	L\$DISPATCH
002042	000000			;DIAGNOSTIC RUN PRIORITY
002044		L\$ENVI::	.WORD	0
002044	000000			;FLAGS DESCRIBE HOW IT WAS SETUP
002046		L\$EXP1::	.WORD	0
002046	000000			;EXPANSION WORD
002050		L\$MREV::	.WORD	0
002050	003			;SVC REV AND EDIT #
002051	003		.BYTE	C\$REVISION
002052			.BYTE	C\$EDIT
002052	000000	L\$EF::		;DIAG. EVENT FLAGS
002054	000000		.WORD	0
002056		L\$SPC::	.WORD	0
002056	000000			
002060		L\$DEVP::	.WORD	0
002060	003402			; POINTER TO DEVICE TYPE LIST
002062		L\$REPP::	.WORD	L\$DV TYP
002062	022700			;PTR. TO REPORT CODE
002064		L\$EXP4::	.WORD	L\$RPT
002064	000000			
002066		L\$EXP5::	.WORD	0
002066	000000			
002070		L\$AUT::	.WORD	0
002070	022366			;PTR. TO ADD UNIT CODE
002072		L\$DUT::	.WORD	L\$AU
002072	022464			;PTR. TO DROP UNIT CODE
002074		L\$LUN::	.WORD	L\$DU
002074	000000			;LUN FOR EXERCISERS TO FILL
002076		L\$DESP::	.WORD	0
002076	003410			;POINTER TO DIAG. DESCRIPTION
002100		L\$LOAD::	.WORD	L\$DESC
002100	104035			;GENERATE SPECIAL AUTOLOAD EMT
002102			EMT	E\$LOAD
002102	000000	L\$ETP::		;POINTER TO ERR TBL
002104			.WORD	0
002104	021572	L\$ICP::		;PTR. TO INIT CODE
002106			.WORD	L\$INIT
002106	022652	L\$CCP::		;PTR. TO CLEAN-UP CODE
002110			.WORD	L\$CLEAN
002110	022572	L\$ACP::		;PTR. TO AUTO CODE
002112			.WORD	L\$AUTO
002112	021562	L\$PRT::		;PTR. TO PROTECT TABLE
002114			.WORD	L\$PROT
002114	000000	L\$TEST::		;TEST NUMBER
002116			.WORD	0
002116	000000	L\$DLY::		;DELAY COUNT
002120			.WORD	0
002120	000000	L\$HIME::		;PTR. TO HIGH MEM
			.WORD	0

PROGRAM HEADER

35
36
37
38
39
40
41
42
43

002122
002122 000013
002124
002124 023462
002126 023702
002130 024400
002132 025072
002134 026426
002136 027532
002140 031004
002142 034462
002144 035366
002146 040522
002150 043634

.SBTTL DISPATCH TABLE

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

DISPATCH 11
L\$DISPATCH::
.WORD 11
.WORD T1
.WORD T2
.WORD T3
.WORD T4
.WORD T5
.WORD T6
.WORD T7
.WORD T8
.WORD T9
.WORD T10
.WORD T11

44
45
46
47
48
49
50
51
52
53
54

002152
002152 000003
002154
002154

.SBTTL DEFAULT HARDWARE P-TABLE

; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.

BGNHM DFPTBL ;DEFAULT HARD-P-TABLE
.WORD L10000-L\$HM/2
L\$HM::
DFPTBL::
.WORD 172520 ; 1ST (OF 2) REGISTERS.
.WORD 224 ; INTERRUPT VECTOR
.WORD PRI04 ; INTERRUPT PRIORITY.
ENDHM
L10000:

55
56
57
58
59
60

002154 172520
002156 000224
002160 000200
002162
002162

M2

TSV2 - PROGRAM HEADER MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 26

SEQ 0025

SOFTWARE P-TABLE

```
62                                     .SBTTL  SOFTWARE P-TABLE
63
64
65                                     ;**
66                                     ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
67                                     ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
68                                     ;--
69                                     BGNSW   SFPTBL
70                                     .WORD   L10001-L$SW/2
71                                     L$SW::
72                                     SFPTBL::
73
74                                     TRANSTST::      .WORD   0      ; ENABLE TEST OF TRANSPORT(S) IF =1
75                                     NOITS::          .WORD   0      ; INHIBIT ITERATION OPTION.
76                                     LERRMAX::         .WORD   15.    ; ... 0 = ITERATE.
77                                     GERRMAX::         .WORD   200.   ; ...NZ = INHIBIT ITERATE.
78                                     L10001:          ; LOCAL (PER TEST) ERROR LIMIT
79                                     ENDSW            ; GLOBAL (PER UNIT) ERROR LIMIT
80
81                                     ENDMOD
82
83
84
```

N2

TSV3 - GLOBAL AREAS

MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 27

SEQ 0026

SOFTWARE P-TABLE

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

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

000040
000037
000036
000035
000034

.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

BIT15== 100000
BIT14== 40000
BIT13== 20000
BIT12== 10000
BIT11== 4000
BIT10== 2000
BIT09== 1000
BIT08== 400
BIT07== 200
BIT06== 100
BIT05== 40
BIT04== 20
BIT03== 10
BIT02== 4
BIT01== 2
BIT00== 1

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

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

EF.START== 32.
EF.RESTART== 31.
EF.CONTINUE== 30.
EF.NEW== 29.
EF.PWR== 28.

; START COMMAND WAS ISSUED
; RESTART COMMAND WAS ISSUED
; CONTINUE COMMAND WAS ISSUED
; A NEW PASS HAS BEEN STARTED
; A POWER-FAIL/POWER-UP OCCURRED

; PRIORITY LEVEL DEFINITIONS

B3

ISV3 - GLOBAL AREAS

MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 27-1

SEQ 0027

GLOBAL EQUATES SECTION

```

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

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

```

33
34 002174

```

      .SBTTL      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

C3

TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 27-2

SEQ 0028

MEMORY MANAGEMENT DEFINITIONS

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


D3

TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 27-3

SEQ 0029

MEMORY MANAGEMENT DEFINITIONS

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

E3

TSV05 REGISTER AND PACKET DEFINITIONS

```

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      NXH=      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      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
69      ;(XST0)
70      ;
71      ;-
72
73      100000      XSOTMK= BIT15      ;TAPE MARK DETECTED
74      040000      XSORLS= BIT14      ;RECORD LENGTH SHORT
75      020000      XSOLET= BIT13      ;LOGICAL END OF TAPE
76      010000      XSORLL= BIT12      ;RECORD LENGTH LONG
77      004000      XSOMLE= BIT11      ;WRITE LOCK ERROR
78      002000      XSONEF= BIT10      ;NON EXECUTABLE FUNCTION
79      001000      XS0ILC= BIT9       ;ILLEGAL COMMAND
80      000400      XS0ILA= BIT8       ;ILLEGAL ADDRESS
81      000200      XS0MOT= BIT7       ;TAPE IN MOTION
82      000100      XS0ONL= BIT6       ;TRANSPORT ON LINE
83      000040      XS0IE=  BIT5       ;INTERRUPT ENABLE
84      000020      XS0VCK= BIT4       ;VOLUME CHECK BIT
85      000010      XS0PED= BIT3       ;PHASE ENCODED DRIVE
86      000004      XS0MLK= BIT2       ;WRITE LOCKED
87      000002      XS0BOT= BIT1       ;BEGINNING OF TAPE
88      000001      XS0EOT= BIT0       ;END OF TAPE
89

```


F3

TSV05 REGISTER AND PACKET DEFINITIONS

```

91
92
93
94
95      100000
96      040000
97      020000
98      017375
99      000400
100     000002
101
102
103
104
105
106     100000
107     040000
108     035400
109     002000
110     000200
111     000100
112     000077
113     000007
114
115
116
117
118
119     177400
120     000200
121     000100
122     000040
123     000020
124     000010
125     000006
126     000001
127
128
129
130
131
132     100000
133     040000
134     020000
135     017400
136     000377
137
138
139
140
141
142
143
144     000006
145     000006

```

```

;
;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
;(XST1)
;-
X1.DLT = BIT15      ;DATA LATE
X1.SPARE= BIT14      ;NOT USED
X1.COR = BIT13      ;CORRECTABLE DATA ERROR
X1.MBZ = BIT12-BIT11-BIT10-BIT9-BIT7-BIT6-BIT5-BIT4-BIT3-BIT2-BIT0 ;ALWAYS 0
X1.RBP = BIT8        ;READ BUS PARITY ERROR
X1.UNC = BIT1        ;UNCORRECTABLE DATA OR HARD ERROR

;
;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
;(XST2)
;-
X2.OPM = BIT15      ;OPERATION IN PROGRESS (TAPE MOVING)
X2.RCE = BIT14      ;RAM CHECKSUM ERROR
X2.SPARE= BIT13-BIT12-BIT11-BIT9-BIT8 ;NOT USED BY TSV05 (ALWAYS=0)
X2.WCF = BIT10      ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
X2.EXTF = BIT7       ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
X2.BUFE = BIT6       ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
X2.REV = 000077     ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
X2.UNIT = BIT2-BIT1-BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.

;
;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
;(XST3)
;-
X3.MDE = 177400     ;MICRO-DIAGNOSTIC ERROR CODE
X3.SPARE= BIT7       ;NOT USED BY TSV05
X3.OPI = BIT6       ;OPERATION INCOMPLETE
X3.REV = BIT5       ;REVERSE
X3.TRF = BIT4       ;TRANSPORT RESPONSE FAILURE
X3.DCK = BIT3       ;DENSITY CHECK
X3.MBZ = BIT2-BIT1  ;NOT USED ALWAYS 0
X3.RIB = BIT0       ;REVERSE INTO BOT

;
;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
;(XST4)
;-
X4.HSP = BIT15      ;HIGH SPEED
X4.RCE = BIT14      ;RETRY COUNT EXCEEDED
X4.TSM = BIT13      ;TRANSPORT SPECIAL MODE
X4.MBZ = BIT12-BIT11-BIT10-BIT9-BIT8 ;NOT USED ALWAYS 0
X4.WRC = 000377     ;WRITE RETRY COUNT FIELD

;
;TSSR TERMINATION CODES (BIT 0-2)
;-
TSREJ= 3*2          ;COMMAND REJECTED
UNREC= 6             ;UNRECOVERABLE ERROR

```

G3

TSV05 REGISTER AND PACKET DEFINITIONS

```

147
148
149
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      ;*
162      ; TSDB ADDRESS BIT DEFINITIONS
163      ;
164      000003      A1716 = BIT1-BIT0      ;ADDRESS BITS 17:16 ARE IN 1:0
165
166      ;*
167      ; COMMAND DEFINITIONS
168      ;
169      000017      P.GETSTAT = 17      ;GET STATUS
170      000013      P.INIT = 13      ;INITIALIZE
171      000012      P.CONTROL = 12      ;CONTROL COMMANDS
172      000011      P.FORMAT = 11      ;FORMAT
173      000010      P.POSITION = 10      ;POSITION
174      000006      P.WRTSUB = 6      ;SUBSYSTEM WRITE
175      000005      P.WRITE = 5      ;WRITE
176      000004      P.WRTCHAR = 4      ;WRITE CHARACTERISTICS
177      000001      P.READ = 1      ;READ
178
179      ;*
180      ; COMMAND PACKET HEADER WORD BIT DEFINITIONS
181      ;
182      100000      P.ACK = BIT15      ;BUFFER AVAIL FOR CONTROLLER
183      040000      P.CVC = BIT14      ;CLEAR VOLUME CHECK
184      020000      P.OPP = BIT13      ;REVERSE SEQUENCE OF DATA BITS
185      010000      P.SWB = BIT12      ;SWAP BYTES IN MEMORY
186      007400      P.MODE = BIT11:BIT10:BIT9:BIT8 ;EXTENDED COMMAND MODE FIELD
187      000200      P.IE = BIT7      ;INTERRUPT ENABLE
188      000140      P.FMT= BIT6:BIT5      ;PACKET HEADER TYPE (ALWAYS=0)
189      000037      P.CMD = 37      ;MAJOR COMMAND FIELD
190
191      ;*
192      ; CONTROL COMMAND MODE CODES
193      ;
194      000000      PC.RELEASE = 0*256.      ;RELEASE BUFFER
195      000400      PC.REWIND = 1*256.      ;REWIND
196      001000      PC.NOOP = 2*256.      ;NO-OP
197      002000      PC.IEREW = 4*256.      ;REWIND IMMEDIATE INTERRUPT
198      002400      PC.ERASE = 5*256.      ;SECURITY ERASE

```


H3

TSV05 REGISTER AND PACKET DEFINITIONS

```

198
199
200      ;*
201      ; CONTROLLER RAM DEFINITIONS
202      ;-
203      000167      RMCHBEG = 167      ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
204      000200      RMCHEND = 200      ;CHARACTERISTICS IO DATA END RAM ADDRESS
205      000201      RMPKTBEGBEG = 201      ;COMMAND PACKET BEGIN RAM ADDRESS
206      000210      RMPKTEND = 210      ;COMMAND PACKET END RAM ADDRESS
207      000215      RMMSGBEG = 215      ;MESSAGE BUFFER BEGIN RAM ADDRESS
208      000234      RMMSGEND = 234      ;MESSAGE BUFFER END RAM ADDRESS
209
210      ;*
211      ; REGISTER DEFINITIONS IN THE MESSAGE BUFFER
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      ; OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
221      ;-
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

```

TSV05 REGISTER AND PACKET DEFINITIONS

```

239
240      ;*
241      ;BSEL0 SELECT CODES FOR WRITE SUBSYSTEM COMMAND
242      ;-
243      000000      PW.NOP          = 0          ;NO-OP
244      000001      PW.RDRAM       = 1          ;READ RAM
245      000002      PW.WTRAM       = 2          ;WRITE RAM
246      000003      PW.RFIFO       = 3          ;READ FIFO
247      000004      PW.WFIFO       = 4          ;WRITE FIFO
248      000005      PW.RDSTAT      = 5          ;READ STATUS
249      000006      PW.WCTL        = 6          ;WRITE TAPE CONTROL
250      000007      PW.WFMT        = 7          ;WRITE TAPE FORMAT
251      000010      PW.WMISC       = 10         ;WRITE MISCELLANEOUS
252      000011      PW.WNPR       = 11         ;WRITE NPR CONTROL
253      000020      PW.D22        = 20         ;DO MICROTEST 22
254      000021      PW.D11        = 21         ;DO MICROTEST 11
255      000022      PW.D13        = 22         ;DO MICROTEST 13
256      000023      PW.NO1311     = 23         ;DISABLE MICROTEST 11 AND 13
257      000024      PW.RDEXT      = 24         ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS)
258
259      ;*
260      ;BSEL1 CODES FOR WRITE TAPE CONTROL
261      ;-
262      000200      WC.IFAD        = BIT7       ;IFAD - FORMATTER ADDRESS
263      000100      WC.IOTAD       = BIT6       ;ITADO - TRANSPORT ADDRESS BIT 0
264      000040      WC.I1TAD      = BIT5       ;ITAD1 - TRANSPORT ADDRESS BIT 1
265      000020      WC.ISRESV     = BIT4       ;IRESV5 - RESERVED #5
266      000010      WC.IREW       = BIT3       ;IREW - REWIND
267      000004      WC.IRWU       = BIT2       ;IRWU - REWIND AND UNLOAD
268      000002      WC.IFEN       = BIT1       ;IFEN - FORMATTER ENABLE
269      000001      WC.IGO        = BIT0       ;GO
270
271      ;*
272      ;BSEL1 CODES FOR WRITE FORMAT
273      ;-
274      000200      WF.IHISP       = BIT7       ;IHISP - HIGH SPEED
275      000100      WF.IWRT       = BIT6       ;IWRT - WRITE
276      000040      WF.IREV       = BIT5       ;IREV - REVERSE
277      000020      WF.IWFM       = BIT4       ;IWFM - WRITE FILE MARK
278      000010      WF.IEDIT      = BIT3       ;IEDIT - EDIT
279      000004      WF.IERASE     = BIT2       ;IERASE - ERASE
280      000002      WF.I3RESV     = BIT1       ;IRESV3 - RESERVED #3
281      000001      WF.I4RESV     = BIT0       ;IRESV4 - RESERVED #4
282
283      ;*
284      ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
285      ;-
286      000200      MS.EXT        = BIT7       ;INVERT SENSE OF EXTENDED FEATURES SWITCH
287      000020      MS.RSFIFO     = BIT4       ;RESET FIFO AND INPUT PARITY ERRORR
288      000010      MS.RSTAPE     = BIT3       ;RESET TAPE STATUS IN 2 FLIP-FLOPS
289      000006      MS.ATTN       = BIT2:BIT1  ;ATTENTION TRIGGER FIELD
290      000001      MS.RSD        = BIT0       ;RESET TIMER A,B THEN DELAY TIMES IN SEL2

```


TSV05 REGISTER AND PACKET DEFINITIONS

```

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 TRANSISTION
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      ;INTERRUPT REQUEST (0-1 TRANSITION)
302      000100      NP.OUT     = BIT6      ;TAPE DATA DIRECTION OUT (0= IN)
303      000040      NP.LOOP    = BIT5      ;ENABLE TRANSPORT LOOPBACK
304      000020      NP.WRP     = BIT4      ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
305
306      ;* 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   = BIT5      ;OUT RDY H
312      000020      S2.INRDY    = BIT4      ;IN RDY H
313      000010      S2.ATIMR    = BIT3      ;TIMER A FLAG H
314      000004      S2.BTIMR    = BIT2      ;TIMER B FLAG H
315      000003      S2.UNDEF    = BIT1-BIT0 ; (UNDEFINED)
316      100000      S1.PARIN    = BIT15     ;WORD #8 BYTE 1 PARIN H
317      040000      S1.I2RESV   = BIT14     ;IRESV2
318      020000      S1.I1RESV   = BIT13     ;IRESV1
319      010000      S1.IEOT     = BIT12     ;IEOT L
320      004000      S1.IIDENT   = BIT11     ;IIDENT H
321      002000      S1.ICER     = BIT10     ;ICER H
322      001000      S1.IFMK     = BIT9      ;IFMK H
323      000400      S1.IHER     = BIT8      ;IHER H
324      000200      S0.ISPEED   = BIT7      ;WORD #8 BYTE 0 ISPEED H
325      000100      S0.IRDY     = BIT6      ;IRDY L
326      000040      S0.IONL     = BIT5      ;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

```

K3

SPECIAL MACROS AND OPDEFS.

```

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

```


L3

SPECIAL MACROS AND OPDEFS.

```

388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407 002174 000000
408
409

```

```

;+
;MACRO TO PERFORM XOR
;-

        .MACRO XOR A,B
        MOV A,-(SP)
        BIC B,(SP)
        BIC A,B
        BIS (SP)+,B
        .ENDM

        EN=0
        .SBTTL FORCER - FORCE ERROR FLAG
; INITIALIZE ERROR NUMBER

; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
;

FORCER:: 0 ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED -
; - BY THE MACRO "IFERROR"). AN ERROR NEED NOT -
; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.

```

M3

TSV3 - GLOBAL AREAS

MACRO V05.03 Tuesday 26-Apr-87 10:28 Page 36

SEQ 0038

GLOBAL DATA SECTION

.SBTTL GLOBAL DATA SECTION

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

002176 000000
002200 000000
002202 000000
002204 000000
002206 000224
002210 000200
002212 000000
002214 000000
002216 000000
002220 000000
002222 000000
002224 000000
002226 000000
002230 000000
002232 000000
002234 000000
002236 000000
002240 000000
002242
002302 000000
002304 000000
002306 000000
002310 000000
002312 000000
002314 000000
002316 000000
002320 000000
002322
002466
002632

```

;--
;THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
;IN MORE THAN ONE TEST.
;--

;THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
;SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.

EPRTSW::      .WORD 0      ;PRINT SWITCH
UNITN::       .WORD 0      ;UNIT # UNDER TEST.
QVP::         .WORD 0      ;QUICK VERIFY FLAG.
CSRADDR::     .WORD 0      ;ADDRESS OF CSR FOR CURRENT DEVICE
IVEC::        .WORD 224    ;INTERRUPT VECTOR
IPRI::        .WORD PRI04  ;INTERRUPT PRIORITY.
TSTCNT::      .WORD 0      ;NUMBER OF TESTS RUN IN THIS PASS
LOOPCNT::     .WORD 0      ;REMAINING ITERATION COUNT FOR TEST
DEVCNT::      .WORD 0      ;NUMBER OF DEVICE UNDER TEST
FATFLG::      .WORD 0      ;SET IF FATAL ERROR IS DETECTED IN TEST
INTRECV::     .WORD 0      ;SET IF TAPE INTERRUPT WAS RECEIVED
EXTFEA::      .WORD 0      ;EXTENDED FEATURES SOFTWARE SW 0-OFF,1-ON
REV::         .WORD 0      ;REV LEVEL (old microcode=1,new micro=2)
BENBSW::      .WORD 0      ;BUFFER ENABLE SWITCH SW 0-OFF,1-ON
EXPD::        .WORD 0      ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
RCV::         .WORD 0      ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
ERRHI::       .WORD 0      ;HIGH ADDRESS MEMORY ERROR
ERRLO::       .WORD 0      ;LOW ADDRESS MEMORY ERROR
RANDATA::     .BLKW 16.    ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
RAMSIZ::      .WORD 0      ;RAM DATA SIZE FOR PRAMPKT ROUTINE
RCVHIADD::    .WORD 0      ;RECEIVED BUFFER HIGH ADDRESS
RCVLOADD::    .WORD 0      ;RECEIVED BUFFER LOW ADDRESS
COUNT::     .WORD 0      ;TEST COUNT PATTERN
DATA::        .WORD 0      ;TEST DATA
TSTFLAG::     .WORD 0      ;TEST FLAG WORD
TSTPTR::      .WORD 0      ;TSTBLK POINTER
PRMNO::       .WORD 0      ;PRINT ROUTINE TEMP
EXPMSG::      .BLKB 100.   ;EXPECTED MESSAGE BUFFER DATA
RCVMSG::      .BLKB 100.   ;RECEIVED MESSAGE BUFFER DATA
TMPBFR::     .BLKB 80.    ;TEMPORARY STORAGE FOR PRINT

```


N3

TSTBLK - TEST DATA TABLE

```

453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469 002752
470 002752 000000
471 002754 177777
472 002756 000001
473 002760 000002
474 002762 000004
475 002764 000010
476 002766 000020
477 002770 000040
478 002772 000100
479 002774 000200
480 002776 000400
481 003000 001000
482 003002 002000
483 003004 004000
484 003006 010000
485 003010 020000
486 003012 040000
487 003014 100000
488 003016 177776
489 003020 177775
490 003022 177773
491 003024 177767
492 003026 177757
493 003030 177737
494 003032 177677
495 003034 177577
496 003036 177377
497 003040 176777
498 003042 175777
499 003044 173777
500 003046 167777
501 003050 157777
502 003052 137777
503 003054 077777
504 003056 125252
505 003060 052525
506 003062

```

.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

```

TBEND==.

B4

GLOBAL ENVIRONMENT STORAGE

```

508                                     .SBTTL  GLOBAL ENVIRONMENT STORAGE
509                                     ;
510                                     ; STORAGE FOR DEVICE REGISTERS
511                                     ;
512 003062 000000 100000 000000 DUMMY: 0,100000,0,0 ; DUMMY DEVICE REGISTERS...
513 003072 000000 000000 000000      0,0,0,0,0,0,0,0 ; ...FOR MULTI-UNIT CHECKOUT.
514                                     ;
515                                     ;
516 003112 000000 DUFLG::          .WORD 0           ; "DROPPED UNIT" FLAG.
517                                     ; INHIBITS CODE IN "CLEAN-UP".
518 003114 000000 NODEV::          .WORD 0           ; FLAG TO SAY NO DEVICE.
519                                     ;
520 003116 000000 TEMP1::          .WORD 0           ; SOME TEMP LOCATIONS.
521 003120 000000 TEMP2::          .WORD 0
522 003122 000000 XXCOMM::        .WORD 0           ; XXDP+ (COMM BLOCK POINTER.
523 003124 000000 FREE::          .WORD 0           ; 1ST FREE MEMORY ADDRESS...
524 003126 000000 FRESIZ::        .WORD 0           ; ...AND SIZE (IN WORDS).
525 003130 000000 FREEHI::        .WORD 0           ; LAST WORD IN FREE SPACE
526 003132 000000 KTFLG::        .WORD 0           ; KT11, MEM AVAIL FLAG -
527                                     ; - .WORD 0 = <24K OR NO KT -
528                                     ; - NZ = >24K AND KT.
529 003134 000000 KTENABLE::       .WORD 0           ; SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
530 003136 000000 NXMFLG::        .WORD 0           ; SET IF WE CAN TEST CLEARED OTHERWISE
531 003140 000000 NXML0::         .WORD 0           ; NXM LO ADDRESS BITS
532 003142 000000 NXMHI::        .WORD 0           ; NXM HI ADDRESS BITS FOR DAL'S 16-21
533 003144 000000 T23A::         .WORD 0           ; 11/23A FLAG
534 003146 000000 T23B::         .WORD 0           ; 11/23B FLAG
535 003150 000000 T3BFLG::        .WORD 0           ; TEST 3B FLAG +0
536 003152 002000 PST32W::        .WORD 2000        ; 32W BLOCK ADDRESS FOR 32K START
537 003154 000000 SIFLAG::        .WORD 0
538 003156 000000 BADDAT::        .WORD 0           ; ACTUAL DATA
539 003160 000000 GDDAT::         .WORD 0           ; EXPECTED DATA
540 003162 000000 LOOPFL::        .WORD 0
541 003164 000000 CTAB::          .WORD 0           ; CONFIGURATION TABLES.
542 003164 000000 CTABM::         .WORD 0           ; CONFIG WORK.
543 003166 000000
544 003170 000000
545 003172 000000
546 003174 177777
547 003176
548                                     ;
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                                     ;
557 003176 000000 ERTABL::         .BLKW 64.
558 003376 000000 ERTABE::        .WORD 0
559
560 003400 000000 SKIPT:          .WORD 0           ; 1=SKIP SUBTEST 0=NO SKIP OF SUBTEST

```


SEQ 0041

562					.SBTTL GLOBAL TEXT MESSAGES
563					;
564					; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
565					; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
566					; MORE THAN ONE TEST.
567					--
568					;
569					; NAMES OF DEVICES SUPPORTED
570					-
571	003402				DEVTYPE <TSV05>
	003402				L#DVTYP::
	003402	124	123	126	.ASCIZ /TSV05/
					.EVEN
572					;
574					; TEST DESCRIPTION
575					-
576					DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****>
577	003410				L#DESC::
	003410				.ASCIZ /**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****/
	003410	052	052	052	.EVEN
598					;
599					; BIT TO ASCII CONVERSION FOR TSSR REGISTER
600					-
601					TSSRBIT::
602	003502	003542	003545	003551	.WORD 1#,2#,3#,4#,5#,6#,7#,8#
603	003522	003603	003607	003613	.WORD 9#,10#,11#,12#,13#,14#,15#,16#
604	003542	123	103	000	1#: .ASCIZ 'SC'
605	003545	102	111	105	2#: .ASCIZ 'BIE'
606	003551	123	103	105	3#: .ASCIZ 'SCE'
607	003555	122	115	122	4#: .ASCIZ 'RMF'
608	003561	116	130	115	5#: .ASCIZ 'NXM'
609	003565	116	102	101	6#: .ASCIZ 'NBA'
610	003571	102	111	124	7#: .ASCIZ 'BIT9'
611	003576	102	111	124	8#: .ASCIZ 'BIT8'
612	003603	123	123	122	9#: .ASCIZ 'SSR'
613	003607	117	106	114	10#: .ASCIZ 'OFL'
614	003613	102	111	124	11#: .ASCIZ 'BIT5'
615	003620	102	111	124	12#: .ASCIZ 'BIT4'
616	003625	102	111	124	13#: .ASCIZ 'BIT3'
617	003632	102	111	124	14#: .ASCIZ 'BIT2'
618	003637	102	111	124	15#: .ASCIZ 'BIT1'
619	003644	102	111	124	16#: .ASCIZ 'BIT0'
620					.EVEN
621	003652	124	123	123	SFIERR: .ASCIZ 'TSSR ERROR AFTER SOFT INIT'
622	003705	124	123	123	SFHERR: .ASCIZ 'TSSR ERROR AFTER BUS RESET'
623	003740	040	040	116	NXR: .ASCIZ '/ NON-EXISTANT DEVICE REGISTER/'
624	003777	045	101	040	NXRX: .ASCIZ '/#A ADDRESS: #06/'
625	004020	045	101	040	TSSX: .ASCII '/#A TSBA,TSSR EXP'D: #06#A,#06#N/'
626	004060	045	101	040	.ASCIZ '/#A TSBA,TSSR REC'D: #06#A,#06/'
627	004117	045	116	045	FUSI: .ASCII '/#N#A/'
628	004123	040	040	125	USI: .ASCIZ '/ UNEXPECTED INTERRUPT/'
629	004152	040	040	111	NSI: .ASCIZ '/ INTERRUPT EXPECTED, NOT RECEIVED/'
630	004215	045	116	045	FNOINTR: .ASCII '/#N#A/'
631	004221	040	040	116	NOINTR: .ASCIZ '/ NO INTERRUPT WAS GENERATED/'
632	004256	040	040	111	IFALT: .ASCIZ '/ INTERRUPT FAULT/'
633	004300	045	101	040	INTX: .ASCIZ '/#A CPU PC: #06#A TSBA: #06/'

GLOBAL TEXT MESSAGES

```

634 004335      040      040      042 NOINIT: .ASCIZ / "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
635 004407      040      040      042 NSINIT: .ASCIZ / "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
636 004457      040      040      042 BRINIT: .ASCIZ / "BUS-RESET" DIDN'T INITIALIZE THE DPU/
637 004527      000
638 004530      045      116      000 NULCR: .ASCIZ /#N/
639 004533      045      101      040 EXPGOT: .ASCIZ /#A EXP'D: #06#A, REC'D: #06/
640 004567      045      116      045 EXPGT2: .ASCIZ /#N#A EXP'D: #06#A, #06#N#A REC'D: #0#A, #06/
641 004643      045      101      040 DUAD12: .ASCIZ /#A REG(W) WRITTEN TO: #06#A REG(R) READ; EXP'D: #06#A, REC'D: #06/
642 004745      122      101      115 PKTRAM: .ASCIZ 'RAM Contents Do Not Match Packet Sent'
643 005013      040      040      103 SCME: .ASCIZ / CONFIG DOESN'T MATCH MFG. MASTER/
644 005056      127      122      111 WRTMSG: .ASCIZ 'WRITE CHARACTERISTICS Failed'
645 005113      124      123      123 WRTERR: .ASCIZ 'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
646 005206      124      123      123 RDERR: .ASCIZ 'TSSR Incorrect After READ Command, More Bits Set Than SSR'
647 005300      106      101      124 SCHERR: .ASCIZ 'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
648 005372      105      122      122 RETERR: .ASCIZ 'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
649 005460      045      116      045 NOMEM: .ASCIZ '#N#A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****N'
650 005554      045      116      045 M8186: .ASCIZ '#N#A ***** 11/23A SYSTEM *****N'
651 005645      045      116      045 M8189: .ASCIZ '#N#A ***** 11/23B SYSTEM *****N'
652
653 .EVEN
654 .SBTTL GLOBAL ERROR REPORT SECTION
655
656
657
658
659

```

```

; **
; 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.
; --

```

```

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

```

```

L10002: TRAP C#MSG

```

```

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

```

```

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

```

```

660 005736
661 005736 013746 003114
005736 012746 003777
005742 012746 000002
005746 012746
005752 010600
005754 104415
005756 062706 000006
662 005762 004737 005770
663 005766
005766 104423
664
665
666
667
668 005770 005727
669 005772 000000
670 005774 001402
671 005776 004777 177770
672 006002
006002 012746 004530
006006 012746 000001
006012 010600
006014 104415
006016 062706 000004
673 006022 000207

```


E4

PRITSSR - PRINT TSSR CONTENTS

```

675                                     .SBTTL  PRITSSR - PRINT TSSR CONTENTS
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
006024
006024
006030
006032
006032
006034
006040
006044
006046
006050
006054
006056
006062
006064
006064
006070
006074
006076
006100
006104
006106
006112
006114
006120
006124
006126
006130
006132
006134
006136
006140
006142
006144
006152
006154
006156
006160
006160
006164
010104
010446
012746
012746
010600
104414
062706
010400
004737
103410
012746
012746
010600
104415
062706
010403
042703
001434
012702
012701
005703
001413
000241
006103
103006
011100
112022
001376
112762
005721
000763
105042
002632
006606
006415
000002
000006
016124
006635
000001
000004
001476
002632
003502
000054
177777
002632
006606

```

```

ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
BY A MESSAGE PRINTING ROUTINE
INPUTS:
R1      CONTENTS OF TSSR
SUBORDINATE ROUTINES:
CHKAMB  CHECK FOR AMBIGUOUS CONTENTS
PRITSSR:
SAVREG      ;SAVE GENERAL REGISTERS
MOV R1,R4   ;SAVE THE TSSR CONTENTS
PRINTB      ;PRINT THE CONTENTS OF TSSR
MOV R4,--(SP)
MOV #TSSRFOR,--(SP)
MOV #2,--(SP)
MOV SP,R0
TRAP C#PNTB
ADD #6,SP
MOV R4,R0   ;GET TSSR BACK FOR CHKAMB
JSR PC,CHKAMB ;ARE CONTENTS AMBIGUOUS ?
BCS 54      ;BRANCH IF NOT
PRINTX      ;SHOW CONTENTS ARE AMBIGUOUS
MOV #AMBTSSR,--(SP)
MOV #1,--(SP)
MOV SP,R0
TRAP C#PNTX
ADD #4,SP
54: MOV R4,R3   ;CONTENTS OF TSSR
BIC #HIADDR!FATERR!TERCLS,R3 ;CLEAR ALL MULTIPLE BIT FIELDS
BEQ 204      ;NO BITS ARE SET
MOV #TMPBFR,R2 ;TEMPORARY ASCII BUFFER
MOV #TSSRBIT,R1 ;ASCII EQUIVALENT OF BITS
104: TST R3   ;REMAINING BITS TO CONVERT
BEQ 154      ;BRANCH WHEN ALL ARE DONE
CLC          ;CLEAR CARRY FOR SHIFT
ROL R3       ;SHIFT NEXT BIT TO CARRY
BCC 134      ;BRANCH IF BIT NOT SET
MOV (R1),R0  ;POINTER TO BIT DEFINITION
114: MOVB (R0),--(R2) ;MOVE ASCII TO BUFFER
BNE 114      ;MOVE ALL BITS
MOVB #'-1(R2) ;INSERT A COMMA TO TERMINATE
134: TST (R1) ;POINT TO NEXT DESCRIPTION
BR 104       ;GET THE REMAINING BITS
154: CLRB --(R2) ;TERMINATE THE LINE
PRINTX #TSSDEF,#TMPBFR ;PRINT THE BIT DEFINITIONS
MOV #TMPBFR,--(SP)
MOV #TSSDEF,--(SP)

```

F4

TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 40-1

SEQ 0044

PRITSSR - PRINT TSSR CONTENTS

006170	012746	000002	MOV	#2, -(SP)	
006174	010600		MOV	SP, R0	
006176	104415		TRAP	C4PNTX	
006200	062706	000006	ADD	#6, SP	
719					
720	006204	010403	204:	MOV	R4, R3
721	006206	042703		BIC	#1CTERCLS, R3
722	006212	016303		MOV	TCOCOD(R3), R3
723	006216			PRINTX	#TCOASC, R3
	006216	010346		MOV	R3, -(SP)
	006220	012746		MOV	#TCOASC, -(SP)
	006224	012746		MOV	#2, -(SP)
	006230	010600		MOV	SP, R0
	006232	104415		TRAP	C4PNTX
	006234	062706		ADD	#6, SP
724	006240	010403		MOV	R4, R3
725	006242	042703		BIC	#1CFATERR, R3
726	006246	001416		BEQ	254
727	006250	006203		ASR	R3
728	006252	006203		ASR	R3
729	006254	006203		ASR	R3
730	006256	016303		MOV	TSFCOD(R3), R3
731	006262			PRINTX	#TFCASC, R3
	006262	010346		MOV	R3, -(SP)
	006264	012746		MOV	#TFCASC, -(SP)
	006270	012746		MOV	#2, -(SP)
	006274	010600		MOV	SP, R0
	006276	104415		TRAP	C4PNTX
	006300	062706		ADD	#6, SP
732	006304	042704	254:	BIC	#1CHIADDR, R4
733	006310	001411		BEQ	304
734	006312			PRINTX	#TEXASC, R4
	006312	010446		MOV	R4, -(SP)
	006314	012746		MOV	#TEXASC, -(SP)
	006320	012746		MOV	#2, -(SP)
	006324	010600		MOV	SP, R0
	006326	104415		TRAP	C4PNTX
	006330	062706		ADD	#6, SP
735	006334	013703	304:	MOV	EPRTSW, R3
736	006340			PRINTX	R3
	006340	010346		MOV	R3, -(SP)
	006342	012746		MOV	#1, -(SP)
	006346	010600		MOV	SP, R0
	006350	104415		TRAP	C4PNTX
	006352	062706		ADD	#4, SP
737	006356	000207		RTS	PC

;GET THE TSSR CONTENTS
 ;CLEAR ALL BUT TERMINATION
 ;GET THE TERMINATION CODE MEANING
 ;PRINT THE TERMINATION CODE

 ;TSSR CONTENTS AGAIN
 ;CLEAR ALL BUT FATAL TERMINATION
 ;DON'T PRINT IF ZERO

 ;ALINE TERMINATION CODE FOR INDEX
 ;GET THE FATAL TERMINATION CODE
 ;PRINT THE FATAL TERMINATION CODE

 ;CLEAR ALL BUT EXTENDED ADDRESS
 ;DON'T PRINT IF ZERO
 ;PRINT THE EXTENDED ADDRESS BITS

 ;PRINT MEASGE BUFFER ADDRESS
 ;PRINT PROPER MESSAGE

 ;RETURN TO CALLER

G4

TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 41

SEQ 0045

PRITSSR - PRINT TSSR CONTENTS

740	006360				EPRT2:		
741	006360	045	116	045	EPRT1:	.ASCIZ	'\$N\$A *****REPLACE M7196*****'
756	006415	045	116	045	TSSRFOR:	.ASCIZ	'\$N\$A TSSR = \$06'
757	006435	045	116	045	TEXASC:	.ASCIZ	'\$N\$A Extended Address Bits = \$06'
758	006476	045	116	045	TCOASC:	.ASCIZ	'\$N\$A Termination Class Code = \$T'
759	006537	045	116	045	TFCASC:	.ASCIZ	'\$N\$A Fatal Termination Class Code = \$T'
760	006606	045	116	045	TSSDEF:	.ASCIZ	'\$N\$A TSSR Bits Set: \$T'
761	006635	045	116	045	AMBTSSR:	.ASCIZ	'\$N\$A TSSR Contents Are Ambiguous'
762						.EVEN	
763	006676	006716	006741	006767	TCOCOD:	.WORD	1\$,2\$,3\$,4\$,5\$,6\$,7\$,8\$
764	006716	116	157	162	1\$:	.ASCIZ	'Normal Termination'
765	006741	124	145	162	2\$:	.ASCIZ	'Termination Condition'
766	006767	124	141	160	3\$:	.ASCIZ	'Tape Status Alert'
767	007011	106	165	156	4\$:	.ASCIZ	'Function Reject'
768	007031	122	145	143	5\$:	.ASCIZ	'Recoverable Error - Tape Position One Record Down'
769	007113	122	145	143	6\$:	.ASCIZ	'Recoverable Error - Tape Was Not Moved'
770	007162	125	156	162	7\$:	.ASCIZ	'Unrecoverable Error'
771	007206	106	141	164	8\$:	.ASCIZ	'Fatal Controller Error'
772						.EVEN	
773							
774	007236	007246	007302	007313	TSFCOD:	.WORD	1\$,2\$,3\$,4\$
775	007246	111	156	164	1\$:	.ASCIZ	'Internal Diagnostic Failure'
776	007302	122	145	163	2\$:	.ASCIZ	'Reserved'
777	007313	102	165	163	3\$:	.ASCIZ	'Bus Interface or Sanity Check Error'
778	007357	122	145	163	4\$:	.ASCIZ	'Reserved'
779						.EVEN	

H4

PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

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

```

781                                     .SBTTL PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET
782
783                                     ;*
784                                     ;THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
785                                     ;THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
786
787                                     ;INPUT:
788
789                                     ;      R0      NUMBER OF WORDS IN PACKET
790                                     ;      R3      HIGH ORDER COMMAND PACKET ADDRESS
791                                     ;      R4      ADDRESS OF COMMAND PACKET
792
793                                     ;      NOTE:   R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
794                                     ;
795
796 007370 PRIPKT:: SAVREG                                ;SAVE THE REGISTERS
797 007370      MOV      R0,R5                                ;SAVE NO. OF WORDS IN PACKET
798 007374 010005      TST      KTENABLE                    ;ABOVE 28K UNDER TEST?
799 007376 005737 003134      BNE      104                 ;BR IF YES
800 007402 001001      CLR      R3                        ;SET HIGH ORDER ADDRESS TO 0
801 007404 005003      MOV      R3,R1                    ;COPY HIGH ORDER ADDRESS
802 007406 010301 104:      MOV      R4,R0                ;GET LOWER ADDRESS
803 007410 010400      ROL      R0                        ;SHIFT BIT 15 INTO C BIT
804 007412 006100      ROL      R1                        ;AND INTO HIGH ORDER.
805 007414 006101      PRINTB  #PKTADD,R1,R4              ;PRINT PACKET ADDRESS
806 007416      MOV      R4,-(SP)
807 007420 010446      MOV      R1,-(SP)
808 007422 012746 007554      MOV      #PKTADD,-(SP)
809 007426 012746 000003      MOV      #3,-(SP)
810 007432 010600      MOV      SP,R0
811 007434 104414      TRAP     C#PNTB
812 007436 062706 000010      ADD      #10,SP
813 007442 010300 154:      MOV      R3,R0                ;GET HIGH ORDER ADDRESS
814 007444 001404      BEQ      204                        ;BR IF NOT ABOVE 28K.
815 007446 010401      MOV      R4,R1                    ;GET LOW ORDER ADDRESS
816 007450 004737 017376      JSR      PC,SETMAP           ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
817 007454 010004      MOV      R0,R4                    ;GET RETURNED PAR6 ADDRESS BIAS
818 007456 005001 204:      CLR      R1                  ;SAVE WORD NUMBER
819 007460 012402 254:      MOV      (R4),R2              ;GET PACKET CONTENTS
820 007462      PRINTB  #PKTFRM,R1,R2                    ;PRINT THE DATA
821 007464 010246      MOV      R2,-(SP)
822 007466 010146      MOV      R1,-(SP)
823 007472 012746 007516      MOV      #PKTFRM,-(SP)
824 007476 012746 000003      MOV      #3,-(SP)
825 007480 010600      MOV      SP,R0
826 007500 104414      TRAP     C#PNTB
827 007502 062706 000010      ADD      #10,SP
828 007506 005201      INC      R1                        ;NEXT WORD NUMBER
829 007510 020105      CMP      R1,R5                    ;DONE ALL PACKET WORDS?
830 007512 002762      BLT      254                        ;LOOP TILL ALL DONE
831 007514 000207      RTS      PC                        ;RETURN
832
833 007516      045      116      045 PKTFRM: .ASCIZ  'N#A Packet Word #D1#A = #06'
834 007554      045      116      045 PKTADD: .ASCIZ  'N#A Packet Address = #01#05'
835                                     .EVEN

```


I4

PRIBXOR - PRINT EXPD, RECV AND XOR BYTE

```

824                                     .SBTTL PRIBXOR - PRINT EXPD, RECV AND XOR BYTE
825
826
827 ;*
828 ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
829 ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
830
831 ;INPUTS:
832 ;
833 ;      R1      RECEIVED DATA
834 ;      R2      EXPECTED DATA
835
836 ;OUTPUT:
837 ;
838 ;      R0      XOR OF EXPECTED/RECEIVED DATA
839
840 PRIBXOR::
841     SAVREG                                ;SAVE THE REGISTERS
842     MOV     R2,R3                        ;EXPECTED DATA
843     XOR     R1,R3                        ;FORM THE EXCLUSIVE OR
844     MOV     R0,R1                        ;BYTE MASK
845     BIC     R0,R1                        ;SAVE LOW BYTE RECV
846     BIC     R0,R2                        ;SAVE LOW BYTE EXPD
847     BIC     R0,R3                        ;SAVE LOW BYTE XOR
848     PRINTB  @XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
849     MOV     R3,-(SP)
850     MOV     R1,-(SP)
851     MOV     R2,-(SP)
852     MOV     @XORBFOR,-(SP)
853     MOV     @4,-(SP)
854     MOV     SP,R0
855     TRAP    C#PNTB
856     ADD     @12,SP
857     MOV     R3,R0                        ;R0 HAS XOR ON RETURN
858     RTS     PC                          ;RETURN TO CALLER
859
860 007612 010203
861 007612 010146
862 007616 010246
863 007620 012700 177400
864 007630 040001
865 007634 040002
866 007636 040003
867 007642 010346
868 007642 010146
869 007644 010246
870 007650 012746 007674
871 007654 012746 000004
872 007660 010600
873 007662 104414
874 007664 062706 000012
875 007670 010300
876 007672 000207
877
878 007674 045 116 045 XORBFOR: .ASCIZ 'N#A EXPD: #03#A RECV: #03#A XOR: #03'
879
880 .EVEN
881 .SBTTL PRIBXOR - PRINT EXPD, RECV AND XOR
882
883 ;*
884 ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
885 ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
886
887 ;INPUTS:
888 ;
889 ;      R1      RECEIVED DATA
890 ;      R2      EXPECTED DATA
891
892 ;OUTPUT:
893 ;
894 ;      R0      XOR OF EXPECTED/RECEIVED DATA
895
896 PRIBXOR::
897     SAVREG                                ;SAVE THE REGISTERS
898     MOV     R2,R3                        ;EXPECTED DATA
899     XOR     R1,R3                        ;FORM THE EXCLUSIVE OR
900     PRINTB  @XORFOR,R2,R1,R3 ;PRINT THE MESSAGE

```

J4

PRIXOR - PRINT EXPD, RECV AND XOR

007760	010346				MOV	R3,-(SP)	
007762	010146				MOV	R1,-(SP)	
007764	010246				MOV	R2,-(SP)	
007766	012746	010012			MOV	#XORFOR,-(SP)	
007772	012746	000004			MOV	#4,-(SP)	
007776	010600				MOV	SP,R0	
010000	104414				TRAP	C4PNTB	
010002	062706	000012			ADD	#12,SP	
873 010006	010300				MOV	R3,R0	;R0 HAS XOR ON RETURN
874 010010	000207				RTS	PC	;RETURN TO CALLER
875							
876 010012	045	116	045	XORFOR: .ASCIZ	'#N#A EXPD: #06#A RECV: #06#A XOR: #06'		
877				.EVEN			

K4

PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT

```

879                                     .SBTTL PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT
880
881                                     ;*
882                                     ;
883                                     ;ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
884                                     ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
885                                     ;
886                                     ;INPUTS:
887                                     ;
888                                     ;      R0      OCTAL VALUE TO CONVERT
889                                     ;      R1      TABLE OF POINTERS TO ASCII EQUIVALENT
890                                     ;
891                                     ;-
892
893 010060                               PRIEQU:
894 010060                               SAVREG                                ;SAVE THE REGISTERS
895 010064 000207                       RTS      PC                        ;RETURN TO CALLER
896
897                                     .SBTTL PRIRAM - PRINT RAM ADDRESS
898
899                                     ;*
900                                     ;
901                                     ;PRINT CONTROLLER RAM ADDRESS.
902                                     ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
903                                     ;
904                                     ;INPUTS:
905                                     ;
906                                     ;      R4      RAM ADDRESS
907                                     ;
908                                     ;-
909
910 PRIRAM:                               SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
911                                     PRINTB   #RAMFOR,R4                ;PRINT RAM ADDRESS IN ERROR
912                                     MOV      R4,-(SP)
913                                     MOV      #RAMFOR,-(SP)
914                                     MOV      #2,-(SP)
915                                     MOV      SP,R0
916                                     TRAP     C4PNTB
917                                     ADD      #6,SP
918                                     RTS      PC                        ;RETURN
919
920 010116      045      116      045 RAMFOR: .ASCIZ 'N/A CONTROLLER RAM ADDRESS = #06'
921                                     .EVEN

```

L4

PRIADD - PRINT MEMORY ERROR ADDRESS

```

916          .SBTTL PRIADD - PRINT MEMORY ERROR ADDRESS
917          ;*
918          ;PRINT MEMORY ADDRESS
919          ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
920          ;
921          ; IMPLICIT INPUTS
922          ;
923          ;
924          ;     ERRHI  - HIGH ORDER ADDRESS
925          ;     ERRLO  - LOW ORDER ADDRESS
926          ;
927          ;-
928          PRIADD:
929          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
930          MOV             ERRHI,R0          ;GET HIGH ADDRESS
931          MOV             ERRLO,R1          ;GET LOW ADDRESS
932          MOV             R1,R2            ;COPY LOW ADDRESS
933          ROL             R1               ;SHIFT BIT 15 TO C BIT
934          ROL             R0               ;SHIFT INTO HIGH ORDER
935          PRINTB          #PRIA0,R0,R2      ;PRINT MEMORY ADDRESS IN ERROR
          MOV             R2,-(SP)
          MOV             R0,-(SP)
          MOV             #PRIA0,-(SP)
          MOV             #3,-(SP)
          MOV             SP,R0
          TRAP             C#PNTB
          ADD             #10,SP
          RTS              PC               ;RETURN

```

```

936          010160
937          010160
938          010164 013700 002236
939          010170 013701 002240
940          010174 010102
941          010176 006101
942          010200 006100
943          010202
944          010202 010246
945          010204 010046
946          010206 012746 010230
947          010212 012746 000003
948          010216 010600
949          010220 104414
950          010222 062706 000010
951          010226 000207
952          010230 045 116 045 PRIA0: .ASCIZ 'MEMORY ERROR ADDRESS = #01#05'
953          .EVEN

```

.SBTTL PRITADD - PRINT MEMORY TEST ADDRESS

```

941          ;*
942          ;PRINT MEMORY ADDRESS
943          ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
944          ;
945          ; IMPLICIT INPUTS
946          ;
947          ;
948          ;     ERRHI  - HIGH ORDER ADDRESS
949          ;     ERRLO  - LOW ORDER ADDRESS
950          ;
951          ;-
952          PRITADD:
953          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
954          MOV             ERRHI,R2          ;GET HIGH ADDRESS
955          MOV             ERRLO,R1          ;GET LOW ADDRESS
956          MOV             R1,R2            ;COPY LOW ADDRESS
957          ROL             R1               ;SHIFT BIT 15 TO C BIT
958          ROL             R0               ;SHIFT INTO HIGH ORDER
959          PRINTB          #PRIT0,R1          ;PRINT MEMORY ADDRESS LOW IN ERROR
          MOV             R1,-(SP)
          MOV             #PRIT0,-(SP)
          MOV             #2,-(SP)
          MOV             SP,R0
          TRAP             C#PNTB

```

```

953          010274
954          010274
955          010300 013702 002236
956          010304 013701 002240
957          010310
958          010310 010146
959          010312 012746 010356
960          010316 012746 000002
          010322 010600
          010324 104414

```


M4

TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 45-1

SEQ 0051

PRITADD - PRINT MEMORY TEST ADDRESS

961	010326	062706	000006		ADD	#6,SP	
	010332				PRINTB	#PRIT1,R2	;PRINT MEMORY ADDRESS HIGH IN ERROR
	010332	010246			MOV	R2,-(SP)	
	010334	012746	010421		MOV	#PRIT1,-(SP)	
	010340	012746	000002		MOV	#2,-(SP)	
	010344	010600			MOV	SP,R0	
	010346	104414			TRAP	C#PNTB	
962	010350	062706	000006		ADD	#6,SP	
963	010354	000207			RTS	PC	;RETURN
964	010356	045	116	045	PRIT0:	.ASCIZ	'#N#A MEMORY TEST ADDRESS LOW = #06'
965	010421	045	116	045	PRIT1:	.ASCIZ	'#N#A MEMORY TEST ADDRESS HIGH = #06'
966						.EVEN	

N4

TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 46

SEQ 0052

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

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

```

;*
;ROUTINE TO ISSUE A SPACE RECORDS
;COMMAND (FORWARD OR REVERSE)
;INPUT:
;      R3      NUMBER OF RECORDS TO BE SPACED OVER
;              BIT15 CONTROLS DIRECTION
;              BIT15 = 0 IS FORWARD
;              BIT15 = 1 IS REVERSE
;      R5      FIRST DEVICE UNIBUS ADDRESS
;
;      REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
;OUTPUT:
;      CARRY    SET - SPACE RECORDS COMMAND OK
;              CLR - SPACE RECORDS FAILED
;
;      R0      THE CONTENTS OF R4 IS MOVED TO R0
;
;IMPLICIT OUTPUT:
;      TAPE HAS BEEN MOVED
;SIDE EFFECTS:
;
;-

```

```

1000
1001
1002
1003 010466
1004 010466
1005 010472 012737 000764 010660
1006 010500 012737 140010 010650
1007 010506 005703
1008 010510 100403
1009 010512 010337 010652
1010 010516 000407
1011 010520 042703 100000
1012 010524 010337 010652
1013 010530 052737 000400 010650
1014 010536 012704 010650
1015 010542 010465 000000
1016 010546 004737 016330
1017 010552 103420
1018 010554
      010554 012727 000250
      010560 000000
      010562 013727 002116
      010566 000000
      010570 005367 177772
      010574 001375

```

```

SPACE::
  SAVREG
  MOV     #500.,SDELAY
  MOV     #140010,804
  TST     R3
  BMI     54
  MOV     R3,904
  BR      104
54:      BIC     #BIT15,R3
  MOV     R3,904
  BIS     #BIT8,804
  MOV     #804,R4
  MOV     R4,TSDB(R5)
104:     JSR     PC,WAITF
154:     BCS     204
  DELAY   250
  MOV     #250,(PC).
  .WORD   0
  MOV     L#DLY,(PC).
  .WORD   0
  DEC     -6(PC)
  BNE     -4
;SAVE THE GENERAL REGISTERS
;SET UP DELAY
;SET UP COMMAND, SPACE FORWARD
;CHECK FOR DIRECTION
;BR IF REVERSE INDICATED
;LOAD UP NUMBER OF RECORDS TO SPACE
;GO DO COMMAND
;CLEAR DIRECTION BIT
;LOAD UP NUMBER OF RECORDS TO SPACE
;SET REVERSE BIT IN COMMAND PACKET
;SET UP R4 WITH PACKET ADDRESS
;SEND OUT COMMAND
;WAIT FOR SSR
;BR IF SSR IS SET AND OK
;DELAY ABOUT .25 SECONDS

```


B5

TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 46-1

SEQ 0053

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

010576	005367	177756	DEC	-22(PC)	
010602	001367		BNE	.-20	
1019 010604	005337	010660	DEC	SDELAY	;BUMP DELAY COUNTER DOWN
1020 010610	001356		BNE	15	;BR, IF MORE DELAY
1021 010612	000411		BR	60	;BR IF TROUBLE CARRY = CLEAR
1022 010614	016501	000002	20: MOV	TSSR(R5),R1	;READ TSSR
1023 010620	012702	000200	MOV	#SSR,R2	;SET UP EXPECTED
1024 010624	020201		25: CMP	R2,R1	;ARE THEY OK
1025 010626	001401		BEQ	40	;BR, IF EQUAL = OK
1026 010630	000402		BR	60	;TROUBLE EXIT
1027 010632	000261		40: SEC		;SET CARRY NO TROUBLE
1028 010634	000401		BR	70	;EXIT
1029 010636	000241		60: CLC		;CARRY CLEAR = ERROR
1030 010640			70: MOV	R4,R0	;PASS PACKET ADDRESS
1031 010640	010400		RTS	PC	;RETURN
1032 010642	000207				
1033			:		
1034			:		
1035			:		
1036			;PACKET FOR SPACE COMMAND		
1037			:		
1039	010650		.	=<.*10>&177770	
1041			:		
1042			;COMMAND WORD		
1043 010650	000000		80: .WORD		
1044			;NUMBER OF RECORDS TO BE SPACED OVER WORD		
1045 010652	000000		90: .WORD		
1046 010654	000000		.WORD		
1047 010656	000000		.WORD		
1048 010660	000000		SDELAY: .WORD	0	;DELAY COUNTER
1049			.EVEN		
1050			.SBTTL	WRTCHR - WRITE CHARACTERISTICS COMMAND	

C5

SEQ 0054

WRTCHR - WRITE CHARACTERISTICS COMMAND

```

1052
1053      ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
1054      ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
1055
1056      ;INPUT:
1057      ;      R4      ADDRESS OF PACKET FROM TEST
1058      ;      R5      FIRST DEVICE UNIBUS ADDRESS
1059      ;      REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
1060
1061      ;OUTPUT:
1062      ;      R0      TSSR CONTENTS
1063      ;      CARRY   SET - WRITE CHARACTERISTICS COMMAND OK
1064      ;              CLR - WRITE CHARACTERISTICS FAILED
1065
1066      ;IMPLICIT OUTPUT:
1067      ;
1068      ;      MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
1069      ;      SOFTWARE SWITCHES SET AS FOLLOWS:
1070      ;      EXTFEA = EXTENDED FEATURES PRESENT
1071      ;      BENBSW = BUFFER ENABLE SWITCH ON OR OFF
1072
1073      ;SIDE EFFECTS:
1074
1075      WRTCHR::
1076      SAVREG
1077      CLR      BENBSW      ;SAVE THE GENERAL REGISTERS
1078      CLR      EXTFEA      ;CLEAR BUFFER ENABLE SWITCH
1079      MOV      R4,TSD8(R5) ;CLEAR EXTENDED FEATURES SW SWITCH
1080      JSR      PC,CHKTSSR  ;SEND OUT COMMAND
1081      BCS      204         ;WAIT FOR SSR
1082      BR       604         ;BR, IF SSR IS SET AND OK
1083      MOV      TSSR(R5),R1 ;BR IF TROUBLE CARRY = CLEAR
1084      MOV      #SSR,R2     ;READ TSSR
1085      BIT      #OFL,R1     ;SET UP EXPECTED
1086      BEQ      254         ;WAS OFF LINE SET IN TSSR
1087      BIS      #OFL,R2     ;BR, IF NO OFL SET
1088      CMP      R2,R1       ;MAKE THEM LOOK ALIKE
1089      BEQ      404         ;ARE THEY OK
1090      BR       604         ;BR, IF EQUAL = OK
1091      ADD      #8,R4        ;TROUBLE EXIT
1092      MOV      (R4),R3      ;POINT TO WRT CHARA DATA PACKET
1093      BIT      #X2.EXTF,XST2(R3) ;GET ADDRESS OF MESSAGE BUFFER
1094      BEQ      454         ;EXTENDED FEATURES BIT SET?
1095      INC      EXTFEA       ;BR IF NO
1096      ;SET EXTENDED FEATURES SW SWITCH
1097      BIT      #X2.BUFE,XST2(R3) ;BUFFER ENABLE SWITCH SET
1098      BEQ      504         ;BR, IF SWITCH NOT SET
1099      INC      BENBSW      ;SET SOFTWARE SWITCH FOR ENABLED
1100
1101      MOV      XST2(R3),REV ;MICROCODE REV LEVEL
1102      BIC      #17700,REV   ;CLEAR UNWANTED BITS
1103      CMP      #1,REV       ;IS IT A NEW MICROCODE
1104      BEQ      554         ;NO BR
1105      MOV      #1,EXTFEA    ;ALWAY EXTENDED FEATURE FOR NEW
1106      ;MICROCODE
1107      BIS      #X2.EXTF,XST2(R3) ;EXTENDED FEATURE ALWAYS SET IN
1108      ;MICROCODE

```


D5

TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 47-1

SEQ 0055

WRTCHR - WRITE CHARACTERISTICS COMMAND

1109 011040 000261
1110 011042 000401
1111 011044 000241
1112 011046 016500 000002
1113 011052 000207

554: SEC
 BR 704
604: CLC
704: MOV TSSR(R5),R0
 RTS PC

;SET CARRY NO TROUBLE
;EXIT
;CARRY CLEAR = ERROR
;RETURN TSSR CONTENTS
;RETURN

E5

SEQ 0056

REWIND - POSITION TAPE (REWIND) COMMAND

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 011054
1143 011054
1144 011060 012704 011150
1145 011064 010465 000000
1146 011070 012703 000550
1147 011074 004737 016330
1148 011100 103417
1149 011102
011102 012727 000372
011106 000000
011110 013727 002116
011114 000000
011116 005367 177772
011122 001375
011124 005367 177756
011130 001367
1150 011132 005303
1151 011134 001357
1152 011136 000241
1153 011140 010400
1154 011142 000207
1155
1157 011150
1159 011150
1160 011150 102010
1161 011152 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
; MOV #RMPACK,R4
; MOV R4,TSDB(R5)
; MOV #360,R3
104: JSR PC,WAITF
BCS 204
DELAY 250
MOV #250.,(PC)+
.WORD 0
MOV L#DLY,(PC)+
.WORD 0
DEC -6(PC)
BNE -4
DEC -22(PC)
BNE -20
DEC R3
BNE 104
CLC
204: MOV R4,R0
RTS PC

;SAVE R1-R5 UNTIL NEXT RETURN
;GET PACKET ADDRESS
;SEND PACKET ADDRESS TO EXECUTE
;ENOUGH TIME FOR 2400' REEL TO REWIND
;WAIT FOR SSR TO SET
;LEAVE WHEN SSR IS SET
;WAIT FOR .25 SECONDS

;BUMP COUNTER DOWN
;KEEP GOING
;CLEAR CARRY TO SET ERROR
;PASS THE PACKET ADDRESS
;RETURN

RMPACK: .=<..10>&177770
.WORD 102010
.WORD 0

;POSTION COMMAND (REWIND)
;NOT USED

```


F5

SEQ 0057

CKRAM - COMPARE RAM TO I/O PACKET

```

1163                                     .SBTTL CKRAM - COMPARE RAM TO I/O PACKET
1164                                     ;*
1165                                     ;ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
1166                                     ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
1167                                     ;
1168                                     ;INPUT:
1169                                     ;
1170                                     ;      R4      ADDRESS OF THE COMMAND PACKET
1171                                     ;      R5      FIRST DEVICE UNIBUS ADDRESS
1172                                     ;
1173                                     ;OUTPUT:
1174                                     ;
1175                                     ;      CARRY   SET - RAM MATCHES PACKET
1176                                     ;            CLR - RAM DOES NOT MATCH PACKET
1177                                     ;
1178                                     ;IMPLICIT OUTPUT:
1179                                     ;
1180                                     ;      THE TABLE RAMDATA IS FILLED WITH THE
1181                                     ;      DATA HELD IN RAM.
1182                                     ;      RAMSIZ IS SET TO 8. FOR PRAMPKT ROUTINE
1183                                     ;
1184                                     ;SIDE EFFECTS:
1185                                     ;
1186                                     ;      THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1187                                     ;
1188                                     ;
1189                                     ;
1190                                     ;
1191 CKRAM:: SAVREG                                ;SAVE THE GENERAL REGISTERS
1192                                     MOV          #RAMDATA,R1      ;ADDRESS TO SAVE THE RAM DATA
1193                                     MOV          #RMPKTBEG,R2     ;BYTE ADDRESS OF FIRST RAM DATA
1194                                     CLR          R3              ;CLEAR THE ERROR FLAG
1195                                     JSR          PC,CHKTSSR        ;WAIT FOR SSR
1196                                     MOVW        #0,TSDB(R5)      ;SET MAINTENANCE MODE
1197                                     JSR          PC,CHKTSSR        ;WAIT FOR SSR TO SET
1198                                     MOVW        R2,TSDB(R5)      ;SELECT NEXT RAM ADDRESS
1199                                     JSR          PC,CHKTSSR        ;WAIT FOR SSR TO SET
1200                                     MOVW        TSBA(R5),(R1)    ;READ THE RAM DATA
1201                                     CMPB        (R1),.(R4)       ;COMPARE TO EXPECTED
1202                                     BEQ          20$             ;BRANCH IF OK
1203                                     INC          R3              ;SET ERROR FLAG
1204                                     INC          R2              ;ADDRESS OF NEXT RAM LOCATION
1205                                     CMP         R2,#RMPKTEND     ;REACHED END YET ?
1206                                     BLE         10$             ;BRANCH TILL ALL READ
1207                                     TST         R3              ;WAS AN ERROR FOUND ?
1208                                     BEQ          30$             ;BRANCH IF NOT
1209                                     CLC                     ;CLEAR CARRY TO SHOW ERROR
1210                                     BR           50$             ;AND EXIT
1211                                     SEC                     ;SHOW GOOD COMPARE
1212                                     MOV          #8.,RAMSIZ       ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
1213                                     RTS          PC              ;RETURN
1214
1191 011154
1192 011154
1193 011160 012701 002242
1194 011164 012702 000201
1195 011170 005003
1196 011172 004737 016416
1197 011176 112765 000000 000000
1198 011204 004737 016416 10$:
1199 011210 010265 000000
1200 011214 004737 016416
1201 011220 116511 000000
1202 011224 122124
1203 011226 001401
1204 011230 005203
1205 011232 005202 20$:
1206 011234 020227 000210
1207 011240 003761
1208 011242 005703
1209 011244 001402
1210 011246 000241
1211 011250 000401
1212 011252 000261
1213 011254 012737 000010 002302 30$:
1214 011262 000207 50$:

```

CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA

```

1216 .SBTTL CKRAM2 - COMPARE RAM TO I/O CHARACTERISTICS DATA
1217
1218 ;*
1219 ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
1220 ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
1221
1222 ;INPUT:
1223
1224 ; R4 ADDRESS OF THE CHARACTERISTICS DATA
1225 ; R5 FIRST DEVICE UNIBUS ADDRESS
1226
1227 ;OUTPUT:
1228
1229 ; CARRY SET - RAM MATCHES PACKET
1230 ; CLR - RAM DOES NOT MATCH PACKET
1231
1232 ;IMPLICIT OUTPUT:
1233
1234 ; THE TABLE RAMDATA IS FILLED WITH THE
1235 ; DATA HELD IN RAM.
1236 ; RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
1237
1238 ;SIDE EFFECTS:
1239
1240 ; THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1241
1242 CKRAM2::
1243 SAVREG
1244 MOV #RAMDATA,R1 ;SAVE THE GENERAL REGISTERS
1245 MOV #RMCHBEG,R2 ;ADDRESS TO SAVE THE RAM DATA
1246 CLR R3 ;BYTE ADDRESS OF FIRST RAM DATA
1247 JSR PC,CHKTSSR ;CLEAR THE ERROR FLAG
1248 MOV #0,TSDB(R5) ;WAIT FOR SSR
1249 JSR PC,CHKTSSR ;SET MAINTENANCE MODE
1250 MOV R2,TSDB(R5) ;WAIT FOR SSR TO SET
1251 JSR PC,CHKTSSR ;SELECT NEXT RAM ADDRESS
1252 MOV TSBA(R5),(R1) ;WAIT FOR SSR TO SET
1253 CMPB (R1),.(R4). ;READ THE RAM DATA
1254 BEQ 20$ ;COMPARE TO EXPECTED
1255 INC R3 ;BRANCH IF OK
1256 INC R2 ;SET ERROR FLAG
1257 MOV #8,,RAMSIZ ;ADDRESS OF NEXT RAM LOCATION
1258 TST EXTFEA ;ASSUME EXTFEA NOT SET
1259 BEQ 25$ ;IS THE SOFTWARE EXTENDED FEATURES SET
1260 MOV #10,,RAMSIZ ;BR, IF NOT SET
1261 CMP R2,#RMCHEND ;SET RAMSIZ FOR EXTEND FEATURES
1262 BLE 10$ ;AT END OF EXTENDED BUFFER
1263 BR 27$ ;BR, IF NOT AT END YET
1264 CMP R2,#RMCHEND-2 ;AT END BRANCH
1265 BLE 10$ ;REACHED END YET ?
1266 TST R3 ;BRANCH TILL ALL READ
1267 BEQ 30$ ;WAS AN ERROR FOUND ?
1268 CLC ;BRANCH IF NOT
1269 BR 50$ ;CLEAR CARRY TO SHOW ERROR
1270 SEC ;AND EXIT
1271 RTS PC ;SHOW GOOD COMPARE
;RETURN

```


TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 51

SEQ 0059

CKMSG - COMPARE WRITE CHAR. MESSAGE BUFFERS

```

1273                                     .SBTTL  CKMSG  - COMPARE WRITE CHAR. MESSAGE BUFFERS
1274
1275
1276                                     ;*
1277                                     ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
1278                                     ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1279                                     ;ERROR PRINT ROUTINES.
1280
1281                                     ;INPUT:
1282                                     ;
1283                                     ;      R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1284                                     ;      R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
1285                                     ;      R2      EXPD MESSAGE BUFFER ADDRESS
1286
1287                                     ;OUTPUT:
1288                                     ;
1289                                     ;      CARRY    SET - MESSAGE BUFFERS MATCH
1290                                     ;              CLR -MESSAGE BUFFERS DON'T MATCH
1291
1292                                     ;IMPLICIT OUTPUT:
1293                                     ;
1294                                     ;      EXPMSG    BUFFER IS SET TO EXPD DATA
1295                                     ;      RECVMSG    BUFFER IS SET TO RECV DATA
1296                                     ;      RCVHIADD   SET TO HIGH ORDER ADDRESS OF RECV
1297                                     ;      RCVLOAD    SET TO LOW ORDER ADDRESS OF RECV
1298
1299                                     ;-
1300                                     CKMSG::
1301                                     SAVREG
1302                                     MOV      R0,RCVHIADD      ;SAVE R1-R5 UNTIL NEXT RETURN
1303                                     MOV      R1,RCVLOAD      ;SAVE RECV HIGH ADDRESS
1304                                     TST      KTENABLE        ;SAVE RECV LOW ADDRESS
1305                                     BEQ      10$,            ;TESTING ABOVE 28K?
1306                                     JSR      PC,SETMAP        ;BR IF NO
1307                                     MOV      R0,R1          ;RETURN ADDRESS BIASED TO PAR6 IN R0
1308                                     CLR      R4              ;GET RETURNED ADDRESS BIASED TO PAR6
1309                                     MOV      R2,R5          ;WORD IN BUFFER
1310                                     MOV      (R2),EXPMSG(R4) ;CLEAR ERROR SEEN FLAG
1311                                     MOV      (R1),RECVMSG(R4);GET EXPD BUFFER ADDRESS
1312                                     CMP      (R2),.(R1),    ;SAVE EXPD FOR ERROR REPORT
1313                                     BEQ      25$,            ;SAVE RECV FOR ERROR REPORT
1314                                     INC      R3              ;EXPD EQUAL RECV?
1315                                     ADD      #2,R4          ;BR IF YES
1316                                     CMP      R4,#14         ;SET ERROR SEEN FLAG
1317                                     BLE      15$,            ;POINT TO NEXT WORD ADDRESS
1318                                     CMP      #1,REV         ;DONE FIRST 7 WORDS?
1319                                     BNE      30$,            ;BR IF NO
1320                                     BIT      #X2.EXTF,XST2(R5);IS IT A NEW MICROCODE
1321                                     BEQ      50$,            ;NO BR
1322                                     CMP      R4,#16         ;IS EXTENDED FEATURES SET IN EXPD?
1323                                     BLE      15$,            ;BR IF NO
1324                                     TST      R3              ;DONE EXTENDED FEATURES WORD?
1325                                     BEQ      55$,            ;BR IF NO
1326                                     CLC                      ;ANY ERRORS SEEN?
1327                                     BR       60$,            ;BR IF NO
1328                                     SEC                      ;SET FAILURE
1329                                     RTS      PC              ;SET SUCCESS
1330                                     PC
1331                                     RETURN

```

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

1330 .SBTTL CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS
1331
1332 ;*
1333 ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
1334 ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1335 ;ERROR PRINT ROUTINES.
1336
1337 ;INPUT:
1338 ; R0 RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1339 ; R1 RECV MESSAGE BUFFER LOW ORDER ADDRESS
1340 ; R2 EXPD MESSAGE BUFFER ADDRESS
1341 ; R3 NUMBER OF BYTES TO COMPARE
1342
1343 ;OUTPUT:
1344 ; CARRY SET - MESSAGE BUFFERS MATCH
1345 ; CLR - MESSAGE BUFFERS DON'T MATCH
1346
1347 ;IMPLICIT OUTPUT:
1348 ; EXPMSG BUFFER IS SET TO EXPD DATA
1349 ; RECVMSG BUFFER IS SET TO RECV DATA
1350 ; RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
1351 ; RCVLOAD SET TO LOW ORDER ADDRESS OF RECV
1352
1352 011550 CKMSG2::
1353 011550 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1354 011554 020327 000144 CMP R3,#RECVMSG-EXPMSG, @ad IS COUNT ABOVE MAX ALLOWED?
1355 011560 003412 BLE 5; @ad BR IF NO
1356 011562 012703 000144 MOV #RECVMSG-EXPMSG,R3; @ad
1357 011566 PRINTF #DEBUGMSG ;@ad
1358 011566 012746 011702 MOV #DEBUGMSG,-(SP)
1359 011572 012746 000001 MOV #1,-(SP)
1360 011576 010600 MOV SP,R0
1361 011600 104417 TRAP C;PNTF
1362 011602 062706 000004 ADD #4,SP
1363 011606 010037 002304 5;: MOV R0,RCVHIADD ;SAVE RECV HIGH ADDRESS
1364 011612 010137 002306 MOV R1,RCVLOAD ;SAVE RECV LOW ADDRESS
1365 011616 005737 003134 TST KTENABLE ;TESTING ABOVE 28K?
1366 011622 001403 BEQ 10; ;BR IF NO
1367 011624 004737 017376 JSR PC,SETMAP ;RETURN ADDRESS BIASED TO PAR6 IN R0
1368 011630 010001 MOV R0,R1 ;GET RETURNED ADDRESS BIASED TO PAR6
1369 011632 005004 10;: CLR R4 ;WORD IN BUFFER
1370 011634 005005 CLR R5 ;CLEAR ERROR SEEN FLAG
1371 011636 111264 002322 15;: MOVB (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
1372 011642 111164 002466 MOVB (R1),RECVMSG(R4) ;SAVE RECV FOR ERROR REPORT
1373 011646 122221 CMPB (R2),-(R1) ;EXPD EQUAL RECV?
1374 011650 001401 BEQ 25; ;BR IF YES
1375 011652 005205 INC R5 ;SET ERROR SEEN FLAG
1376 011654 062704 000001 25;: ADD #1,R4 ;POINT TO NEXT BYTE
1377 011660 020403 CMP R4,R3 ;DONE ALL BYTES?
1378 011662 002001 BGE 50; ;BR IF YES
1379 011664 000764 BR 15; ;DO NEXT BYTE
1380 011666 005705 50;: TST R5 ;ANY ERRORS SEEN?
1381 011670 001402 BEQ 55; ;BR IF NO
1382 011672 000241 CLC ;SET FAILURE
1383 011674 000401 BR 60;
1384 011676 000261 55;: SEC ;SET SUCCESS
1385 011700 000207 60;: RTS PC ;RETURN

```


J5

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

1382 011702      120      122      117 DEBUGMSG: .ASCIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-';a0D
1383 011772      045      116      045 FERCM:  .ASCII  /%N%#A  ***/
1384 012003      040      040      124 ERCH:   .ASCIZ  /  TSSR ERROR CODE REC'D =  /
1385 012036      056      056      056 SIMSG:  .ASCIZ  /.... AFTER DOING SOFT INIT/
1386 012071      124      105      123 TINERR: .ASCIZ  /TEST: .../
1387                                     .EVEN
1388                                     ;*
1389                                     ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
1390                                     ;
1391                                     ;INPUT:
1392                                     ;
1393                                     ;      R1      CONTENTS OF TSSR AT ERROR
1394                                     ;
1395                                     ;SIDE EFFECTS:
1396                                     ;
1397                                     ;      EXECUTES DROP UNIT TO CEASE TESTING
1398                                     ;
1399                                     ;
1400                                     ;-
1401
1402 012104      012104      004737  006024      BGNMSG  SFMSG
1403 012104      004737  006024      SFMSG:  :
1404 012110      004737  017262      JSR      PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1405 012114      012114      104423      JSR      PC,CKDROP      ;DROP UNIT, IF ALLOWED
1406 012114      012114      104423      ENDMSG
1407                                     L10003:
1408                                     TRAP      C#MSG
1409
1410                                     ;*
1411                                     ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1412                                     ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
1413                                     ;
1414                                     ;INPUTS:
1415                                     ;
1416                                     ;      R1      TSSR CONTENTS
1417                                     ;      R4      ADDRESS OF COMMAND PACKET
1418                                     ;
1419                                     ;-
1420
1421 012116      012116      004737  006024      BGNMSG  PKTSSR
1422 012116      012116      012700  000004      PKTSSR:  :
1423 012122      004737  007370      JSR      PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1424 012122      004737  007370      MOV      #4,R0      ;NO. OF WORDS IN PACKET
1425 012122      004737  007370      JSR      PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
1426 012122      004737  007370      ENDMSG
1427 012132      012132      104423      L10004:
1428 012132      012132      104423      TRAP      C#MSG

```

K5

SEQ 0062

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```

1424
1425      ;*PRINT ROUTINE TO PRINT THE CONTENTS OF
1426      ;TSSR AND A GET STATUS COMMAND PACKET.
1427
1428      ;INPUTS:
1429
1430      ;      R1      TSSR CONTENTS
1431      ;      R4      ADDRESS OF COMMAND PACKET
1432
1433      ;-
1433 012134      BGNMSG  PKTGETS
1434 012134      PKTGETS::
1435 012134 004737 006024      JSR      PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1436 012140 012700 000002      MOV      #2,R0      ;NO. OF WORDS IN GET STATUS PACKET
1437 012144 004737 007370      JSR      PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
1438 012150      ENDMSG
1439 012150      L10005:
1440 012150      TRAP      C#MSG
1441
1442      ;*PRINT TSSR ERRORS FOR INITIALIZATION TESTS
1443
1444      ;INPUTS:
1445
1446      ;      R1      TSSR CONTENTS
1447      ;      R4      ADDRESS OF COMMAND PACKET
1448
1449      ;-
1449 012152      BGNMSG  SFFMSG
1450 012152      SFFMSG::
1451 012152 004737 006024      JSR      PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1452 012156      ENDMSG
1453 012156      L10006:
1454 012156      TRAP      C#MSG
1455      .SBTTL  PKTMES  - PRINT TSSR AND MESSAGE BUFFER
1456
1457      ;*PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
1458      ;BUFFER FOR ERROR REPORTS
1459
1460      ;INPUTS:
1461
1462      ;      R1      CONTENTS OF TSSR
1463      ;      R2      LOW ORDER MESSAGE BUFFER
1464      ;      R3      HIGH ORDER MESSAGE BUFFER ADDRESS
1465      ;      NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
1466
1467      ;-
1467 012160      BGNMSG  PKTMES
1468 012160      PKTMES::
1469 012160 004737 006024      JSR      PC,PRITSSR      ;PRINT CONTENTS OF TSSR
1470 012164 010200      MOV      R2,R0      ;LOW ORDER ADDRESS
1471 012166 010301      MOV      R3,R1      ;HIGH ORDER ADDRESS
1472 012170 004737 014312      JSR      PC,PRMESS      ;PRINT THE MESSAGE BUFFER
1473 012174      ENDMSG
1474 012174      L10007:
1475 012174      TRAP      C#MSG

```


L5

ADDSSR - PRINT TEST ADDRESS AND TSSR

```

1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479 012176
      012176
1480 012176 004737 010274
1481 012202 016501 000002
1482 012206 004737 006024
1483 012212
      012212
      012212 104423
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497 012214
      012214
1498 012214 012700 000007
1499 012220 005737 002224
1500 012224 001402
1501 012226 012700 000010
1502 012232 004737 014632
1503 012236
      012236
      012236 104423

      .SBTTL ADDSSR - PRINT TEST ADDRESS AND TSSR
      ;*
      ;PRINT ROUTINE TO PRINT THE CONTENTS OF
      ;TSSR AND A MEMORY TEST ADDRESS
      ;INPUTS:
      ;
      ;      R5      FIRST DEVICE UNIBUS ADDRESS
      ;      ERRHI   HIGH ORDER MEMORY TEST ADDRESS
      ;      ERRLO   LOW ORDER MEMORY TEST ADDRESS
      ;
      ;
      BGNMSG ADDSSR
ADDSSR:  JSR      PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
        MOV      TSSR(R5),R1    ;GET CURRENT TSSR
        JSR      PC,PRITSSR     ;PRINT THE CONTENTS OF TSSR REGISTER
        ENDMMSG
L10010:  TRAP     C#MSG

      .SBTTL MSGEXP - PRINT WRITE CHAR. EXPD-RCV MESSAGE BUFFERS
      ;*
      ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
      ;IMPLICIT INPUTS:
      ;
      ;      EXPMSG  - EXPECTED MESSAGE BUFFER
      ;      RECMMSG - RECEIVED MESSAGE BUFFER
      ;      RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
      ;      RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
      ;
      ;
      BGNMSG MSGEXP
MSGEXP:  MOV      #7,R0          ;ASSUME NO EXT FEATURES
        TST      EXTFEA        ;EXT FEATURES SET?
        BEQ      5$,          ;BR IF NO
        MOV      #8,R0          ;EXT FEATURE BUFFER IS 8 WORDS
        JSR      PC,PRMSGEXP    ;PRINT EXPD/RCV MESSAGE BUFFERS
        ENDMMSG
5$:      L10011:  TRAP     C#MSG

```

M5

FIFEXP - PRINT FIFO EXP/RECV DATA

```

1505                                     .SBTTL FIFEXP - PRINT FIFO EXP/RECV DATA
1506                                     ;*
1507                                     ;PRINT ROUTINE TO PRINT FIFO EXP/RECV DATA
1508                                     ;
1509                                     ;      R1      - BYTE COUNT
1510                                     ;
1511                                     ;IMPLICIT INPUTS:
1512                                     ;
1513                                     ;      EXPMSG  - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
1514                                     ;      RECMMSG  - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
1515                                     ;
1516                                     ;-
1517 012240      BGNMSG FIFEXP
1518 012240      FIFEXP::
1519 012240      PRINTX #FIF1MSG,R1      ;PRINT BYTES TRANSFERRED
1520 012240      MOV    R1,-(SP)
1521 012242      MOV    #FIF1MSG,-(SP)
1522 012246      MOV    #2,-(SP)
1523 012252      MOV    SP,R0
1524 012254      TRAP   C#PNTX
1525 012256      ADD    #6,SP
1526 012262      PRINTX #FIF2MSG      ;PRINT HEADER MSG
1527 012266      MOV    #FIF2MSG,-(SP)
1528 012272      MOV    #1,-(SP)
1529 012274      MOV    SP,R0
1530 012276      TRAP   C#PNTX
1531 012302      ADD    #4,SP
1532 012304      MOV    R1,R0      ;GET BYTE COUNT
1533 012310      JSR    PC,PRBYTEXP ;PRINT FIFO BYTES IN ERROR
1534 012312      ENDMSG
1535 012310      L10012:
1536 012310      TRAP   C#MSG
1537 012312      .ASCIZ 'N#A NUMBER OF BYTES TRANSFERRED = #D2'
1538 012361      .ASCIZ 'N#A FIFO DATA BYTES IN ERROR:'
1539                                     .EVEN

```


TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 57

SEQ 0065

MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS

```

1527          .SBTTL MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
1528
1529          ;
1530          ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
1531          ;
1532          ;
1533          ;IMPLICIT INPUTS:
1534          ;
1535          ;      EXPMSG - EXPECTED MESSAGE BUFFER
1536          ;      RECMSG - RECEIVED MESSAGE BUFFER
1537          ;      RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1538          ;      RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1539          ;
1540          BGNMSG MSGSTAT
1541          MSGSTAT:
1542              MOV     #STA^COD,R1      ;ASCII ADDRESS TABLE
1543              MOV     (R1)+,R0         ;DONE ALL MSG LINES?
1544              BEQ     20$,             ;BR IF YES
1545              PRINTX  R0               ;PRINT STATUS BIT NAMES
1546              MOV     R0,-(SP)
1547              MOV     #1,-(SP)
1548              MOV     SP,R0
1549              TRAP    C$PNTX
1550              ADD     #4,SP
1551              BR      10$              ;DO ANOTHER MSG LINE
1552              MOV     #10,R0           ;NUMBER OF WORDS IN A READ STATUS BUFFER
1553              JSR     PC,PRMSGEXP      ;PRINT EXPD/RCV MESSAGE BUFFERS
1554              ENDMMSG
1555          L10013:
1556              TRAP    C$MSG
1557
1558          STATCOD:
1559              .WORD    1$,2$,3$,4$,5$,6$,0
1560              1$:ASCIZ 'sNMA Tape Bus Signals in Word #8:'
1561              2$:ASCIZ 'PARERR<15> IEOT <12> IFMK <9> IRDY<6> IRMD<2>'
1562              3$:ASCIZ 'sNMA IRESV2<14> IIDENT<11> IHER <8> IONL<5> IFBY<1>'
1563              4$:ASCIZ 'sNMA IRESV1<13> ICER <10> ISPEED<7> ILDP<4> IFPT<0>'
1564              5$:ASCIZ 'sNMA Tape Bus Signals in Word #9:'
1565              6$:ASCIZ 'sNMA DATMIS<7> ILW<6> OUTRDY<5> INRDY<4>'
1566          .EVEN

```

B6

MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS

```

1560                                     .SBTTL MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
1561                                     ;*
1562                                     ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
1563                                     ;
1564                                     ;IMPLICIT INPUTS:
1565                                     ;
1566                                     ;
1567                                     ;   EXPMSG - EXPECTED MESSAGE BUFFER
1568                                     ;   RECMSG - RECEIVED MESSAGE BUFFER
1569                                     ;   RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1570                                     ;   RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1571                                     ;
1572                                     ;-
1573                                     BGNMSG MSGLOOP
1574 MSGLOOP:
1575     MOV     #LOOPCOD,R1           ;ASCII ADDRESS TABLE
1576     MOV     (R1)+,R0             ;DONE ALL MSG LINES?
1577     BEQ     20$,R0              ;BR IF YES
1578     PRINTX  R0                  ;PRINT STATUS BIT NAMES
1579     MOV     R0,-(SP)
1580     MOV     #1,-(SP)
1581     MOV     SP,R0
1582     TRAP    C#PNTX
1583     ADD     #4,SP
1584     BR      10$,R0              ;DO ANOTHER MSG LINE
1585     MOV     #10,R0              ;NUMBER OF WORDS IN A READ STATUS BUFFER
1586     JSR     PC,PRMSGEXP          ;PRINT EXPD/RECV MESSAGE BUFFERS
1587     ENDMSG
1588 L10014:
1589     TRAP    C#MSG
1590
1591 LOOPCOD: .WORD 1$,2$,3$,4$,5$,6$,7$,0
1592 1$: .ASCIZ 'N/A Tape Bus Loopback Signals in Word #8:'
1593 2$: .ASCIZ 'N/A PARERR<15> IRESV2<14> IRESV1<13>'
1594 3$: .ASCIZ 'N/A IHISP=>IEOT<12> IWRT=>IIDENT<11> IREV =>ICER <10>'
1595 4$: .ASCIZ 'N/A IWM =>IFMK<09> IEDIT=>IHER <08> IFAD =>ISPEED<07>'
1596 5$: .ASCIZ 'N/A ITADO=>IRDY<06> ITAD1=>IONL <05> IERASE=>ILDPA <04>'
1597 6$: .ASCIZ 'N/A IREW =>IDBY<03> IRWU =>IRWD <02> IFEN =>IFBY <01>'
1598 7$: .ASCIZ 'N/A IGO =>IFPT<00>'
1599 .EVEN

```


TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 59

MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER

```

1592 .SBTTL MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
1593 ;*
1594 ;
1595 ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
1596 ;
1597 ;
1598 ;IMPLICIT INPUTS:
1599 ;
1600 ; EXPMSG - EXPECTED MESSAGE BUFFER
1601 ; RECMSG - RECEIVED MESSAGE BUFFER
1602 ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1603 ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1604 ;
1605 ;-
1606 014012 BGNMSG MSGSUB
1607 014012 MSGSUB::
1608 014012 MOV #10,R0 ;SIZE OF WRITE SUBSYSTEM BUFFER
1609 012700 JSR PC,PRMSGEXP ;PRINT EXPD/RECV MESSAGE BUFFERS
1610 004737 ENDMSG
1611 014022
1612 014022 L10015:
1613 014022 TRAP C#MSG
1614 104423
1615 ;
1616 .SBTTL MEMADD - PRINT MEMORY ADDRESS DATA ERROR
1617 ;*
1618 ;
1619 ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
1620 ;
1621 ;IMPLICIT INPUTS:
1622 ;
1623 ; ERRHI - MEMORY ERROR HIGH ORDER ADDRESS
1624 ; ERRLO - MEMORY ERROR LOW ORDER ADDRESS
1625 ; EXP - EXPECTED DATA
1626 ; RECV - RECEIVED DATA
1627 ;-
1628 014024 BGNMSG MEMADD
1629 014024 MEMADD::
1630 004737 JSR PC,PRIADD ;PRINT MEMORY ADDRESS IN ERROR
1631 010160 MOV EXPD,R1 ;GET EXPD DATA
1632 013701 MOV RECV,R2 ;GET RECEIVED DATA
1633 002232 JSR PC,PRIXOR ;PRINT EXPD/RECV
1634 002234 ENDMSG
1635 004737 L10016:
1636 007742 TRAP C#MSG
1637 014044
1638 014044
1639 014044 104423

```

D6

PRAMPKT - PRINT RAM AND PACKET DATA

```

1629      .SBTTL  PRAMPKT - PRINT RAM AND PACKET DATA
1630
1631      ;*
1632      ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1633      ;WHEN THE RAM DATA DOES NOT MATCH.
1634
1635      ;
1636      ;INPUTS:
1637      ;
1638      ;       R4      POINTER TO COMMAND PACKET
1639      ;IMPLICIT INPUTS:
1640      ;       RAMDATA  DATA AS READ FROM THE RAM
1641      ;       RAMSIZ   NUMBER OF BYTES IN PACKET
1642      ;                   IF RAMSIZ=0 THEN DEFAULT TO 8.
1643
1644      ;IMPLICIT OUTPUTS:
1645      ;       RAMSIZ   SET TO 0
1646
1647      PRAMPKT:
1648      SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
1649      MOV      #RAMDATA,R1                ;DATA FROM THE RAM
1650      CLR      R2                          ;INIT BYTE NUMBER
1651      CMPB     (R1),.(R4).                 ;COMPARE EXPECTED, RECEIVED
1652      BNE      7$,NOTSSR                  ;BR IF NO MATCH
1653
1654      FORCERROR 7$,NOTSSR
1655      BR       10$
1656
1657      7$:    MOVB  -1(R1),R5                ;GET RECV RAM DATA
1658      MOVB     -1(R4),R3                  ;GET EXPD PACKET DATA
1659      XOR      R5,R3                      ;XOR EXPD/RECV
1660      BIC      #177400,R3                 ;LOW BYTE ONLY
1661      MOVB     -1(R1),RECV                ;GET RECEIVED RAM DATA
1662      MOVB     -1(R4),EXPD                ;GET EXPECTED RAM DATA
1663      PRINTB   #RAMASC,R2,RECV,EXPD,R3
1664      MOV      R3,-(SP)
1665      MOV      EXPD,-(SP)
1666      MOV      RECV,-(SP)
1667      MOV      R2,-(SP)
1668      MOV      #RAMASC,-(SP)
1669      MOV      #5,-(SP)
1670      MOV      SP,R0
1671      TRAP     C#PNTB
1672      ADD      #14,SP
1673
1674      10$:   INC      R2                    ;UPDATE BYTE COUNT
1675      TST      RAMSIZ                      ;DEFAULT TO 8.?
1676      BEQ      15$                          ;BR IF YES
1677      CMP      R2,RAMSIZ                   ;DONE ALL BYTES?
1678      BLE      5$                          ;BR IF NO
1679      BR       25$
1680
1681      15$:   CMP      R2,#8.                ;DONE DEFAULT NUMBER OF BYTES?
1682      20$:   BLT      5$                      ;BR IF NO
1683      25$:   CLR      RAMSIZ                ;SET DEFAULT RAMSIZ
1684      RTS      PC                          ;RETURN
1685
1686      1671 014226      045      116      045 RAMASC: .ASCIZ 'N#A BYTE: #D2#A RAM: #03#A Packet: #03#A XOR:#03'
1672      .EVEN

```


E6

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```

1674                                     .SBTTL PRMESS - PRINT CONTENTS OF MESSAGE BUFFER
1675
1676                                     ;*
1677                                     ;THIS ROUTINE PRINTS THE CONTENTS OF
1678                                     ;THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE TSV-05.
1679                                     ;
1680                                     ;INPUT:
1681                                     ;      R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
1682                                     ;      R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
1683                                     ;      NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
1684                                     ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
1685
1685 014312 PRMESS: SAVREG                                     ;SAVE THE REGISTERS
1686 014316 010005 MOV      R0,R5                               ;SAVE LOW ORDER ADDRESS
1687 014320 005737 003134 TST      KTENABLE                     ;ADDRESS ABOVE 28K?
1688 014324 001001 BNE      104                                ;BR IF YES
1689 014326 005001 CLR      R1                                 ;SET HIGH ORDER ADDRESS TO 0
1690 014330 010103 104: MOV      R1,R3                         ;SAVE HIGH ORDER ADDRESS
1691 014332 006100 ROL      R0                                ;SHIFT BIT15 TO C BIT
1692 014334 006101 ROL      R1                                ;SHIFT TO HIGH ORDER FOR PRINTOUT
1693 014336 PRINTX  #PROASC,R1,R5                               ;PRINT MESSAGE BUFFER ADDRESS
1694 014336 010546 MOV      R5,-(SP)
1695 014340 010146 MOV      R1,-(SP)
1696 014342 012746 014500 MOV      #PROASC,-(SP)
1697 014346 012746 000003 MOV      #3,-(SP)
1698 014352 010600 MOV      SP,R0
1699 014354 104415 TRAP     C#PNTX
1700 014356 062706 000010 ADD      #10,SP
1701 014362 PRINTX  #PRIASC                                     ;PRINT HEADER FOR CONTENTS
1702 014362 012746 014545 MOV      #PRIASC,-(SP)
1703 014366 012746 000001 MOV      #1,-(SP)
1704 014372 010600 MOV      SP,R0
1705 014374 104415 TRAP     C#PNTX
1706 014376 062706 000004 ADD      #4,SP
1707 014402 005004 CLR      R4                                 ;NUMBER OF THE NEXT WORD
1708 014404 010501 MOV      R5,R1                               ;COPY LOW ORDER ADDRESS
1709 014406 010300 MOV      R3,R0                               ;COPY HIGH ORDER ADDRESS
1710 014410 001403 BEQ      204                                ;BR IF NOT ABOVE 28K
1711 014412 004737 017376 JSR      PC,SETMAP                     ;SETUP PAR ADDRESS IN R0
1712 014416 010005 MOV      R0,R5                               ;GET PAR FORMAT ADDRESS ABOVE 28K
1713 014420 204: PRINTX  #PRASC,R4,(R5)                         ;PRINT THE CONTENTS OF MEMORY BUFFER
1714 014420 012546 MOV      (R5),-(SP)
1715 014422 010446 MOV      R4,-(SP)
1716 014424 012746 014603 MOV      #PRASC,-(SP)
1717 014430 012746 000003 MOV      #3,-(SP)
1718 014434 010600 MOV      SP,R0
1719 014436 104415 TRAP     C#PNTX
1720 014440 062706 000010 ADD      #10,SP
1721 014444 005204 INC      R4                                 ;NUMBER OF THE NEXT
1722 014446 020427 000007 CMP      R4,#7                       ;DONE ALL YET ?
1723 014452 003011 BGT      504                                ;BRANCH IF ALL DONE
1724 014454 002761 BLT      204                                ;PRINT FIRST 7 WORDS
1725 014456 022737 000001 002226 CMP      #1,REV                ;IS IT A NEW MICROCODE
1726 014464 001355 BNE      204                                ;NO BR
1727 014466 032763 000200 000012 BIT      #X2.EXTF,XST2(R3)        ;EXTENDED FEATUTES ON ?
1728 014474 001351 BNE      204                                ;PRINT EXTENDED STATUS WORD
1729 014476 000207 504: RTS      PC                               ;RETURN
1730 014500 045 116 045 PROASC: .ASCIZ '##A Message Buffer Address = #01#05'

```

F6

TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 61-1

SEQ 0070

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

1712	014545	045	116	045	PR1ASC: .ASCIZ	'#N#A Message Buffer Contents:'
1713	014603	045	116	045	PRASC: .ASCIZ	'#N#A Word#D1#A: #0'

G6

SEQ 0071

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```

1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726 014632
1727 014632
1728 014636 010005
1729 014640 013700 002306
1730 014644 010004
1731 014646 013701 002304
1732 014652 006100
1733 014654 006101
1734 014656
    014656 010446
    014660 010146
    014662 012746 015012
    014666 012746 000003
    014672 010600
    014674 104415
    014676 062706 000010
1735 014702
    014702 012746 015057
    014706 012746 000001
    014712 010600
    014714 104415
    014716 062706 000004
1736 014722 005004
1737 014724 012701 002322
1738 014730 012702 002466
1739 014734 011100
1740 014736 011203
1741 014740
1742 014750
    014750 010346
    014752 012246
    014754 012146
    014756 010446
    014760 012746 015115
    014764 012746 000005
    014770 010600
    014772 104415
    014774 062706 000014
1743 015000 005204
1744 015002 020405
1745 015004 002001
1746 015006 000752
1747 015010 000207
1748 015012 045 116
1749 015057 045 116
1750 015115 045 116

          .EVEN
          .SBTTL PRMSGEXP - PRINT EXPD/RCV MESSAGE BUFFERS
          ;*
          ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
          ;RO - NUMBER OF WORDS IN BUFFER
          ;IMPLICIT INPUTS:
          ;EXPMSG - EXPECTED MESSAGE BUFFER
          ;RCMSG - RECEIVED MESSAGE BUFFER
          ;RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
          ;RCVLOAD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
          ;-
PRMSGEXP:
          SAVREG
          MOV RO,R5 ;SAVE R1-R5 UNTIL NEXT RETURN
          MOV RCVLOAD,RO ;SAVE NUMBER OF WORDS
          MOV RO,R4 ;GET RECV LOW ADDRESS
          MOV RCVHIADD,R1 ;COPY LOW ADDRESS
          ROL RO ;GET RECV HIGH ADDRESS
          ROL R1 ;SHIFT BIT15 TO C BIT
          PRINTX #PRMSG0,R1,R4 ;SHIFT TO HIGH ORDER FOR PRINTOUT
          MOV R4,-(SP) ;PRINT MESSAGE BUFFER ADDRESS
          MOV R1,-(SP)
          MOV #PRMSG0,-(SP)
          MOV #3,-(SP)
          MOV SP,RO
          TRAP C:PNTX
          ADD #10,SP
          PRINTX #PRMSG1 ;PRINT HEADER FOR CONTENTS
          MOV #PRMSG1,-(SP)
          MOV #1,-(SP)
          MOV SP,RO
          TRAP C:PNTX
          ADD #4,SP
          CLR R4 ;NUMBER OF THE CURRENT WORD
          MOV #EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
          MOV #RCMSG,R2 ;GET RECV BUFFER ADDRESS
204:      MOV (R1),RO ;GET EXPD
          MOV (R2),R3 ;GET RECV
          XOR RO,R3 ;XOR EXPD/RCV
          PRINTX #PRMSG2,R4,(R1),-(R2),R3
          MOV R3,-(SP)
          MOV (R2),-(SP)
          MOV (R1),-(SP)
          MOV R4,-(SP)
          MOV #PRMSG2,-(SP)
          MOV #5,-(SP)
          MOV SP,RO
          TRAP C:PNTX
          ADD #14,SP
          INC R4 ;NUMBER OF THE NEXT
          CMP R4,R5 ;DONE ALL YET?
          BGE 504 ;BR IF YES
          BR 204 ;DO ANOTHER
          RTS ;RETURN
          PC
          504:
          045 PRMSG0: .ASCIZ 'N#A Message Buffer Address = #01#05'
          045 PRMSG1: .ASCIZ 'N#A Message Buffer Contents:'
          045 PRMSG2: .ASCIZ 'N#A WORD #D2#A EXPD: #06#A RECV: #06#A XOR: #06'

```

H6

PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS

```

1752          .EVEN
1753          .SBTTL PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
1754
1755          ;*
1756          ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
1757          ;ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
1758
1759          ;RO      - NUMBER OF BYTES IN BUFFER
1760
1761          ;IMPLICIT INPUTS:
1762
1763          ;EXPMSG  - EXPECTED MESSAGE BUFFER
1764          ;RECMMSG - RECEIVED MESSAGE BUFFER
1765
1766          PRBYTEXP::
1767          SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
1768          MOV     R0,R5                        ;SAVE NUMBER OF BYTES
1769          CLR     PRMNO                         ;INIT ERROR COUNT
1770          CLR     R4                          ;NUMBER OF THE CURRENT BYTE
1771          MOV     @EXPMSG,R1                   ;GET EXPD BUFFER ADDRESS
1772          MOV     @RECMMSG,R2                  ;GET RECV BUFFER ADDRESS
1773          MOV     (R1),R0                      ;GET EXPD BYTE
1774          BIC     @C<377>,R0                   ;CLEAR UPPER BYTE
1775          MOV     R0,PRBEXP                    ;SAVE FOR ERROR REPORT
1776          MOV     (R2),R3                     ;GET RECV BYTE
1777          BIC     @C<377>,R3                   ;CLEAR UPPER BYTE
1778          MOV     R3,PRBREC                    ;FOR ERROR REPORT
1779          XOR     R0,R3                       ;XOR EXPD/RECV
1780          CMP     (R1)+,(R2)+                  ;EXPD = RECV?
1781          BEQ     304                          ;BR IF YES
1782          INC     PRMNO                        ;UPDATE ERROR COUNT
1783          CMP     PRMNO,#8.                    ;PRINTED 8?
1784          BHI     304                          ;BR IF YES
1785          PRINTX @PRBMSG,R4,PRBEXP,PRBREC,R3
1786          MOV     R3,-(SP)
1787          MOV     PRBREC,-(SP)
1788          MOV     PRBEXP,-(SP)
1789          MOV     R4,-(SP)
1790          MOV     @PRBMSG,-(SP)
1791          MOV     #5,-(SP)
1792          MOV     SP,R0
1793          TRAP    C#PNTX
1794          ADD     #14,SP
1795          FORCEXIT 504                          ;22D
1796          BR      354                          ;22D
1797
1798          304:  FORCERROR 274,NOTSSR            ;22D
1799          354:
1800          INC     R4                          ;NUMBER OF THE NEXT
1801          CMP     R4,R5                      ;DONE ALL YET?
1802          BGE     504                          ;BR IF YES
1803          BR      204                          ;DO ANOTHER
1804          504:  PRINTX @PRBTOT,PRMNO           ;PRINT TOTAL ERROR COUNT
1805          MOV     PRMNO,-(SP)
1806          MOV     @PRBTOT,-(SP)
1807          MOV     #2,-(SP)
1808          MOV     SP,R0

```


I6

TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 63-1

SEQ 0073

PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

```

1796 015406 104415
1797 015410 062706 000006
1798 015414 000207
1798 015416 045 116 045 PRBMSG: .ASCIZ 'N#A BYTE #D2#A EXPD: #03#A RECV: #03#A XOR: #03'
1799 015503 045 116 045 PRBTOT: .ASCIZ 'N#A NUMBER OF BYTES IN ERROR = #D2'
1800
1801 015550 000000 PRBEXP: .WORD 0 ;EXPD
1802 015552 000000 PRBREC: .WORD 0 ;RECV
1803 .SBTTL EXPREC - PRINT EXPD/RECV WORD DATA
1804
1805 ;*
1806 ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1807 ;
1808 ;INPUTS:
1809 ;
1810 ; R1 RECEIVED DATA
1811 ; R2 EXPECTED DATA
1812 ;
1813 ;-
1814
1815 015554 BGNMSG EXPREC
1816 015554 EXPREC: JSR PC,PRIXOR ;PRINT THE DATA
1817 015560 ENDMSG
1818 015560 L10017: TRAP C#MSG
1819 015560 .SBTTL EXPBREC - PRINT EXPD/RECV BYTE DATA
1820
1821 ;*
1822 ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
1823 ;
1824 ;INPUTS:
1825 ;
1826 ; R1 RECEIVED DATA BYTE
1827 ; R2 EXPECTED DATA BYTE
1828 ;
1829 ;-
1830
1831 015562 BGNMSG EXPBREC
1832 015562 EXPBREC: JSR PC,PRIBXOR ;PRINT THE DATA
1833 015566 ENDMSG
1834 015566 L10020: TRAP C#MSG
1835
1836 .SBTTL RAMERR - PRINT RAM AND PACKET DATA
1837
1838 ;*
1839 ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1840 ;
1841 ;INPUTS:
1842 ;
1843 ; R4 POINTER TO COMMAND PACKET
1844 ;

```

J6

RAMERR - PRINT RAM AND PACKET DATA

```

1845      ;IMPLICIT INPUTS:
1846      ;
1847      ;      RAMDATA      DATA AS READ FROM THE RAM
1848      ;      RAMSIZ      NUMBER OF BYTES IN PACKET
1849      ;                      IF RAMSIZ=0 THEN DEFAULT TO 8.
1850      ;
1851      ;IMPLICIT OUTPUTS:
1852      ;
1853      ;      RAMSIZ  SET TO 0
1854      ;
1855      ;
1856      015570      BGNMSG  RAMERR
1857      015570      RAMERR:: JSR      PC,PRAMPKT      ;PRINT RAM/PACKET DATA
1858      015574      004737  014046      ENDMSG
1859      015574      L10021:  TRAP      C4MSG
1860      015574      104423      .SBTTL  RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
1861      ;
1862      ;
1863      ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1864      ;
1865      ;INPUTS:
1866      ;
1867      ;      R4      POINTER TO COMMAND PACKET
1868      ;
1869      ;IMPLICIT INPUTS:
1870      ;
1871      ;      RAMDATA      DATA AS READ FROM THE RAM
1872      ;      RAMSIZ      NUMBER OF BYTES IN PACKET
1873      ;                      IF RAMSIZ=0 THEN DEFAULT TO 8.
1874      ;      ERRHI      HIGH ORDER TEST ADDRESS
1875      ;      ERRLO      LOW ORDER TEST ADDRESS
1876      ;
1877      ;IMPLICIT OUTPUTS:
1878      ;
1879      ;      RAMSIZ  SET TO 0
1880      ;
1881      ;
1882      015576      BGNMSG  RAMTADD
1883      015576      RAMTADD:: JSR      PC,PRITADD      ;PRINT TEST ADDRESS
1884      015602      004737  010274      JSR      PC,PRAMPKT      ;PRINT RAM/PACKET DATA
1885      015606      015606      ENDMSG
1886      015606      L10022:  TRAP      C4MSG
1887      015606      104423      .SBTTL  RAMEXP - PRINT RAM EXPD/RECV DATA
1888      ;
1889      ;
1890      ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1891      ;
1892      ;INPUTS:
1893      ;
1894      ;      R1      RECEIVED DATA
1895      ;      R2      EXPECTED DATA

```


K6

RAMEXP - PRINT RAM EXPD/RECV DATA

```

1896      ;      R4      CONTROLLER RAM ADDRESS
1897      ; -
1898
1899 015610      BGNMSG  RAMEXP
1900 015610      RAMEXP::
1901 015610 042701 177400      BIC      #C<377>,R1      ;SAVE EXPD RAM DATA BYTE
1902 015614 042702 177400      BIC      #C<377>,R2      ;SAVE EXPD RAM DATA BYTE
1903 015620 004737 010066      JSR      PC,PRIRAM      ;PRINT THE RAM ADDRESS
1904 015624 004737 007742      JSR      PC,PRIXOR      ;PRINT THE DATA
1905 015630      ENDMMSG
1906 015630 104423      L10023:      TRAP      C#MSG
1907      .SBTTL  TIMEXP - PRINT TIMER A,B AND EXP/REC
1908      ;*
1909      ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1910      ;AND TIMER A,B HEADER MESSAGE
1911      ;
1912      ;INPUTS:
1913      ;
1914      ;      R1      RECEIVED DATA
1915      ;      R2      EXPECTED DATA
1916      ; -
1917
1918 015632      BGNMSG  TIMEXP
1919 015632      TIMEXP::
1920 015632 012746 015660      PRINTX   #TIMSGO      ;PRINT HEADER
1921 015636 012746 000001      MOV      #TIMSGO,-(SP)
1922 015642 010600      MOV      #1,-(SP)
1923 015644 104415      MOV      SP,R0
1924 015646 062706 000004      TRAP     C#PNTX
1925 015652 004737 007742      ADD      #4,SP
1926 015656      JSR      PC,PRIXOR      ;PRINT THE DATA
1927 015656      ENDMMSG
1928 015656 104423      L10024:      TRAP      C#MSG
1929 015660 045 116 045 TIMSGO: .ASCIZ  'TIMER A STATUS IS IN BIT 3,TIMER B STATUS IS IN BIT 2'
1930 .EVEN
1931      .SBTTL  BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS
1932      ;*
1933      ;PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
1934      ;
1935      ;INPUTS:
1936      ;
1937      ;      R1      CONTENTS OF TSSR
1938      ;      R2      DATA WRITTEN (8 BITS)
1939      ; -
1940
1941 015760      BGNMSG  BADSSR
1942 015760      BADSSR::
1943 015760 010246      MOV      R2,-(SP)      ;SAVE DATA TRANSFERRED
1944 015762 042702 177400      BIC      #177400,R2      ;GET JUST ONE BYTE

```

L6

BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

1941	015766				PRINTB	#XFERASC,R2	
	015766	010246			MOV	R2,-(SP)	
	015770	012746	016020		MOV	#XFERASC,-(SP)	
	015774	012746	000002		MOV	#2,-(SP)	
	016000	010600			MOV	SP,R0	
	016002	104414			TRAP	C#PNTB	
	016004	062706	000006		ADD	#6,SP	
1942	016010	012602			MOV	(SP)+,R2	;RESTORE R2
1943	016012	004737	006024		JSR	PC,PRITSSR	;DECODE TSSR CONTENTS
1944	016016				ENDMSG		
	016016			L10025:			
	016016	104423			TRAP	C#MSG	
1945	016020	045	116	045	XFERASC:	.ASCIZ '#N#A Data Transferred = #03'	

GLOBAL SUBROUTINES SECTION

```

1947 .SBTTL GLOBAL SUBROUTINES SECTION
1948
1949
1950 ;**
1951 ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
1952 ; THAT ARE USED IN MORE THAN ONE TEST.
1953 ;--
1954 .SBTTL SOFINIT - SOFT INITIALIZE OF CONTROLLER
1955
1956 ;*
1957 ; ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
1958 ; BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
1959 ; THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
1960 ; DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
1961
1962 ; INPUTS:
1963 ;
1964 ; R5 ADDRESS OF FIRST REGISTER
1965
1966 ; OUTPUTS:
1967 ;
1968 ; R0 CONTENTS OF TSSR, IF ERROR
1969 ; CARRY SET IF INIT WAS OKAY
1970 ; CLEAR IF FATAL ERROR
1971
1972 ; CALLING SEQUENCE:
1973 ;
1974 ; MOV #ADDRESS,R5
1975 ; JSR PC,SOFINIT
1976 ; BCS CONTINUE
1977 ; ERRDF ;REPORT FATAL ERROR
1978
1979 ;-
1980
1981 SOFINIT::
1982 SAVREG
1983 MOV #0,TSSR(R5) ; SAVE THE REGISTERS
1984 JSR PC,WAITF ; DO THE INIT.
1985 MOV TSSR(R5),R0 ; WAIT FOR SSR
1986 MOV R0,R4 ; GET THE TSSR REGISTER
1987 BIC #1<HIADDR!OFL>,R4 ; TSSR CONTENTS
1988 BIS #SSR!NBA,R4 ; R4 HAS EXPECTED CONTENTS
1989 CMP R4,R0 ; ONLY EXPECTED BITS SET ?
1990 BEQ 54 ; BRANCH IF OKAY
1991 CLC ; CLEAR THE CARRY FOR ERROR
1992 BR 104 ; GO TO EXIT
1993 54: SEC ; SET THE CARRY BIT
1994 104: RTS PC ; RETURN TO CALLER

```

N6

TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 65

SEQ 0078

CHKAMB - CHECK TSSR FOR AMBIGUITY

```

1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016 016124
2017 016124
2018 016130 010004
2019 016132 032700 100000
2020 016136 001004
2021 016140 032700 174077
2022 016144 001023
2023 016146 000424
2024 016150 032700 000200
2025 016154 001011
2026 016156 032700 000040
2027 016162 001414
2028 016164 042704 177761
2029 016170 020427 000016
2030 016174 001007
2031 016176 000410
2032 016200 032700 000040
2033 016204 001405
2034 016206 032700 000006
2035 016212 001002
2036 016214 000241
2037 016216 000401
2038 016220 000261
2039 016222 000207

.SBTTL  CHKAMB - CHECK TSSR FOR AMBIGUITY
;+
;THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
;FOR AMBIGUITY
;INPUT:
;      RO      CONTENTS OF TSSR
;OUTPUT:
;      RO      CONTENTS OF TSSR
;      CARRY   SET - NO AMBIGUITY
;              CLR - AMBIGUOUS CONTENTS
;-

CHKAMB:
    SAVREG      ;SAVE THE GENERAL REGISTERS
    MOV         RO,R4      ;CONTENTS OF TSSR
    BIT         #SC,RO     ;IS BIT 15 SET ?
    BNE         54         ;BRANCH IF YES
    BIT         #1C<NBA!OFL!SSR!HIADDR>,RO ;ANY OTHER BITS SET ?
    BNE         40         ;MUST BE AN ERROR
    BR          45         ;RETURN WITH SUCCESS
    BIT         #SSR,RO     ;IS READY BIT SET ?
    BNE         104        ;BRANCH IF READY BIT IS SET.
    BIT         #BITS,RO    ;IS FATAL ERROR BIT SET ?
    BEQ         40         ;ERROR IF NOT
    BIC         #1CTERCLS,R4 ;CLEAR ALL BUT TERMINATION CODE
    CMP         R4,#16     ;ALL THREE BITS MUST BE SET
    BNE         40         ;ERROR IF NOT SET
    BR          45         ;OK IF ALL ARE SET
    BIT         #BITS,RO    ;IS FATAL ERROR BIT SET ?
    BEQ         45         ;ERROR IF BIT IS SET WITH SSR
    BIT         #BIT2!BIT1,RO ;IS THIS A FUNCTION REJECT
    BNE         45         ;BR, IF TSSR IS OK
    CLC          ;AMBIGUOUS CONTENTS
    BR          50         ;
    SEC          ;SHOW SUCCESS - NO AMBIGUITY
    RTS         PC        ;RETURN TO CALLER
54:
104:
404:
454:
504:

```


B7

ENAIN,DSBINT - ENABLE/DISABLE INTERRUPTS

```

2041 .SBTTL ENAIN,DSBINT - ENABLE/DISABLE INTERRUPTS
2042 ;
2043 ; DEFAULT DISPLAY INTERRUPT HANDLERS.
2044 ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
2045 ; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
2046 ;
2047 ;
2048 ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
2049 ;
2050 ; IOKCKIN=BIT7 ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
2051 ; IOKSTP=BIT0 ; EXPECT "STOP" INTERRUPT.
2052 ;
2053 ; INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
2054 016224 000 INTMASK: .BYTE 0
2055 ; INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
2056 016225 000 INTFLAG: .BYTE 0
2057 ;
2058 ; SAVED INTERRUPT VECTOR:
2059 016226 000000 INTVEC: .WORD 0
2060 ; SAVE CPU PC
2061 016230 000000 INTCPC: .WORD 0
2062 ;
2063 ; SUBROUTINE TO ENABLE INTERRUPTS:
2064 016232 010046 ENAIN: MOV RO, -(SP) ;SAVE RO
2065 016234 013700 002206 MOV IVEC, RO ;GET POINTER TO VECTORS
2066 016240 012720 016276 MOV #INTR, (RO) ;SET UP INTERRUPT VECTOR
2067 016244 012720 000340 MOV #PRI07, (RO)
2068 016250 012600 MOV (SP), RO ;RESTORE RO
2069 016252 011646 MOV (SP), -(SP)
2070 016254 012766 000000 000002 MOV #0,2(SP) ;SET CPU TO LEVEL 0
2071 016262 000002 RTI
2072 ;
2073 ; SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
2074 016264 011646 DSBINT: MOV (SP), -(SP)
2075 016266 012766 000340 000002 MOV #PRI07,2(SP)
2076 016274 000002 RTI
2077 .SBTTL INTR - INTERRUPT HANDLERS
2078 ;
2079 016276 BGNSRV INTR ;DEFINE INTERRUPT ENTRY
2080 016276 012737 000001 002222 INTR:: MOV #1,INTRECV ;SET FLAG TO SHOW INTERRUPT RECEIVED
2081 016304 105037 016225 CLR B INTFLAG ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2082 016310 132737 000001 016224 BIT B #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
2083 016316 001003 BNE 1 ;BR IF YES
2084 016320 152737 000001 016225 BIS B #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.
2085 ;
2086 ; SAVE REGISTERS, MSG BUFFER, ETC.
2087 016326 1:
2088 016326 ENDSRV
016326 L10026:
016326 000002 RTI

```

C7

WAITF - WAIT FOR SUBSYSTEM READY

```

2090          .SBTTL WAITF - WAIT FOR SUBSYSTEM READY
2091          ;
2092          ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
2093          ;
2094          ; INPUTS:
2095          ;
2096          ; R5      ADDRESS OF FIRST DEVICE REGISTER
2097          ;
2098          ; OUTPUTS:
2099          ;
2100          ; R0      CONTENTS OF LAST TSSR READ
2101          ; CARRY   SET - READY BIT SET
2102          ;         CLR - TIMEOUT WAITING FOR READY
2103          ;
2104          016330 000401 WAITF:: BR      1#          ;NOP WHEN SUPER FIXED
2105          016332          BREAK          ; DO A SUPVSR BREAK FIRST.
2106          016332 104422          TRAP     C#BRK
2107          016334 012746 011000 1#:      MOV     #11000,-(SP) ;25-APRIL-83 REV B - 1100 MSEC TIMER
2108          016340 016500 000002 2#:      MOV     TSSR(R5),R0 ;READ THE TSSR REGISTER
2109          016344 105700          TSTB     R0 ;TEST FOR READY BIT SET
2110          016346 100420          BMI      3#          ; EXIT ON STOP FLAG.
2111          016350          DELAY     1 ; WAIT 100 USEC
2112          016350 012727 000001          MOV     #1,(PC)+
2113          016354 000000          .WORD     0
2114          016356 013727 002116          MOV     L#DLY,(PC)+
2115          016362 000000          .WORD     0
2116          016364 005367 177772          DEC     -6(PC)
2117          016370 001375          BNE      -.4
2118          016372 005367 177756          DEC     -22(PC)
2119          016376 001367          BNE      -.20
2120          016400 005316          DEC     (SP) ;REDUCE DELAY COUNT
2121          016402 001356          BNE      2#          ;RETRY UNTIL TIMER EXPIRES
2122          016404 000241          CLC
2123          016406 000401          BR       4#          ; C = 0, CONTROLLER STILL RUNNING...
2124          016410 000261          3#:      SEC
2125          016412 005326          4#:      DEC     (SP)+ ;...OR HUNG-UP AFTER 300 MSEC.
2126          016414 000207          RTS      PC ; C = 1, CONTROLLER IS STOPPED.
2127          ;RESTORE STACK WITHOUT CHANGING CARRY BIT

```


D7

CHKTSSR - CHECK TSSR FOR READY

```

2120 .SBTTL CHKTSSR - CHECK TSSR FOR READY
2121 ;*
2122 ;THIS ROUTINE WAITS FOR READY IN THE TSSR
2123 ;AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
2124 ;
2125 ;INPUT:
2126 ;      R5      ADDRESS OF CSR REGISTERS
2127 ;
2128 ;OUTPUT:
2129 ;      R0      CONTENTS OF TSSR
2130 ;      CARRY   SET - OKAY
2131 ;             CLR - NOT READY AMBIGUOUS, OR SC SET
2132 ;
2133 CHKTSSR:
2134 JSR PC, WAITF      ;WAIT FOR READY
2135 BCC 20$            ;BRANCH IF TIME OUT
2136 JSR PC, CHKAMB    ;TSSR AMBIGUOUS?
2137 BCC 10$           ;BR IF YES
2138 BIT #SC, R0       ;SPECIAL CONDITION SET?
2139 BEQ 15$           ;BR IF NO
2140 BIT #<SCC!BIE!RMR!NXM>, R0 ;ANY ERROR BITS SET?
2141 BEQ 15$           ;BR IF NO
2142 CLC               ;SET FAILURE
2143 BR 20$            ;
2144 SEC               ;SET SUCCESS
2145 RTS PC            ;RETURN TO CALLER
2146 .SBTTL XNXM      - CHECK FOR NONEXISTENT MEMORY
2147 ;*
2148 ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
2149 ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
2150 ; "C" = 0, ALL ADDRESSES OK.
2151 ;
2152 ;CALL: MOV ADR1, R1
2153 ;      MOV ADR2, R2
2154 ;      JSR PC, NXM
2155 ;      RETURN
2156 XNXM: MOV #2$, #04 ;TEST "C" AND PROCEED.
2157 MOV #PRI04, #06 ; SET BUSERR VECTOR.
2158 CLR R3
2159 TST (R1)
2160 ;FLAG.
2161 CMP R1, R2
2162 BEQ 3$
2163 ADD #2, R1
2164 BR 1$
2165 COM R3
2166 MOV #3$, (SP)
2167 RTI
2168 CLRVEC #4
2169 MOV #4, R0
2170 TRAP C#CVEC
2171 TST R3
2172 BEQ .+4
2173 SEC
2174 RTS PC

```

;TEST THE ADDRESS(ES).
 ;IF ANY TRAP, CONTINUE AT 2\$.
 ;OTHERWISE, CONTINUE HERE.
 ;BR IF FINISHED (NO NEXM'S).
 ;SET NEXT ADDRESS...
 ;...AND CONTINUE.
 ;GOT ONE, SET FLAG...
 ;...AND DISMISS INTERRUPT...
 ;...AND GIVE BACK THE VECTOR.
 ;DID WE CATCH ONE ??
 ;NO, "C" = 0, SKIP NEXT.
 ;YES, "C" = 1, (R1) = NEXM ADDR.

E7

SEQ 0082

TSTLOOP - CHECK ITERATION COUNT

```

2174                                     .SBTTL  TSTLOOP - CHECK ITERATION COUNT
2175                                     ;*
2176                                     ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
2177                                     ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
2178                                     ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
2179                                     ;
2180                                     ; CALL: LOOPTO  ARG
2181                                     ;
2182 016536                                TSTLOOP:
2183 016536 005737 002166                 TST      NOITS          ; ITERATIONS INHIBITED?
2184 016542 001006                       BNE      1$            ; YES.
2185 016544 005737 002202                 TST      QVP           ; NO.
2186 016550 100403                       BMI      1$            ; LOOPS DISALLOWED IN QUICK PASS.
2187 016552 005337 002214                 DEC      LOOPCNT      ; BUMP LOOP COUNTER.
2188 016556 001002                       BNE      2$
2189 016560 000241                       1$: CLC              ; LOOP DISALLOWED, OR DONE.
2190 016562 000401                       BR       3$
2191 016564 000261                       2$: SEC              ; LOOP ENABLED.
2192 016566 000207                       3$: RTS      PC
2193
2194                                     .SBTTL  TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
2195                                     ;*
2196                                     ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
2197                                     ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
2198                                     ; IN THE CURRENT RUN SEQUENCE.
2199                                     ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
2200                                     ;
2201                                     ; INPUT:
2202                                     ;
2203                                     ;      R0      POINTER TO TEST ID ASCIZ STRING
2204                                     ;
2205                                     ; OUTPUT:
2206                                     ;
2207                                     ;      R5      ADDRESS OF FIRST DEVICE REGISTER
2208                                     ;
2209                                     ; IMPLICIT OUTPUTS:
2210                                     ;
2211                                     ;      TSTCNT  UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
2212                                     ;
2213                                     ; SIDE EFFECTS:
2214                                     ;
2215                                     ;      INTERRUPT LEVEL IS RASIED TO LEVEL OF
2216                                     ;      THE DEVICE UNDER TEST
2217                                     ;
2218                                     ; -
2219
2220 016570                                TSTSETUP:
2221 016570 010046                        MOV      R0, -(SP)      ; SAVE THE TEST ID MESSAGE
2222 016572 005037 003154                 CLR      SIFLAG        ; CLEAR "SOFT INIT" FLAG
2223 016576 005037 017036                 CLR      ERRK         ; CLEAR LOCAL ERROR COUNTER.
2224 016602 005037 005772                 CLR      EXTA        ; CLEAR ERROR EXTENSION FLAG.
2225 016606 105037 016224                 CLRB     INTMASK      ; CLEAR INTERRUPT MASK (CHECK ERROR)
2226 016612 013700 002200                 MOV      UNITN, R0     ; GET THE UNIT NUMBER.
2227 016616 006300                       ASL      R0            ; ... AND MAKE IT A WORD OFFSET.
2228 016620 005737 003114                 TST      NODEV        ; DID STARTUP FIND THE DEVICE?
2229 016624 001430                       BEQ      4$            ; BR IF YES
2230 016626 100010                       BPL      3$            ; BR IF NOT IDLE

```


F7

TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS

```

2231 016630 052760 160000 003176      BIS      #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
2232 016636      ERRDF  1,NXR,NXRERR ; NO DEVICE HERE -- PRINT IT
      016636      TRAP   C#ERRDF
      016640      .WORD  1
      016642      .WORD  NXR
      016644      .WORD  NXRERR
2233 016646 000407      BR      2#
2234 016650 052760 160001 003176 3# : BIS      #160001,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
2235 016656      ERRDF  2,NOINIT ; DEVICE NOT IDLE
      016656      TRAP   C#ERRDF
      016660      .WORD  2
      016662      .WORD  NOINIT
      016664      .WORD  0
2236 016666 012737 177777 003112 2# : MOV      #-1,DUFLG ; DROP THE UNIT
2237 016674      DODU    UNITN
      016674      MOV     UNITN,R0
      016700      TRAP   C#DODU
2238 016702      DOCLN   ; ABORT THE PASS
      016702      TRAP   C#DCLN
2239 016704 000423      BR      5#
2240
2241 016706      4# : RFLAGS  R0 ; GET THE OPERATOR FLAGS.
      016706      TRAP   C#RFLA
2242 016710 032700 001000      BIT      #PNT,R0 ; PRINT THE TEST NUMBERS?
2243 016714 001412      BEQ      1# ; BR IF NO
2244 016716 011600      MOV      (SP),R0 ; GET THE ID MESSAGE
2245 016720      PRINTF  #TNAM,R0 ; DISPLAY THE TEST ID
      016720      MOV     R0,-(SP)
      016722      MOV     #TNAM,-(SP)
      016726      MOV     #2,-(SP)
      016732      MOV     SP,R0
      016734      TRAP   C#PNTF
      016736      ADD     #6,SP
2246 016742 005237 002212      1# : INC      TSTCNT ; BUMP TEST COUNTER.
2247 016746      SETPRI  IPRI ; PRIORITY THAT OF DEVICE
      016746      MOV     IPRI,R0
      016752      TRAP   C#SPRI
2248 016754 005726      5# : TST      (SP) ; FIX UP THE STACK
2249 016756 013705 002204      MOV     CSRADDR,R5 ; ADDRESS OF TSV REGISTERS ON UNIBUS
2250 016762 000207      RTS      PC
2251 016764      045      123      045  TNAM: .ASCIZ  '#S#T#A Test'
2252
2253      .EVEN
2254      .SBTTL  TSTEND - PRINT ERRORS RECEIVED
2255
2256      ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
2257      ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
2258
2258 017000      TSTEND: RFLAGS  R0
      017000      TRAP   C#RFLA
2259 017002 030027 020000      BIT      R0,#IER
2260 017006 001412      BEQ      1# ; BR IF "IER" NOT SET.
2261 017010      PRINTF  #ESUM,ERRK ; PRINT ERROR COUNT.
      017010      MOV     ERRK,-(SP)
      017014      MOV     #ESUM,-(SP)
      017020      MOV     #2,-(SP)
      017024      MOV     SP,R0
      017026      TRAP   C#PNTF

```

G7

TSTEND - PRINT ERRORS RECEIVED

2262	017030	062706	000006			ADD	#6.SP	
2263	017034	000207		14:		RTS	PC	
2264	017036	000000				ERRK:	0	: LOCAL ERROR COUNT.
2265	017040	045	101	040	ESUM:	.ASCIZ	/#A #D#A ERRORS/	
2266	017057	105	122	122	EMAXDU:	.ASCIZ	/ERROR LIMIT REACHED -- DROPPING UNIT/	
2267						.EVEN		

H7

INCERK - INCREMENT LOCAL ERROR COUNT

```

2269                                     .SBTTL  INCERK - INCREMENT LOCAL ERROR COUNT
2270                                     ;*
2271                                     ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
2272                                     ;
2273 017124 005237 017036 INCERK: INC  ERRK      ; INCREMENT LOCAL ERROR COUNT
2274 017130 010046      MOV  RO,-(SP)    ; SAVE RO
2275 017132 013700      MOV  UNITN,RO    ; GET UNIT NUMBER,
2276 017136 006300      ASL   RO          ; ... AND MAKE IT A WORD OFFSET.
2277 017140 062700      ADD  @ERTABL,RO  ; RO GETS ADDRESS OF ERROR TABLE ENTRY.
2278 017144 005210      INC  (RO)        ; INCREMENT THE DEVICE ERROR COUNT
2279 017146 032710      BIT   @7777,(RO) ; DID WE OVERFLOW THE FIELD?
2280 017152 001001      BNE   1$         ; BR IF NO.
2281 017154 005310      DEC  (RO)        ; YES -- BACK IT UP TO 7777.
2282 017156 012600      1$: MOV  (SP)+,RO ; RESTORE RO
2283 017160 000207      RTS   PC         ; RETURN TO CALLER.
2284
2285 017162 010046      CKEMAX: MOV  RO,-(SP) ; SAVE RO
2286 017164 013700      MOV  UNITN,RO    ; GET UNIT NUMBER
2287 017170 006300      ASL   RO          ; ... AND MAKE IT A WORD OFFSET
2288 017172 016000      MOV  ERTABL(RO),RO ; GET ERROR TABLE ENTRY
2289 017176 042700      BIC   @170000,RO  ; EXTRACT ERROR COUNT FIELD
2290 017202 020037      CMP   RO,GERRMAX  ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
2291 017206 103004      BHIS  1$         ; BR IF YES
2292 017210 023737      CMP   ERRK,LERRMAX ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
2293 017216 103417      BLO   2$         ; BR IF NO
2294 017220      1$:  RFLAGS RO          ; GET OPERATOR FLAGS
2295 017222 104421      TRAP  C$RFLA
2296 017226 032700      BIT   @IDU,RO    ; IS DROPPING INHIBITED?
2297 017230 001013      BNE   2$         ; BR IF YES.
2298 017236 012737      MOV  #-1,DUFLG   ; NO -- DROP THE UNIT
2299 017246 017236 104455      ERRDF 4,EMAXDU
2300 017254 017240 000004      TRAP  C$ERDF
2301 017256 017242 017057      .WORD 4
2302 017260 000000      .WORD  EMAXDU
2303 017262 000000      .WORD  0
2304 017264 017246 013700 002200      DODU UNITN
2305 017266 017252 104451      MOV  UNITN,RO
2306 017268 017254 104444      TRAP  C$DODU
2307 017270 017256 012600      DOCLN
2308 017272 000207      TRAP  C$DCLN
2309 017274 000207      2$: MOV  (SP)+,RO ; RESTORE RO
2310 017276 000207      RTS   PC         ; RETURN TO CALLER

```

I7

CKDROP - CHECK IF UNIT SHOULD BE DROPPED

```

2304                                     .SBTTL CKDROP - CHECK IF UNIT SHOULD BE DROPPED
2305                                     ;*
2306                                     ; CHECK IF UNIT SHOULD BE DROPPED
2307                                     ;-
2308 017262 010046 CKDROP: MOV      R0,-(SP)
2309 017264 FORCERROR 1$,NOTSSR
2310 017274 RFLAGS R0
2311 017276 104421 TRAP      C$RFLA
2312 017302 032700 000040 BIT      $IDU,R0
2313 017304 001010 BNE      1$
2314 017306 012737 177777 003112 MOV      (SP),R0
2315 017314 013700 002200 MOV      #-1,DUFLG
2316 017322 104451 DODU      UNITN
2317 017322 104444 TRAP      C$DODU
2318 017324 012600 1$:      TRAP      C$DCLN
2319 017326 000207 MOV      (SP)+,R0
2320                                     RTS      PC
2321                                     .SBTTL CONFIG - DETERMINE CONFIGURATION OF SYSTEM
2322                                     ;
2323                                     ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
2324                                     ;
2325 017330 CONFIG: JSR      PC,SOFINIT
2326 017330 004737 016054 RTS      PC
2327 017334 000207 .SBTTL KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT
2328                                     ;
2329                                     ; SUBROUTINE - ENABLE MEM MGT.
2330                                     ;
2331 017336 005737 003132 KTON: TST      KTFLG
2332 017342 001403 BEQ      1$
2333 017344 012737 000001 177572 MOV      $1,SRO
2334 017352 000207 1$:      RTS      PC
2335                                     ;
2336                                     ; SUBROUTINE - DISABLE MEM MGT.
2337                                     ;
2338 017354 005737 003132 KTOFF: TST      KTFLG
2339 017360 001405 BEQ      1$
2340 017362 000240 NOP
2341 017364 000240 NOP
2342 017366 012737 000000 177572 MOV      $0,SRO
2343 017374 000207 1$:      RTS      PC

```

;ABORT THE PASS

- ENABLE/DISABLE MEMORY MANAGEMENT

; GOT KT?
; NO.
; YES. ENABLE KT11.

; GOT KT11?
; NO.

; DISABLE KT.

J7

SETMAP - SETUP PAR6 MAPPING

```

2347                                     .SBTTL  SETMAP - SETUP PAR6 MAPPING
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366 017376
2367 017376
2368 017402 005737 003132
2369 017406 001433
2370 017410 010102
2371
2372
2373
2374
2375 017442 042701 000177
2376 017446 020137 003132
2377 017452 103011
2378 017454 010137 172354
2379 017460 042702 160000
2380 017464 062702 140000
2381 017470 010200
2382 017472 000261
2383 017474 000401
2384 017476 000241
2385 017500 000207
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401 017502
2402 017502
2403 017506 004737 017354

```

```

; *
; THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
; AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
; IS RETURNED BIASED TO PAR6.
; INPUTS:
; R0      HIGH ORDER ADDRESS BITS
; R1      LOW ORDER ADDRESS BITS
; OUTPUTS:
; R0      OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
; CARRY   SET IF SUCCESS
;         CLR IF ERROR
; SETMAP:
; SAVREG
; TST      KTFLG
; BEQ      10$
; MOV      R1,R2
; .REPT    6
; ASR      R0
; ROR      R1
; .ENDR
; BIC      #177,R1
; CMP      R1,KTFLG
; BHIS     10$
; MOV      R1,#KIPAR6
; BIC      #160000,R2
; ADD      #140000,R2
; MOV      R2,R0
; SEC
; BR       15$
; 10$:    CLC
; 15$:    RTS      PC
; .SBTTL  FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN
; *
; FILL MEMORY WITH A BACKGROUND PATTERN
; INPUTS:
; R0 = BACKGROUND PATTERN
; FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
; KTFLG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
; OUTPUTS:
; NONE
; FILLMEM:
; SAVREG
; JSR      PC,KTOFF
; SAVE R1-R4 UNTIL NEXT RETURN
; SYSTEM HAVE ABOVE 28K?
; BR IF NO
; SAVE LOW ORDER BITS
; CONVERT WORD ADDRESS TO 32W BLOCKS
; MAKE IT DOUBLE PRECISION
; ALINE FOR LOWER 4K BOUNDARY
; HIGHER THAN EXISTING MEMORY?
; BR IF YES
; SETUP MAPPING REGISTER PAR6
; SETUP DISPLACEMENT IN PAGE
; ADD IN PAR6 BIAS
; RETURN IN R0
; SET SUCCESS
; SET FAILURE
; RETURN

```

K7

TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 72-1

SEQ 0088

FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN

2404	017512	010003			MOV	R0,R3	;COPY TEST PATTERN
2405	017514	013701	003124		MOV	FREE,R1	;GET FIRST FREE LOCATION
2406	017520	013702	003126		MOV	FRESIZ,R2	;SIZE OF FREE SPACE BELOW 28K.
2407	017524	010321		104:	MOV	R3,(R1)+	;STORE A BACKGROUND WORD
2408	017526	005302			DEC	R2	;DONE ALL MEMORY IN FREE SPACE?
2409	017530	003375			BGT	104	;BR IF NO
2410	017532	005737	003132		TST	KTFLG	; GOT KT?
2411	017536	001477			BEQ	554	; NO. GET OUT.
2412	017540	004737	017336		JSR	PC,KTON	; YES. ENABLE KT.
2413	017544	005000			CLR	R0	;HIGH ORDER ADDRESS START
2414	017546	013701	003152		MOV	PST32W,R1	;GET >28K START ADDRESS (IN 32W BLOCKS)
2415		000006			.REPT	6	
2416					CLC		;CLEAR C BIT
2417					ROL	R1	;CONVERT BLOCKS TO WORDS
2418					ROL	R0	;MAKE IT DOUBLE PRECISION
2419					.ENDR		
2420	017616	004737	017376		JSR	PC,SETMAP	;SETUP PAR6 MAPPING REGISTER
2421	017622	010320		304:	MOV	R3,(R0)+	;STORE TEST PATTERN IN >28K ADDRESS
2422	017624	020027	160000		CMP	R0,#160000	;END OF PAR6 MAPPING AREA?
2423	017630	103774			BLO	304	;BR IF NO
2424	017632	162700	020000		SUB	#20000,R0	;BACKUP INTO PAR6 MAPPING BEGIN
2425	017636	062737	000200	172354	ADD	#200,@#KIPAR6	;POINT TO NEXT 4K BLOCK >28K.
2426	017644	023737	172354	003132	CMP	@#KIPAR6,KTFLG	;END OF MEMORY?
2427	017652	001427			BEQ	504	;BR IF YES
2428	017654	005737	003144		TST	T23A	;11/23A?
2429	017660	001407			BEQ	354	;NO KEEP GOING
2430	017662	013704	177572		MOV	SR0,R4	;GET SR0 CONTENTS
2431	017666	042704	177761		BIC	#177761,R4	;CLEAR ALL BUT PAGE NUMBER
2432	017672	022704	000016		CMP	#16,R4	;SEE IF PAGE 7
2433	017676	001415			BEQ	504	;EXIT IF THERE
2434	017700	005737	003146	354:	TST	T23B	;11/23B?
2435	017704	001410			BEQ	454	;NO KEEP GOING
2436	017706	023727	172354	007600	CMP	@#KIPAR6,#7600	;REACHED 18 BITS?
2437	017714	103001			BHIS	404	;YES
2438	017716	000403			BR	454	;NO KEEP GOING
2439	017720	012737	000020	172516	MOV	#20,SR3	;SET 22 BIT RELOCATION
2440	017726	000137	017622	454:	JMP	304	;KEEP GOING ON ETC.
2441	017732	004737	017354	504:	JSR	PC,KTOFF	; DISABLE KT.
2442	017736	000207		554:	RTS	PC	

L7

CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN

```

2444      .SBTTL  CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN
2445
2446      ;*
2447      ; COMPARE MEMORY WITH A BACKGROUND PATTERN
2448      ;
2449      ; INPUTS:
2450      ;
2451      ;     RO = BACKGROUND PATTERN
2452      ;     FREE  = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2453      ;     KTFLG  = SET TO HIGHEST MEMORY LOCATION IF > 28K.
2454      ;
2455      ; OUTPUTS:
2456      ;
2457      ;     CARRY  - SET IF NO ERROR
2458      ;     CARRY  - CLR IF ERROR
2459      ;
2460      ; IMPLICIT OUTPUTS:
2461      ;
2462      ;     ERRHI  - ERROR HIGH ADDRESS
2463      ;     ERRLO  - ERROR LOW ADDRESS
2464      ;     EXPD   - EXPECTED DATA
2465      ;     RECV   - RECEIVED DATA
2466      ;
2467      ;-
2468      CMPMEM:
2469      SAVREG
2470      MOV     RO,R3
2471      JSR     PC,KTOFF
2472      MOV     FREE,R1
2473      MOV     FRESIZ,R2
2474      CMP     R3,(R1)
2475      BEQ     154
2476      MOV     R1,ERRLO
2477      CLR     ERRHI
2478      MOV     R3,EXPD
2479      MOV     (R1),RECV
2480      BR      504
2481      TST     (R1)
2482      DEC     R2
2483      BGT     104
2484      TST     KTFLG
2485      BEQ     554
2486      JSR     PC,KTON
2487      CLR     RO
2488      MOV     PST32W,R1
2489      REPT     6
2490      ROL     R1
2491      ROL     RO
2492      ENDR
2493      BIC     #177,R1
2494      MOV     RO,-(SP)
2495      MOV     R1,-(SP)
2496      JSR     PC,SETHAP
2497      MOV     RO,R4
2498      MOV     (SP),R1
2499      MOV     (SP),R0
2500      CMP     R3,(R4)
2501      BEQ     324
2502      MOV     RO,ERRHI

```

;SAVE R1-R5 UNTIL NEXT RETURN
 ;COPY TEST PATTERN
 ;DISABLE KT.
 ;GET FIRST FREE LOCATION
 ;SIZE OF FREE SPACE BELOW 28K.
 ;FREE SPACE LOCATION EQUAL TO EXPD?
 ;BR IF YES
 ;SAVE ADDRESS IN ERROR
 ;NO HIGH ADDRESS
 ;SAVE EXPD FOR ERROR REPORT
 ;SAVE RECV FOR ERROR REPORT
 ;
 ;POINT TO NEXT ADDRESS
 ;DONE ALL MEMORY IN FREE SPACE?
 ;BR IF NO
 ; GOT KT?
 ; NO. GET OUT.
 ; YES. ENABLE KT.
 ;HIGH ORDER ADDRESS START
 ;GET >28K START ADDRESS (IN 32W BLOCKS)
 ;
 ;CONVERT BLOCKS TO WORDS
 ;MAKE IT DOUBLE PRECISION
 ;
 ;ALINE 4K BOUNDARY
 ;SAVE HIGH ORDER
 ;SAVE LOW ORDER
 ;SETUP PAR6 MAPPING REGISTER
 ;COPY ADDRESS BIASED TO PAR6
 ;RESTORE LOW ORDER IN NON PAR6 FORMAT
 ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
 ;ABOVE 28K LOCATION EQUAL EXPD?
 ;BR IF YES
 ;SAVE HIGH ORDER IN ERROR

M7

TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 73-1

SEQ 0090

CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN

```

2501 020120 010137 002240      MOV      R1,ERRLO      ;SAVE LOW ORDER IN ERROR
2502 020124 010337 002232      MOV      R3,EXPD      ;SAVE EXPD FOR ERROR REPORT
2503 020130 011437 002234      MOV      (R4),RECV      ;SAVE RECV FOR ERROR REPORT
2504 020134 000421              BR          50$          ;
2505 020136 062701 000002      32$:  ADD      #2,R1      ;UPDATE NON PAR6 ADDRESS
2506 020142 005500              ADC      R0          ;MAKE IT DOUBLE PRECISION ADD
2507 020144 062704 000002      ADD      #2,R4      ;UPDATE PAR FORMAT ADDRESS
2508 020150 020427 160000      CMP      R4,#160000    ;END OF PAR6 MAPPING AREA?
2509 020154 103755              BLO      30$          ;BR IF NO
2510 020156 162704 020000      SUB      #20000,R4      ;BACKUP INTO PAR6 MAPPING BEGIN
2511 020162 062737 000200 172354  ADD      #200,#KIPAR6    ;POINT TO NEXT 4K BLOCK >28K.
2512 020170 023737 172354 003132  CMP      #KIPAR6,KTFLG    ;END OF MEMORY?
2513 020176 101744              BLOS     30$          ;BR IF NO
2514 020200 004737 017354      50$:  JSR      PC,KTOFF      ;TURN OFF MEMORY MAPPING
2515 020204 000241              CLC          ;SET FAILURE
2516 020206 000403              BR          60$          ;
2517 020210 004737 017354      55$:  JSR      PC,KTOFF      ;TURN OFF MEMORY MAPPING
2518 020214 000261              SEC          ;SET SUCCESS
2519 020216 000207      60$:  RTS       PC
2520                      .SBTTL   REGSAV - SAVE R1-R5 ON STACK
2521                      ;*
2522                      ;
2523                      ;ROUTINE TO
2524                      ;SAVE R1 THROUGH R5 ON THE STACK
2525                      ;
2526                      ;CALLING SEQUENCE:
2527                      ;
2528                      ;      JSR      R5,REGSAV
2529                      ;
2530                      ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
2531                      ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
2532                      ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
2533                      ;REGISTERS.
2534                      ;
2535                      ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
2536                      ;CALLED VIA A JSR PC INSTRUCTION
2537                      ;
2538                      ;-
2539                      ;
2540                      REGSAV:
2541 020220 010446      MOV      R4,-(SP)
2542 020222 010346      MOV      R3,-(SP)
2543 020224 010246      MOV      R2,-(SP)
2544 020226 010146      MOV      R1,-(SP)
2545 020230 010546      MOV      R5,-(SP)
2546 020232 016605 000012      MOV      10.(SP),R5
2547 020236 004736      JSR      PC,@(SP)+
2548 020240 012601      MOV      (SP)+,R1
2549 020242 012602      MOV      (SP)+,R2
2550 020244 012603      MOV      (SP)+,R3
2551 020246 012604      MOV      (SP)+,R4
2552 020250 012605      MOV      (SP)+,R5
2553 020252 000207      RTS       PC

```


N7

GETPAT - GET 8 BIT PATTERN FROM OPERATOR

```

2555 .SBTTL GETPAT - GET 8 BIT PATTERN FROM OPERATOR
2556 ;*
2557 ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
2558 ;
2559 ;INPUTS: NONE.
2560 ;
2561 ;OUTPUTS:
2562 ; RO OCTAL NUMBER FROM THE OPERATOR
2563 ;
2564 ;CALLING SEQUENCE:
2565 ; JSR PC,GETPAT
2566 ;-
2567 GETPAT::
2568 ; SAVREG ;SAVE THE GENERAL REGISTERS
2569 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$HILIM
      10000$: BNCOMPLETE 1$ ;RETRY IF ERROR
      BCC 1$
      MOV PATDAT,RO ;DATA PATTERN FROM OPERATOR
      RTS PC ;RETURN TO CALLER

2570 ;*
2571 ;LOCAL DATA AREA
2572 ;-
2573 ;
2574 ;
2575 ;
2576 ;
2577 ;
2578 PATDAT: .WORD 0 ;TEMPORARY STORAGE FOR DATA
2579 DATASC: .ASCIZ 'ENTER DATA PATTERN'
2580 .EVEN

```

GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE

```

2582 .SBTTL GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE
2583 ;
2584 ;ROUTINE TO ISSUE A MENU AND GET THE OPERATOR'S RESPONSE.
2585 ;
2586 ;INPUTS:
2587 ;      R0      ADDRESS OF ASCIZ STRING OF MENU
2588 ;      R1      MAXIMUM ALLOWABLE OPERATOR RESPONSE
2589 ;
2590 ;OUTPUTS:
2591 ;      R0      NUMBER OF THE OPERATOR'S SELECTION
2592 ;
2593 GETSEL::
2594     SAVREG          ;SAVE GENERAL REGISTERS
2595     MOV R0,R2       ;SAVE THE MENU ADDRESS
2596     MOV R2,R3       ;START OF MENU STRING
2597     TST (R3)        ;END OF ASCII ?
2598     BEQ 34          ;BRANCH IF ALL LINES DISPLAYED
2599     PRINTF #SELASC,(R3)+
2600     MOV (R3)+,-(SP) ;DISPLAY THE MENU
2601     MOV #SELASC,-(SP)
2602     MOV #2,-(SP)
2603     MOV SP,R0
2604     TRAP C#PNTF
2605     ADD #6,SP
2606     BR 24
2607     34: GMANID MENASC,MENRES,D,-1,0,-1,NO
2608     TRAP C#GMAN
2609     BR 100014
2610     .WORD MENRES
2611     .WORD T#CODE
2612     .WORD MENASC
2613     .WORD -1
2614     .WORD T#LOLIM
2615     .WORD T#HILIM
2616     100014: BNCOMPLETE 14 ;RETRY IF ERROR
2617     BCC 14
2618     MOV MENRES,R0 ;GET THE OPERATOR'S REPLY
2619     CMP R0,R1 ;COMPARE TO MAXIMUM ALLOWED
2620     BLOS 54 ;BRANCH IF OK
2621     PRINTF #MENERR ;DISPLAY ERROR MESSAGE
2622     MOV #MENERR,-(SP)
2623     MOV #1,-(SP)
2624     MOV SP,R0
2625     TRAP C#PNTF
2626     ADD #4,SP
2627     BR 14 ;RETRY
2628     54: RTS PC ;RETURN TO CALLER
2629     MENERR: .ASCIZ 'N#A *** Menu Selection Too Large ***'
2630     SELASC: .ASCIZ 'N#T'
2631     MENASC: .ASCIZ 'Enter Menu Selection: '
2632     .EVEN
2633     MENRES: .WORD 0

```


C8

CHKMAN - CHECK MANUAL INTERVENTION LEGALITY

```

2615          .SBTTL  CHKMAN - CHECK MANUAL INTERVENTION LEGALITY
2616          ;*
2617          ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
2618          ;
2619          ;INPUT:
2620          ;
2621          ;      NONE.
2622          ;
2623          ;OUTPUT:
2624          ;
2625          ;      CARRY    0      MANUAL INTERVENTION NOT ALLOWED
2626          ;              1      MANUAL INTERVENTION IS OK
2627          ;
2628          ;SIDE EFFECTS:
2629          ;
2630          ;      A MESSAGE IS DISPLAYED WARNING THAT TEST IS
2631          ;      NOT EXECUTED IF MANUAL INTERVENTION IS NOT
2632          ;      ALLOWED.
2633          ;
2634          ;-
2635
2636          CHKMAN::
2637          SAVREG          ;SAVE THE REGISTERS
2638          MANUAL          ;SEE IF MANUAL INTERVENTION OK
2639          TRAP    C#MANI
2640          BCOMPLETE 1#    ;BRANCH IF ALLOWED
2641          BCS      1#
2642          PRINTF    #NOMAN    ;PRINT THE WARNING MESSAGE
2643          MOV      #NOMAN, -(SP)
2644          MOV      #1, -(SP)
2645          MOV      SP, R0
2646          TRAP    C#PNTF
2647          ADD      #4, SP
2648          CLC
2649          RTS      PC      ;CLEAR CARRY FOR ERROR
2650          ;RETURN
2651
2652          1#::
2653          NOMAN: .ASCIZ  'N/A *** Manual Intervention not Allowed - Test Aborted ***'
2654          .even

```

D8

ENVIRN - SETUP FREE DIAGNOSTIC SPACE

```

2648          .SBTTL  ENVIRN  - SETUP FREE DIAGNOSTIC SPACE
2649          ;
2650          ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
2651          ;
2652          ; ENVIRN: MEMORY  R0
2653          TRAP  C#MEM
2654          MOV  R0,FREE          ; GET 1ST FREE ADDRESS...
2655          ADD  #2,FREE          ; ...AND WORD COUNT.
2656          MOV  (R0),FRESIZ
2657          SUB  #4,FRESIZ
2658          MOV  L#UNIT,R2      ; GET NUMBER OF UNITS
2659          SUB  #7,FRESIZ      ; TAKE AWAY 7 WORDS PER UNIT
2660          DEC  R2
2661          BNE  10$
2662          MOV  FREE,R0          ; GET FIRST FREE ADDRESS
2663          ADD  FRESIZ,R0        ; POINT TO LAST FREE ADDRESS
2664          SUB  #2,R0           ; BACKUP 1 WORD
2665          MOV  R0,FREEHI       ; STORE LAST FREE ADDRESS
2666          NOP
2667          MOV  #BDVPCR,R1      ; GET BDV11 PCR ADDRESS
2668          MOV  R1,R2           ; COPY TO R2
2669          ADD  #2,R2           ; SET THE RANGE
2670          JSR  PC,XNXM         ; SEE IF WE HAVE ONE
2671          BCC  15$             ; OK TO SET FLAGS
2672          BR   40$             ; RETURN WITH FLAGS CLEAR
2673          MOV  BDVPCR,R1        ; SAVE PCR CONTENTS
2674          ADD  #1,R1           ; ADD ONE TO IT
2675          MOV  #BDVPCR,R2      ; GET BDV11 PCR ADDRESS
2676          INC  (R2)            ; TRY TO WRITE TO IT
2677          MOV  BDVPCR,R3      ; GET RESULTS
2678          CMP  R1,R3           ; DID IT CHANGE?
2679          BNE  20$             ; NO, MUST BE 11/238
2680          INC  T23A            ; SET THE FLAG
2681          BIC  #170000,L#HIME  ; SUPERVISOR COULD BE WRONG
2682          NOP
2683          PRINTF #M8186        ; BR 40$ FOR RELEASE
2684          BR   40$             ; TELL THE SYSTEM TYPE
2685          INC  T23B            ; RETURN
2686          NOP
2687          PRINTF #M8189        ; SET THE FLAG
2688          BR   40$             ; BR 40$ FOR RELEASE
2689          TELL THE SYSTEM TYPE
2690          RTS  PC              ; RETURN

```


E8

TSV3 - GLOBAL AREAS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 78

SEQ 0095

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

2690                                     .SBTTL  KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS
2691                                     ;*
2692                                     ;ROUTINE TO INIT KT-11
2693                                     ;
2694                                     ;-
2695
2696
2697 021070      KTINIT:
2698 021070      005037  003132      CLR      KTFLG      ; INIT >28K MEMORY FLAG
2699 021074      005037  003134      CLR      KTENABLE   ; INIT TEST >28K FLAG
2700 021100      023727  002120  001577      CMP      L#HIME,#1577 ; GOT ENOUGH MEMORY (>28K)?
2701 021106      101444      BLOS      9#              ; NO.
2702 021110      013700      000004      MOV      @#ERRVEC,R0 ; SAVE OLD ERR VEC PTR.
2703 021114      012737  021206  000004      MOV      @2#,@#ERRVEC ; SET ERR VEC PTR.
2704 021122      005737  177572      TST      @#SRO      ; GOT KT11?
2705 021126      000240      NOP                      ; (TRAP IF NO).
2706 021130      013737  002120  003132      MOV      L#HIME,KTFLG ; YES. SET KT FLAG.
2707 021136      042737  000177  003132      BIC      @177,KTFLG ;
2708 021144      010037  000004      MOV      R0,@#ERRVEC ; RESTORE OLD ERR VEC PTR.
2709 021150      005000      CLR      R0              ; R0 = AR DATA.
2710 021152      012701  172340      MOV      #KIPAR0,R1 ; R1 = KI REGS PTR.
2711 021156      012761  077406  177740  1# :      MOV      @77406,-40(R1) ; SET DESCRIPTOR REG.
2712 021164      010021      MOV      R0,(R1) ; SET KIPAR REG.
2713 021166      062700  000200      ADD      @200,R0 ; BUMP AR DATA BY "4K".
2714 021172      020027  002000      CMP      R0,@2000 ; AT "I/O"?
2715 021176      001367      BNE      1#              ; NO.
2716 021200      012741  177600      MOV      @177600,-(R1) ; YES. SET KTPAR7 FOR I/O.
2717 021204      000405      BR       9#              ;
2718
2719 021206      012716  021214      2# :      MOV      @6#,(SP) ; SET UP RETURN
2720 021212      000002      RTI                      ; RTI TO NEXT LOCATION
2721
2722 021214      010037  000004      6# :      MOV      R0,@#ERRVEC ; RESTORE OLD ERR VEC PTR.
2723
2724 021220      000207      9# :      RTS      PC

```

F8

SEQ 0096

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739 021222
2740
2741 021222 005737 002224
2742 021226 001020
2743 021230 012737 100206 021274
2744 021236 012737 021304 021276
2745 021244 012737 000006 021302
2746 021252 012737 100010 021304
2747 021260 012704 021274
2748 021264 004737 010662
2749 021270 000207
2750
2751
2752
2753 021274
2754
2755 021274 000000
2756 021276 000000
2757 021300 000000
2758 021302 000000
2759
2760
2761
2762 021304 000000
2763 021306 000000
2764 021310 000000
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776 021312
2777
2778 021312
2779 021316 005037 003136
2780 021322 005037 003140
2781 021326 005037 003142
2782 021332 005737 003146

;
; SUBROUTINE TO SET EXTENDED FEATURES SWITCH
;
; Requires that SOFINIT and WRTCHR have been done previous to call.
;
; INPUTS:
; R5 CURRENT UNIT NUMBER
; OUTPUTS:
; The Extended Features Switch is set.
;
;
INVERT::
TST EXTFEA ; IS SWITCH SET?
BNE 1$ ; YES, EXIT STAGE RIGHT! (or the next one outa town!)
MOV #100206, CNDPKT ; WRT SUB-SYS MEM CMD
MOV #WSMBK, CNDPKT+2 ; MSG BUF ADDR
MOV #6, CNDPKT+6 ; BYTE COUNT
MOV #100010, WSMBK ; INVERT THE SWITCH
MOV #CNDPKT, R4 ; SET CNDPKT INTO R4
JSR PC, WRTCHR ; DO IT
RTS PC ; RETURN

;
; COMMAND PACKET.
;
; = <..3>E177774 ; MUST BE ON MOD 4 BOUNDARY.
;
CNDPKT:: 0 ; 1ST WORD IS TS05 COMMAND.
0 ; 2ND WORD IS THE BUFFER LOW ADDRESS.
0 ; 3RD WORD IS THE BUFFER HIGH ADDRESS.
0 ; 4TH WORD IS THE BYTE/RECORD/FILE COUNT.
;
; WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
;
WSMBK:: 0 ; 1ST WORD:: SEL 0
0 ; 2ND WORD:: SEL 2
0 ; 3RD WORD:: SEL 4
.EVEN
;
; SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
;
; INPUTS:
; OUTPUTS:
; The NXMFLG is set if we can test.
; The NXMLO and NXMHI addresses are setup.
;
;
MEMCK::
SAVREG ; SAVE THE REGISTERS
CLR NXMFLG ; CLEAR THE FLAG
CLR NXMLO ; CLEAR THE TEST ADDRESS LO
CLR NXMHI ; CLEAR THE TEST ADDRESS HI
TST T238 ; IS IT A 11/238?

```


KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

2783 021336 001407      BEQ      1#      ;NO
2784 021340 023727 002120 007777    CMP      L#HIME,#7777    ; GREATER THAN 128K
2785 021346 103406      BLO      2#      ; NO
2786 021350 004737 021466      JSR      PC,NXMTST    ;SETUP THE ADDRESS
2787 021354 000427      BR       13#     ;SET THE FLAG AND EXIT
2788 021356 005737 003144      TST      T23A    ;IS IT A 11/23A?
2789 021362 001413      BEQ      4#      ;NO
2790 021364 023727 002120 005777 2# :    CMP      L#HIME,#5777    ;GREATER THAN 96K
2791 021372 101023      BHI      14#     ;YES, 23A/23B WITH 128K MEMORY
2792 021374 023727 002120 003777      CMP      L#HIME,#3777    ;GREATER THAN 64K BUT LESS THAN 92K?
2793 021402 103403      BLO      4#      ;NO, CHECK 24K
2794 021404 004737 021466      JSR      PC,NXMTST    ;SETUP THE ADDRESS
2795 021410 000411      BR       13#     ;SET THE FLAG AND EXIT
2796 021412 023727 002120 001577 4# :    CMP      L#HIME,#1577    ;GREATER THAN 24K BUT LESS THAN 64K?
2797 021420 103410      BLO      14#     ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
2798 021422 004737 021466      JSR      PC,NXMTST    ;SETUP THE ADDRESS
2799 021426 062737 000077 003142      ADD      #77,NXMHI    ;FOOL THE 11/02 & 11/03
2800 021434 005237 003136      INC      NXMFLG    ;SET THE FLAG
2801 021440 000411      BR       15#     ;EXIT
2802 021442 000410      BR       14#     ;NOP FOR PRINTOUT
2803 021444      PRINTF    #NOMEM    ;TELL THEM & EXIT ***NO PRINT*****
      021444 012746 005460      MOV      #NOMEM,-(SP)
      021450 012746 000001      MOV      #1,-(SP)
      021454 010600      MOV      SP,R0
      021456 104417      TRAP     C#PNTF
      021460 062706 000004      ADD      #4,SP
2804 021464 000207      15# :    RTS      PC      ;RETURN
2805
2806      ;*
2807      ; SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
2808      ;
2809      ; OUTPUTS: NXMLO, NXMHI      ; SETUP WITH NXM ADDRESS
2810      ;
2811      ;-
2812
2813 021466 013701 002120      NXMTST: MOV      L#HIME,R1    ;GET TOP OF MEMORY
2814 021472 062701 000200      ADD      #200,R1    ;MAKE IT I/O BLOCK OR OTHER NXM
2815 021476 042701 000177      BIC      #177,R1
2816 021502 010102      MOV      R1,R2    ;RESAVE RESULTS
2817      000006      .REPT      6
2818      .ASL      R1    ;PUT IN PLACE FOR XFER
2819      .ENDR
2820 021520 010137 003140      MOV      R1,NXMLO    ;SAVE TEST ADDRESS LOW
2821      000012      .REPT      10
2822      .ASR      R2    ;PUT IN PLACE FOR XFER
2823      .ENDR
2824 021550 042702 177700      BIC      #177700,R2    ;DON'T WANT ILA!
2825 021554 010237 003142      MOV      R2,NXMHI    ;SAVE TEST ADDRESS HIGH
2826 021560 000207      RTS      PC      ;RETURN
2827
2828 021562      ENDMOD

```

H8

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```

7
8
9 021562      .TITLE  TSV4 - MISCELLANEOUS SECTIONS
  021562      BGNMOD  TSV4
10
11      TSV4::
12
13
14
15
16
17
18
19      .SBTTL  PROTECTION TABLE
20 021562      BGNPROT
  021562      L4PROT::
21 021562      .WORD  -1, -1, -1, -1      ;NO DEVICE PROTECTION REQUIRED.
22 021572      177777 177777 177777      ENDPROT
```


INITIALIZE SECTION

```

24                                     .SBTTL  INITIALIZE SECTION
25
26                                     ;**
27                                     ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
28                                     ;AT THE BEGINNING OF EACH PASS.
29
30                                     ;
31                                     ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
32                                     ;IF "CONTINUE", NOTHING IS REQUIRED.
33                                     ;
34                                     ;--
35                                     ;*
36                                     ;INSERT TEMPORARY JUMP TO ODT
37                                     ;-
38                                     BGNINIT
39                                     L$INIT::
40                                     40$: CLR      EXTFEA
41                                     CLR      NXMFLG
42                                     MOV      @EPT1,EPTSW
43                                     CLR      SIFLAG
44                                     CLR      KTENABLE
45                                     CLR      RAMSIZ
46                                     READDEF @EF.CONTINUE
47                                     MOV      @EF.CONTINUE,R0
48                                     TRAP     C$REFG
49                                     BNCOMPLETE 1$
50                                     BCC      1$
51                                     CMP      UNITN,L$UNIT
52                                     BHIS     4$
53                                     TST      DUFLG
54                                     BMI      NXTU
55                                     MOV      UNITN,R1
56                                     ASL      R1
57                                     TST      ERTABL(R1)
58                                     BEQ      SETU
59                                     BIT      @BIT14,ERTABL(R1)
60                                     BNE      NXTU
61                                     EXIT     INIT
62                                     TRAP     C$EXIT
63                                     .WORD    L10030-
64                                     1$: READDEF @EF.NEW
65                                     MOV      @EF.NEW,R0
66                                     TRAP     C$REFG
67                                     BNCOMPLETE NXTU
68                                     BCC      NXTU
69                                     READDEF @EF.START
70                                     MOV      @EF.START,R0
71                                     TRAP     C$REFG
72                                     BCOMPLETE 2$
73                                     BCS      2$
74                                     READDEF @EF.RESTART
75                                     MOV      @EF.RESTART,R0
76                                     TRAP     C$REFG
77                                     BNCOMPLETE 31$
78                                     BCC      31$
79                                     2$: BRESET
80                                     TRAP     C$RESET
81
82                                     ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
83                                     ;CLEAR "SOFT INIT" FLAG
84                                     ;CLEAR TEST ABOVE 28K FLAG
85                                     ;CLEAR RAM SIZE FOR RAMERR ROUTINE
86
87                                     ;UNIT IN RANGE?
88                                     ;BR IF NO.
89                                     ;DROPPED UNIT?
90                                     ;BR IF YES
91
92                                     ;DROPPED?
93                                     ;DO NOTHING IF "CONTINUE".
94
95                                     ;TAKE NEXT UNIT IF NOT NEW PASS.
96
97                                     ;1ST PASS, BUS-INIT...
98                                     ;BUS RESET.

```

INITIALIZE SECTION

65	021734	005037	002212		CLR	TSTCNT		;NUMBER OF TESTS RUN IN PASS
66	021740	005037	002220		CLR	FATFLG		;CLEAR FATAL ERROR COUNT
67	021744	005037	003144		CLR	T23A		;CLEAR 11/23A FLAG
68	021750	005037	003146		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	021754	005037	003400		CLR	SKIPT		;CLEAR THE SUBTEST "SKIPPER"
73	021760			20:				
74	021760	012737	177777	002202	MOV	#-1,QVP		;...QUICK VERIFY...
75	021766	004737	020710		JSR	PC,ENVIRN		;SET ENVIRONMENT.
76	021772	004737	021070		JSR	PC,KTINIT		;INITIALIZE KT MEMORY MANAGEMENT
77	021776	012700	003176		MOV	#ERTABL,RO		
78	022002	005020		30:	CLR	(RO)+		;CLEAR THE ERROR TABLE
79	022004	020027	003376		CMP	RO,#ERTABE		
80	022010	103774			BLO	30		
81	022012	000404			BR	4		
82	022014	005037	002202	31:	CLR	QVP		
83	022020	000137	022070		JMP	PASRPT		;GO REPORT THE STATUS
84								
85	022024			4:				
86	022024	012737	177777	002200	NEWPAS:	MOV	#-1,UNITN	;INIT UNIT NUMBER...
87	022032	005037	002216		CLR	DEVCNT		;CLEAR COUNT OF DEVICES RUNNING
88	022036			NXTU:	BREAK			
89	022040	005237	002200		TRAP	C#BRK		
90	022044	023737	002200	002012	INC	UNITN		;...AND SET NEXT UNIT NUMBER.
91	022052	103423			CMP	UNITN,L#UNIT		
92	022054	012737	177777	003112	SETU			
93	022062	000401			MOV	#-1,DUFLG		
94	022064				BR	11		
95	022066	000240		11:	DOCLN			;ABORT, NO MORE UNITS.
96	022070			PASRPT:	TRAP	C#DCLN		
97	022070	023727	002012	000001	NOP			
98	022076	101752			CMP	L#UNIT,#1		;HOW MANY UNITS SELECTED?
99	022100	005737	002216		BLOS	NEWPAS		;BR IF ONLY 1
100	022104	001747			TST	DEVCNT		;ARE ANY STILL RUNNING?
101	022106				BEQ	NEWPAS		;BR IF NO
102	022110	104421			RFLAGS	RO		
103	022114	001343	000100		TRAP	C#RFLA		
104					BIT	#ISR,RO		;SHOULD WE PRINT STATISTICS
105	022116				BNE	NEWPAS		;BR IF NO
106	022120	104424			DORPT			
107	022122	000741		10:	TRAP	C#DRPT		
108					BR	NEWPAS		
109	022122			SETU:				;GET UNIT N P-TABLE POINTER.
110	022130	013700	002200		MOV	UNITN,RO		
111	022132	005037	003112		TRAP	C#GPHRD		
112	022136	005237	002216		BNCOMPLETE	NXTU		;BR IF UNIT NOT AVAILABLE.
113	022142	012001			BCC	NXTU		
114	022144	010137	002204		CLR	DUFLG		;CLEAR "DROPPED" FLAG.
					INC	DEVCNT		
					MOV	(RO)+,R1		;GET 1ST REGISTER ADDRESS.
					MOV	R1,CSRADDR		;ADDRESS OF REGISTERS OF UNIT UNDER TEST

K8

SEQ 0101

INITIALIZE SECTION

```

115
116 022150 012001      MOV      (R0),R1      ;GET VECTOR ADDRESS.
117                   ;MOV      (R0),R2      ;GET INTERRUPT PRIORITY
118                   ;MOV      R2,IPRI      ;SET INTERRUPT PRIORITY.
119 022152 010137 002206 MOV      R1,IVEC      ;SET INTERRUPT VECTOR POINTER...
120 022156 012721 016276 MOV      #INTR,(R1)+    ;...VECTOR...
121 022162 013721 002210 MOV      IPRI,(R1)+    ;...AND PRIORITY.
122
123 022166             14:      TST      QVP      ;1ST PASS ??
124                   ;          BEQ      54      ;NO, SKIP THE PASS 1 STUFF.
125
126
127                   ;
128                   ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
129                   ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
130                   ;
131 022166 013701 002200      MOV      UNITN,R1
132 022172 006301           ASL      R1
133 022174 052761 100000 003176 BIS      #BIT15,ERTABL(R1)      ;SAY DEVICE RUNNING
134 022202 005037 005772     CLR      EXTA      ;CLEAR ERROR EXTENSION FLAG.
135 022206 023727 002012 000001 CMP      L#UNIT,#1      ;ARE WE TESTING MULTIPLE UNITS?
136 022214 101416           BLOS     104      ;BR IF NO.
137 022216           RFLAGS     R0      ;YES -- GET OPERATOR FLAGS.
138 022220 032700 001000     TRAP     C#RFLA
139 022224 001412           BIT      #PNT,R0      ;SHOULD WE PRINT UNIT #?
140 022226           BEQ      104      ;BR IF NOT.
141 022226 013746 002200     PRINTF   #PUNIT,UNITN      ;PRINT THE UNIT #
142 022232 012746 022320     MOV      UNITN,-(SP)
143 022236 012746 000002     MOV      #PUNIT,-(SP)
144 022242 010600           MOV      #2,-(SP)
145 022244 104417           MOV      SP,R0
146 022246 062706 000006     TRAP     C#PNTF
147 022252           ADD      #6,SP
148 022252 005037 003114     104:    CLR      NODEV
149 022256 013701 002204     MOV      CSRADDR,R1      ;ADDRESS OF FIRST REGISTER
150 022262 010102           MOV      R1,R2      ;START OF REGISTERS
151 022264 062702 000002     ADD      #TSSR,R2      ;ADDRESS OF TSSR REGISTER
152 022270 004737 016456     JSR      PC,XNXM      ;TEST BOTH CONTROLLER REGISTERS...
153 022274 103005           BCC      24      ;...AND BR IF ALL OK.
154 022276 010137 003114     MOV      R1,NODEV      ;FLAG DEVICE AS NON-EXISTENT
155 022302 012737 177777 003112     MOV      #-1,DUFLG      ;DROP THIS UNIT.
156 022310
157
158                   ;
159                   ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
160                   ;
161 022310 012700 000000     54:      SETPRI   #PRI00      ;ENABLE INTERRUPTS.
162 022314 104441           MOV      #PRI00,R0
163 022316           TRAP     C#SPRI
164 022316           ENDINIT
165 022316 104411           L10030:    TRAP     C#INIT
166 022320 045 116 045 PUNIT: .ASCIZ  /#N#N#A***** TESTING UNIT #D2#A *****/
167 .EVEN

```


M8

ADD AND DROP UNITS SECTIONS

LINE	ADDRESS	OPERAND 1	OPERAND 2	OPERAND 3	INSTR	COMMENT
197	022540	045	116	045	14:	.ASCIZ /%N% UNIT %D% A DROPPED/
198						.EVEN
199	022570					ENDDU
	022570				L10032:	
	022570	104453				TRAP C4DU
200					;	*
201					;	AUTO-DROP CODE SECTION.
202					;	--
203	022572					BGNAUTO
	022572				L4AUTO::	
204	022572	013705	002204		MOV	CSRADDR,R5 ;POINT TO DEVICE REGISTER
205	022576	012703	000550		MOV	#360.,R3 ;ENOUGH TIME FOR 2400' REEL TO REWIND
206	022602	004737	016330	104:	JSR	PC, WAITF ;WAIT FOR SSR TO SET
207	022606	103420			BCS	204 ;LEAVE WHEN SSR IS SET
208	022610				DELAY	250. ;WAIT FOR .25 SECONDS
	022610	012727	000372		MOV	#250.,(PC)+
	022614	000000			.WORD	0
	022616	013727	002116		MOV	L4DLY,(PC)+
	022622	000000			.WORD	0
	022624	005367	177772		DEC	-6(PC)
	022630	001375			BNE	.-4
	022632	005367	177756		DEC	-22(PC)
	022636	001367			BNE	.-20
209	022640	005303			DEC	R3 ;BUMP COUNTER DOWN
210	022642	001357			BNE	104 ;KEEP GOING
211	022644	004737	017262		JSR	PC,CKDROP ;TRY AND DROP UNIT
212	022650			204:		
213	022650				ENDAUTO	; UNUSED.
	022650			L10033:		
	022650	104461			TRAP	C4AUTO

CLEAN-UP AND REPORT CODING SECTIONS

```

023016 012746 000002      MOV      #2,-(SP)
023022 010600      MOV      SP,R0
023024 104416      TRAP     C#PNTS
023026 062706 000006      ADD      #6,SP
254 023032 000431      BR      44
255 023034 020227 160001    34:      CMP      R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
256 023040 001012      BNE     304      ; BR IF NO.
257 023042      PRINTS    #DEVNRD,R3
023042 010346      MOV      R3,-(SP)
023044 012746 023331      MOV      #DEVNRD,-(SP)
023050 012746 000002      MOV      #2,-(SP)
023054 010600      MOV      SP,R0
023056 104416      TRAP     C#PNTS
023060 062706 000006      ADD      #6,SP
258 023064 000414      BR      44
259 023066 042702 170000    304:     BIC      #1C7777,R2
260 023072      PRINTS    #DEVDR0,R3,R2
023072 010246      MOV      R2,-(SP)
023074 010346      MOV      R3,-(SP)
023076 012746 023412      MOV      #DEVDR0,-(SP)
023102 012746 000003      MOV      #3,-(SP)
023106 010600      MOV      SP,R0
023110 104416      TRAP     C#PNTS
023112 062706 000010      ADD      #10,SP
261 023116 062704 000002    44:      ADD      #2,R4
262 023122 005203      INC      R3
263 023124 020427 003376    CMP      R4,#ERTABE
264 023130 103701      BLO     14
265 023132 012604      MOV      (SP)+,R4
266 023134 012603      MOV      (SP)+,R3
267 023136 012602      MOV      (SP)+,R2
268 023140      ENDRPT      ; UNUSED.
023140      L10035:
023140 104425      TRAP     C#RPT
269
270 023142      045      116      045  DEVSUM: .ASCIZ  /#N#ADEVICE STATUS SUMMARY:#N/
271 023177      045      101      040  DEVONL: .ASCIZ  /#A UNIT #D3#A ONLINE, ERRORS = #D#N/
272 023247      045      101      040  DEVNXR: .ASCIZ  /#A UNIT #D3#A DROPPED, NON-EXISTENT REGISTER#N/
273 023331      045      101      040  DEVNRD: .ASCIZ  /#A UNIT #D3#A DROPPED, NOT READY AT STARTUP#N/
274 023412      045      101      040  DEVDR0: .ASCIZ  /#A UNIT #D3#A DROPPED, ERRORS = #D#N/
275      .EVEN
276
277 023462      ENDMOD
278

```

C9

CLEAN-UP AND REPORT CODING SECTIONS

```

1      .TITLE  TSVSA - HARDWARE TESTS
2
9
10 023462      BGNMOD  TSVS
    023462      TSV5::

16
24
25      .SBTTL  TEST  1: BUS RESET TEST
26
27
28      :
29      : THIS TEST VERIFIES THAT THE M7196 MODULE'S DEVICE REGISTERS ARE
30      : ACCESSIBLE ON THE BUS (SUBTEST 1) AND THEN CHECKS THAT THE
31      : BUILT-IN INITIALIZATION SELF-TEST MICRODIAGNOSTIC DID NOT FIND
32      : ANY BASIC PROBLEMS IN THE MODULE. AREAS OF LOGIC TESTED BY THE
33      : SELF-TEST SEQUENCE ARE AS FOLLOWS: ROM AND PIPELINE REGISTER,
34      : SEQUENCER, INTERNAL BUSES, 2901 MICROPROCESSOR, AND, RAM. THIS
35      : TEST INITIALIZES THE CONTROLLER BY ISSUING THE BUS INIT SIGNAL
36      : VIA A RESET INSTRUCTION, OR BY WRITING INTO THE TSSR REGISTER,
37      : WAITS A PERIOD OF TIME (TO ALLOW THE CONTROLLER'S INITIALIZATION
38      : MICRODIAGNOSTIC SEQUENCE TO BE COMPLETED), AND THEN CHECKS THE
39      : CONTENTS OF THE TSSR REGISTER. SUCCESSFUL INITIALIZATION IS
40      : INDICATED BY SUBSYSTEM READY (SSR) AND NEED BUFFER ADDRESS (NBA)
41      : BITS BEING SET (1) AND ALL OTHER BITS (EXCEPT A17 AND A16 AND
42      : OFL, WHICH ARE IGNORED FOR THIS TEST) BEING CLEAR (0). IF THE
43      : CONTENTS OF TSSR ARE NOT AS EXPECTED, AN ERROR REPORT IS ISSUED
44      : LISTING THE EXPECTED DATA, ACTUAL DATA, AND THE DISCREPANCIES.
45      : THE ERROR REPORT ANALYZES THE TSSR CONTENTS AND DISCERNs AND
46      : REPORTS ONE OF THREE POSSIBILITIES:
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      : 2. SSR = 0, SC = 0 AND THE ERROR CODE IN BITS 0-5 IS IN
60      : THE RANGE 17-13: THIS IS A FATAL ERROR. THE ERROR
61      : CODE IS DECODED AND THE APPROPRIATE DESCRIPTION GIVEN.
62      : INDICATES THAT A SERIOUS PROBLEM EXISTS.
63
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    :

```


TEST 1: BUS RESET TEST

76							
77	023504			BRESET			;ISSUE A BUS RESET
	023504	104433					TRAP C4RESET
78	023506	004737	016330	JSR	PC, WAITF		;WAIT FOR READY
79	023512	016501	000002	MOV	TSSR(R5), R1		;GET THE CONTENTS OF TSSR
80	023516	010102		MOV	R1, R2		;CONTENTS OF TSSR
81	023520	042702	176277	BIC	#1C<HIADDR!OFL>, R2		;THESE BITS MAY BE SET
82	023524	052702	002200	BIS	#SSR!NBA, R2		;READY AND NEW DATA SHOULD BE SET
83	023530	020102		CMF	R1, R2		;COMPARE EXPECTED TO RECEIVED
84	023532	001405		BEQ	104		;BRANCH IF COMPARE
88	023534			ERRDF	ERRNO, SFHERR, SFFMSG		;REPORT A FATAL ERROR
	023534	104455					TRAP C4ERDF
	023536	000145					.WORD 101
	023540	003705					.WORD SFHERR
	023542	012152					.WORD SFFMSG
89	023544	005203		INC	R3		;SET THE FATAL ERROR FLAG
90	023546						
91	023546			ENDSUB			;////////// END SUBTEST //////////
	023546						L10037: TRAP C4ESUB
92							
93	023550	005703		TST	R3		;DID WE HAVE FATAL ERROR ?
94	023552	001402		BEQ	204		;BRANCH IF NOT
95	023554	004737	017262	JSR	PC, CKDROP		;GO DROP THIS UNIT, IF ALLOWED
96	023560	005003		CLR	R3		;RESET FATAL ERROR FLAG
97							
98							
99	023562			BGNSUB			;////////// BEGIN SUBTEST //////////
	023562						T1.2: TRAP C4BSUB
	023562	104402					
100							
101	023564	005065	000002	CLR	TSSR(R5)		;WRITE TO ISSUE A SOFT RESET
102	023570	004737	016330	JSR	PC, WAITF		;WAIT FOR READY TO SET
103	023574	016501	000002	MOV	TSSR(R5), R1		;GET REGISTER TSSR DATA
104	023600	010102		MOV	R1, R2		;CONTENTS OF TSSR
105	023602	042702	176277	BIC	#1C<HIADDR!OFL>, R2		;THESE BITS MAY BE SET
106	023606	052702	002200	BIS	#SSR!NBA, R2		;READY AND NEW DATA SHOULD BE SET
107	023612	020102		CMF	R1, R2		;COMPARE EXPECTED TO RECEIVED
108	023614	001405		BEQ	104		;BRANCH IF COMPARE
112	023616			ERRDF	ERRNO, SFIERR, SFFMSG		;REPORT A FATAL ERROR
	023616	104455					TRAP C4ERDF
	023620	000146					.WORD 102
	023622	003652					.WORD SFIERR
	023624	012152					.WORD SFFMSG
113	023626	005203		INC	R3		;SET THE ERROR FLAG
114	023630						
115	023630			ENDSUB			;////////// END SUBTEST //////////
	023630						L10040: TRAP C4ESUB
	023630	104403					
116							
117							
118	023632	005703		TST	R3		;FATAL ERROR DETECTED ?
119	023634	001402		BEQ	204		;BRANCH IF NOT
120	023636	004737	017262	JSR	PC, CKDROP		;SEE IF TIME TO DROP UNIT
121	023642	004737	016536	JSR	PC, TSTLOOP		;SHOULD WE DO ITERATIONS ?
122	023646	103002		BCC	404		;BRANCH IF NOT
123	023650	000137	023500	JMP	T1LOOP		;LOOP UNTIL COUNT EXPIRED

TSVSA - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-2

SEQ 0108

TEST 1: BUS RESET TEST

```

124 023654 104432 404: EXIT TST ;ALL DONE THIS TEST TRAP C#EXIT
023656 000022 .WORD L10036-
125
126 ;*
127 ;LOCAL TEXT MESSAGES FOR TEST
128 ;-
129
130 023660 111 156 151 TST1ID: .ASCIZ 'Initialization'
131 .EVEN
132 023700 .ENDTST
023700
023700 104401 L10036: TRAP C#ETST
133
134 .SBTTL TEST 2: WRAP DATA - HIGH BYTE
135
136
137 ; THIS TEST VERIFIES OPERATION OF:
138 ;
139 ;
140 ; 1. PART OF THE LSI-11 BUS INTERFACE SECTION OF THE M7196
141 ; MODULE: PART OF THE INPUT FILE (TSDB HIGH BYTE), PART
142 ; OF THE OUTPUT FILE (TSSR HIGH BYTE AND TSBA, BOTH
143 ; BYTES), PART OF THE DCO05 TRANSCEIVER CIRCUITS (ADDRESS
144 ; DECODER, BDAL DRIVERS, HIGH BYTE OF INTERNAL DAL BUS
145 ; DRIVERS), AND BASIC PROGRAMMED I/O CONTROL SEQUENCES
146 ; AND LOGIC;
147 ;
148 ; 2. PART OF 2901 MICROPROCESSOR ELEMENTS (Q-REGISTER,
149 ; REGISTER 0, ROTATE AND NEGATE FUNCTIONS;
150 ;
151 ; 3. Y AND SOURCE BUSES;
152 ;
153 ; 4. BASIC MICROPROGRAM SEQUENCES.
154 ;
155 ;
156 ;
157 ; THE PROGRAM WRITES A TEST DATA BYTE INTO THE HIGH BYTE OF TSDB,
158 ; WAITS FOR THE SSR BIT IN TSSR TO SET, THEN CHECKS THE CONTENTS
159 ; OF BOTH TSBA AND TSSR. THE MODULE IS FUNCTIONING CORRECTLY IF
160 ; DATA WRITTEN APPEARS IN BOTH BYTES OF TSBA AND THE FINAL CONTENT
161 ; OF TSSR IS CORRECT (SAME AS AFTER INITIALIZATION EXCEPT FOR BITS
162 ; 8 AND 9, WHICH SHOULD CONTAIN BITS 8 AND 9 OF THE DATA PATTERN
163 ; WRITTEN. AN ERROR IS REPORTED AND A DESCRIPTIVE ANALYSIS GIVEN
164 ; IF A DISCREPANCY IN TSBA OR TSSR IS DETECTED. THE ANALYSIS
165 ; LISTS LIKELY FAULTY CANDIDATES FROM THE LOGIC ELEMENTS LISTED
166 ; ABOVE. THE TEST IS REPEATED FOR ALL COMBINATIONS OF TEST DATA
167 ; BYTES (0-377 OCTAL).
168 ;
169 ; BGNTST
023702
023702
174 023702 012700 024350 MOV #TST2ID,R0 ;ASCII MESSAGE TO IDENTIFY TEST
175 023706 004737 016570 JSR PC,TSTSETUP ;DO INITIAL TEST SETUP
176 023712 012737 000024 002214 MOV #20,LOOPCNT ;PERFORM 20 ITERATIONS
177 023720 005004 T2LOOP: CLR R4 ;STARTING DATA PATTERN
178 023722 012703 177777 MOV #1,R3 ;DO INIT ON FIRST TIME THROUGH
179 023726 005703 54: TST R3 ;DO WE NEED SOFT INIT

```


TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-5

SEQ 0111

TEST 3: WRAP DATA - LOW BYTE

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

```

373
374 .SBTTL TEST 4: RAM TEST
375
376 ; THIS TEST VERIFIES THAT ALL LOCATIONS OF THE RAM ON THE M7196
377 ; CAN PROPERLY STORE AND READ BACK ALL DATA PATTERNS, AND THAT
378 ; EACH RAM LOCATION IS UNIQUELY ADDRESSED (I.E., THAT ONE AND ONLY
379 ; ONE LOCATION IS ACCESSED BY ANY PARTICULAR ADDRESS). THESE
380 ; TESTS ARE PERFORMED BY THREE SUBTESTS, DESCRIBED BELOW. A
381 ; BYPRODUCT OF THESE TESTS IS A VERIFICATION OF TWO REGISTERS IN
382 ; THE 2901 AND THE CAPABILITY OF THE 2901 TO CORRECTLY PERFORM AN
383 ; ADD.
384 ;

```


TEST 4: RAM TEST

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.

```

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 025072      BGNTST
    025072
422
423 025072      BGNSUB
    025072
    025072      104402
424
429 025074 012700 026402      MOV      #TST4ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
430 025100 004737 016570      JSR      PC,TSTSETUP    ;DO INITIAL TEST SETUP
431 025104 012737 000005 002214  MOV      #5,LOOPCNT    ;PERFORM 5 ITERATIONS
432 025112
433 025112 004737 016054      JSR      PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
434 025116 103405      BCS      204,      ;BR IF INIT WAS OK
438 025120 010001      MOV      R0,R1      ;CONTENTS OF TSSR REGISTER
439 025122      ERDIF      ERRNO,SFIERR,SFIMSG    ;FATAL ERROR TSSR WAS NOT OK
    025122      104455
    025124      000621
    025126      003652
    025130      012104
440 025132 005004      204:      CLR      R4      ;SET RAM ADDRESS AT ZERO
441 025134 004737 016416      JSR      PC,CHKTSSR    ;WAIT FOR READY, NON-AMBIGUOUS

```

T4.1: TRAP C#BSUB
 TRAP C#ERDF
 .WORD 401
 .WORD SFIERR
 .WORD SFIMSG

SEQ 0114

TEST 4: RAM TEST

Address	Offset	Hex	Label	Instruction	Comment
442	025140	105065	000000		
443	025144		254:	CLRB	TSDB(R5)
444	025144			BGNSEG	
445	025144	104404			
446	025146	110402		MOVB	R4,R2
447	025150	004737	016416	JSR	PC,CHKTSSR
448	025154	010465	000000	MOV	R4,TSDB(R5)
449	025160	004737	016416	JSR	PC,CHKTSSR
450	025164	110265	000000	MOVB	R2,TSDB(R5)
451	025170	004737	016416	JSR	PC,CHKTSSR
452	025174	016501	000000	MOV	TSBA(R5),R1
453	025200	120102		CMPB	R1,R2
454	025202	001404		BEQ	304
458	025204			ERRHRD	ERRNO,TSBAM2,EXPREC
	025204	104456			
	025206	000622			
	025210	026240			
	025212	015554			
459	025214		304:		
460	025214			ENDSEG	
	025214				
	025214	104405			
461	025216	005204		INC	R4
463	025220	020427	010000	CMP	R4,#10000
464	025224	001347		BNE	254
465	025226			ESCAPE	SUB
	025226	104410			
	025230	000102			
466	025232	005002			
468	025234	005304		CLR	R2
469	025236	110402		DEC	R4
470	025240	004737	016416	MOVB	R4,R2
471	025244	010465	000000	JSR	PC,CHKTSSR
472	025250	004737	016416	MOV	R4,TSDB(R5)
473	025254	016501	000000	JSR	PC,CHKTSSR
474	025260	120102		MOV	TSBA(R5),R1
475	025262	001404		CMPB	R1,R2
479	025264			BEQ	454
	025264	104456		ERRHRD	ERRNO,TSBAM2,EXPREC
	025266	000623			
	025270	026240			
	025272	015554			
480	025274		454:		
	025274	104406		CKLOOP	
481	025276	116501	000001	MOVB	TSBAH(R5),R1
482	025302	010402		MOV	R4,R2
483	025304	000302		SWAB	R2
484	025306	060402		ADD	R4,R2
485	025310	120102		CMPB	R1,R2
486	025312	001404		BEQ	504
490	025314			ERRHRD	ERRNO,M2901,EXPREC
	025314	104456			
	025316	000624			
	025320	026150			

TEST 4: RAM TEST

```

537
538 025452 005304
539 025454 005002
540 025456 004737 016416
541 025462 010465 000000
542 025466 004737 016416
543 025472 016501 000000
544 025476 005002
545 025500 120102
546 025502 001404
550 025504
    025504 104456
    025506 000627
    025510 026322
    025512 015554
551 025514 012702 000377
552 025520 010465 000000
553 025524 004737 016416
554 025530 110265 000000
555 025534 004737 016416
556 025540 016501 000000
557 025544 120102
558 025546 001404
562 025550
    025550 104456
    025552 000630
    025554 026240
    025556 015554
563 025560
    025560 104406
564 025562 004737 016416
565 025566 010465 000000
566 025572 004737 016416
567 025576 116501 000001
568 025602 010403
569 025604 000303
570 025606 060403
571 025610 120103
572 025612 001404
576 025614
    025614 104456
    025616 000631
    025620 026150
    025622 015554
577 025624
    025624 104406
578 025626 005304
579 025630 002312
580
581 025632
    025632
    025632 104403
582
583
584 025634
    025634
    025634 104402

354: DEC R4
    CLR R2
404: JSR PC,CHKTSSR
    MOV R4,TSDB(R5)
    JSR PC,CHKTSSR
    MOV TSBA(R5),R1
    CLR R2
    CMPB R1,R2
    BEQ 434
    ERRHRD ERRNO,TSBAM3,EXPREC

434: MOV #000377,R2
    MOV R4,TSDB(R5)
    JSR PC,CHKTSSR
    MOVB R2,TSDB(R5)
    JSR PC,CHKTSSR
    MOV TSBA(R5),R1
    CMPB R1,R2
    BEQ 454
    ERRHRD ERRNO,TSBAM2,EXPREC

454: CKLOOP
    JSR PC,CHKTSSR
    MOV R4,TSDB(R5)
    JSR PC,CHKTSSR
    MOVB TSBAH(R5),R1
    MOV R4,R3
    SWAB R3
    ADD R4,R3
    CMPB R1,R3
    BEQ 504
    ERRHRD ERRNO,M2901,EXPREC

504: CKLOOP
    DEC R4
    BGE 404
    ENDSUB

BGNSUB

;SET BACK TO 7777
;SET TO ALL ZEROS
;WAIT FOR READY (SSR) TO SET
;LOAD UP THE ADDRESS FOR RAM
;WAIT FOR READY (SSR) TO SET
;READ THE RAM CONTENTS BACK
;LOOKING FOR 000000 (EXPECTED)
;BOTH SHOULD BE 00000000 BINARY
;BR, IF DATA IS GOOD
;CHARACTERISTICS DATA NOT CORRECT
    TRAP C4ERHRD
    .WORD 407
    .WORD TSBAM3
    .WORD EXPREC

;SET ALL ONES WORD
;LOAD UP RAM ADDRESS POINTER
;WAIT FOR READY, NON-AMBIGUOUS
;WRITE DATA INTO RAM
;WAIT FOR READY, NON-AMBIGUOUS
;READ RAM CONTENTS BACK
;CHECK WITH DATA WRITTEN
;BR IF OK, DATA IN = DATA OUT
;WRITTEN DATA NOT = TO READ
    TRAP C4ERHRD
    .WORD 408
    .WORD TSBAM2
    .WORD EXPREC

;SCOPE LOOP
    TRAP C4CLP1
;WAIT FOR READY, NON-AMBIGUOUS
;WORD WRITE TO SET UP ADDRESS
;WAIT FOR READY, NON-AMBIGUOUS
;HIGH BYTE READ OF TSBA
;DATA PATTERN WRITTEN
;HIGH TO LOW
;TOTAL OF BYTES IN LOW BYTE
;SUM OF BYTES WRITTEN TO TSDB = TSBAH
;BR IF OK, THEY SHOULD BE
;2901 PROBLEM ADDER
    TRAP C4ERHRD
    .WORD 409
    .WORD M2901
    .WORD EXPREC

;SCOPE LOOP
    TRAP C4CLP1
;DROP RAM ADDRESS POINTER
;NOT AT LOC. ZERO YET

;////////// END SUBTEST ////////////
    L10045:
    TRAP C4ESUB

;////////// BEGIN SUBTEST ////////////
    T4.3:
    TRAP C4BSUB

```


SEQ 0117

TEST 4: RAM TEST

Address	Hex Address	Hex Data	Hex Data	Assembly	Comments	Labels
585				TEST 4, SUBTEST 3		
586						
587						
588				THIS SUBTEST WRITES RAM WITH ALL ONES		
589				THEN WALKS A ZERO WORD DOWN THROUGH MEMORY		
590						
591	025636	004737	016054	JSR PC,SOFINIT	DO INITIALIZE ON CONTROLLER	
592	025642	103405		BCS 204	BR IF INIT WAS OK	
596	025644	010001		MOV R0,R1	CONTENTS OF TSSR REGISTER	
597	025646			ERRDF ERRNO,SFIERR,SFIMSG	FATAL ERROR TSSR WAS NOT OK	
	025646	104455			TRAP	C4ERDF
	025650	000632			.WORD	410
	025652	003652			.WORD	SFIERR
	025654	012104			.WORD	SFIMSG
598	025656	012702	177777	204: MOV #177777,R2	SET DATA AT ALL ONES	
599	025662	005004		CLR R4	SET RAM ADDRESS AT ZERO	
600	025664	004737	016416	JSR PC,CHKTSSR	WAIT FOR READY, NON-AMBIGUOUS	
601	025670	105065	000000	CLRB TSDB(R5)	SET INTO MAINTENANCE MODE	
602	025674			254: BGNSEG		
603	025674				)))))) BEGIN SEGMENT))))	
	025674	104404			TRAP	C4BSEG
604	025676	004737	016416	JSR PC,CHKTSSR	WAIT FOR READY, NON-AMBIGUOUS	
606	025702	010465	000000	MOV R4,TSDB(R5)	LOAD ADDRESS INTO TSDB	
607	025706	004737	016416	JSR PC,CHKTSSR	WAIT FOR READY, NON-AMBIGUOUS	
608	025712	110265	000000	MOVB R2,TSDB(R5)	LOADS DATA INTO RAM LOCATION	
609	025716	004737	016416	JSR PC,CHKTSSR	WAIT FOR READY, NON-AMBIGUOUS	
610	025722	016501	000000	MOV TSBA(R5),R1	READS WRAP DATA	
611	025726	120102		CPB R1,R2	DOES WRITTEN(WRAP) = READ ?	
612	025730	001404		BEQ 304	BR IF OK, THEY ARE EQUAL	
616	025732			ERRHRD ERRNO,TSBAM2,EXPREC	DATA NOT WRAPPED CORRECTLY	
	025732	104456			TRAP	C4ERHRD
	025734	000633			.WORD	411
	025736	026240			.WORD	TSBAM2
	025740	015554			.WORD	EXPREC
617	025742			304: ENDSEG		
618	025742				)))))) END SEGMENT))))	
	025742	104405			100004: TRAP	C4ESEG
619	025744	005204				
620	025744	005204		INC R4	NEXT ADDRESS	
621	025746	020427	010000	CMP R4,#10000	END OF RAM MEMORY CHECK	
622	025752	001350		BNE 254	BR, MORE RAM TO GO	
623						
624	025754			ESCAPE SUB	NO CHECK IF WRITTEN INCORRECTLY	
	025754	104410			TRAP	C4ESCAPE
	025756	000152			.WORD	L10046--
625	025760	005304		354: DEC R4	SET BACK TO 7777	
626	025762	004737	016416	JSR PC,CHKTSSR	WAIT FOR READY, NON-AMBIGUOUS	
627	025766	012702	000377	MOV #000377,R2	SET UP EXPECTED DATA REGISTER	
628	025772	005001		CLR R1	CLEAN OUT REGISTER	
629	025774	010465	000000	MOV R4,TSDB(R5)	SELECT ADDRESS IN RAM	
630	026000	004737	016416	JSR PC,CHKTSSR	WAIT FOR READY (SSR)	
631	026004	016501	000000	MOV TSBA(R5),R1	PICK UP RAM CONTENTS	
632	026010	120102		CPB R1,R2	IS MEMORY STILL ALL ONES	
633	026012	001404		BEQ 434	BR, IF OK (ALL ONES)	
637	026014			ERRHRD ERRNO,TSBAM3,EXPREC	MEMORY CHANGED AFTER ALL ONES WRITE	

TEST 4: RAM TEST

```

026014 104456
026016 000634
026020 026322
026022 015554
638 026024 005002
639 026026 010465 000000
640 026032 004737 016416
641 026036 110265 000000
642 026042 004737 016416
643 026046 016501 000000
644 026052 120102
645 026054 001404
649 026056
026056 104456
026060 000635
026062 026240
026064 015554
650 026066
026066 104406
651 026070 004737 016416
652 026074 116501 000001
653 026100 010203
654 026102 000303
655 026104 060203
656 026106 120103
657 026110 001404
661 026112
026112 104456
026114 000636
026116 026150
026120 015554
662 026122
026122 104406
663 026124 005304
664 026126 001315
665
666 026130
026130
026130 104403
667
668 026132 004737 016536
669 026136 103002
670 026140 000137 025112
671 026144
026144 104432
026146 000256
672
673
674
675
676 026150 040 124 123 M2901: .ASCIZ ' TSBA High Byte Not Sum of Last TSDB Write (2901 Error)'
677 026240 040 127 162 TSBAM2: .ASCIZ ' Write to TSDB Not Equal to Read of TSBA Low Byte'
678 026322 127 162 151 TSBAM3: .ASCIZ 'Write To RAM Location Modified Another Location'
679 026402 122 101 115 TST4ID: .ASCIZ 'RAM Verification'
680
681 026424
026424

```

```

434: CLR R2
MOV R4,TSDB(R5)
JSR PC,CHKTSSR
MOVB R2,TSDB(R5)
JSR PC,CHKTSSR
MOV TSBA(R5),R1
CMPB R1,R2
BEQ 454
ERRHRD ERRNO,TSBAM2,EXPREC

454: CKLOOP
JSR PC,CHKTSSR
MOVB TSBAH(R5),R1
MOV R2,R3
SWAB R3
ADD R2,R3
CMPB R1,R3
BEQ 504
ERRHRD ERRNO,M2901,EXPREC

504: CKLOOP
DEC R4
BNE 404
ENDSUB

634: JSR PC,TSTLOOP
BCC 634
JMP T4LOOP
EXIT TST

;LOCAL TEXT MESSAGES FOR TEST
;
M2901: .ASCIZ ' TSBA High Byte Not Sum of Last TSDB Write (2901 Error)'
TSBAM2: .ASCIZ ' Write to TSDB Not Equal to Read of TSBA Low Byte'
TSBAM3: .ASCIZ 'Write To RAM Location Modified Another Location'
TST4ID: .ASCIZ 'RAM Verification'

```

```

;SET UP NEW EXPECTED
;LOAD UP RAM ADDRESS POINTER
;WAIT FOR READY, NON-AMBIGUOUS
;WRITE DATA INTO RAM
;WAIT FOR READY, NON-AMBIGUOUS
;READ RAM CONTENTS BACK
;CHECK WITH DATA WRITTEN
;BR IF OK, DATA IN = DATA OUT
;WRITTEN DATA NOT = TO READ
TRAP C4ERHRD
.WORD 412
.WORD TSBAM3
.WORD EXPREC

;SCOPE LOOP
TRAP C4CLP1
;WAIT FOR READY, NON-AMBIGUOUS
;HIGH BYTE READ OF TSBA
;DATA PATTERN WRITTEN
;HIGH TO LOW
;TOTAL OF BYTES IN LOW BYTE
;SUM OF BYTES WRITTEN TO TSDB = TSBAH
;BR IF OK, THEY SHOULD BE
;2901 PROBLEM ADDER
TRAP C4ERHRD
.WORD 414
.WORD M2901
.WORD EXPREC

;SCOPE LOOP
TRAP C4CLP1
;DROP RAM ADDRESS POINTER
;NOT AT LOC. ZERO YET

;///////////////// END SUBTEST ///////////////////
L10046: TRAP C4ESUB

;DO WE NEED TO ITERATE TEST ?
;BRANCH IF NOT
;EXECUTE AGAIN
;ALL DONE THIS TEST
TRAP C4EXIT
.WORD L10043--

```

```

L10043:

```


TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-15

SEQ 0121

TEST 5: SECOND INITIALIZATION TEST

783	026716	005002		CLR	R2	;MEMORY EXPECTED SHOULD BE 000000		
784	026720	004737	016416	JSR	PC,CHKTSSR	;WAIT FOR READY, NON-AMBIGUOUS		
785	026724	010465	000000	MOV	R4,TSD8(R5)	;SELECT LOCATION SPECIFIED		
786	026730	004737	016416	JSR	PC,CHKTSSR	;WAIT FOR READY, NON-AMBIGUOUS		
787	026734	116501	000000	MOVB	TSDA(R5),R1	;READ LOC CONTENTS		
788	026740	120102		CMPB	R1,R2	;CHECK MEMORY FOR 000000		
789	026742	001406		BEB	40+	;BRANCH IF DATA OKAY		
790	026744			ERRDF	ERRNO,TSMEM,SFFMSG	;MEMORY NOT ZERO AFTER INIT.		
	026744	104455					TRAP	C#ERDF
	026746	000766					.WORD	502
	026750	027430					.WORD	TSMEM
	026752	012152					.WORD	SFFMSG
791	026754	005237	002220	INC	FATFLG	;SET THE FATAL ERROR FLAG		
792	026760			CKLOOP				
	026760	104406					TRAP	C#CLP1
793	026762			ESCAPE	SUB	;EXIT ON FATAL ERROR		
	026762	104410					TRAP	C#ESCAPE
	026764	000012					.WORD	L10050-
794	026766	005204		INC	R4	;LOOK AT NEXT RAM LOC.		
795	026770	020427	000400	CMP	R4,#400	;AT TOP OF RAM ADDRESS SPACE		
796	026774	001353		BNE	30+	;BRANCH TILL ALL MEMORY TESTED		
797								
798	026776			ENDSUB		;////////// END SUBTEST //////////		
	026776					L10050:		
	026776	104403					TRAP	C#ESUB
799								
800	027000	005737	002220	TST	FATFLG	;IS FATAL ERROR FLAG SET ?		
801	027004	001404		BEB	50+	;BRANCH IF NOT		
802	027006	004737	017262	JSR	PC,CKDROP	;NO LOOP, TRY TO DROP DEVICE		
803	027012	005037	002220	CLR	FATFLG	;CLEAR THE FATAL ERROR FLAG		
804	027016							
805								
806	027016			BGN SUB		;////////// BEGIN SUBTEST //////////		
	027016					TS.2:		
	027016	104402					TRAP	C#BSUB
807								
808	027020	004737	016054	JSR	PC,SOFINIT	;DO A SOFT TO START		
809	027024	103404		BCS	10+	;BRANCH IF O.K.		
813	027026			ERRDF	ERRNO,SFIERR,SFIMSG	;REPORT ERROR AND DROP DRIVE		
	027026	104455					TRAP	C#ERDF
	027030	000767					.WORD	503
	027032	003652					.WORD	SFIERR
	027034	012104					.WORD	SFIMSG
814	027036	012702	177777	MOV	#-1,R2	;ALL ONE DATA PATTERN		
815	027042	005004		CLR	R4	;STARTING RAM ADDRESS		
816	027044	004737	016416	JSR	PC,CHKTSSR	;WAIT FOR READY, NON-AMBIGUOUS		
817	027050	105065	000000	CLRB	TSD8(R5)	;SET MAINTENANCE MODE		
818	027054	004737	016416	JSR	PC,CHKTSSR	;WAIT FOR READY, NON-AMBIGUOUS		
819	027060	010465	000000	MOV	R4,TSD8(R5)	;SET THE NEXT RAM ADDRESS		
820	027064	004737	016416	JSR	PC,CHKTSSR	;WAIT FOR READY, NON-AMBIGUOUS		
821	027070	110265	000000	MOVB	R2,TSD8(R5)	;LOAD TEST DATA		
822	027074	005204		INC	R4	;NEXT ADDRESS TO TEST		
823	027076	020427	007777	CMP	R4,#7777	;COMPARE TO LAST ADDRESS		
824	027102	003762		BLE	15+	;BRANCH TILL ALL DATA WRITTEN		
825	027104	005065	000002	CLR	TSSR(R5)	;ISSUE A SOFT RESET		
826	027110	004737	016416	JSR	PC,CHKTSSR	;WAIT FOR READY, NON-AMBIGUOUS		
827	027114	016501	000002	MOV	TSSR(R5),R1	;GET THE CONTENTS OF TSSR		

TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-16

SEQ 0122

TEST 5: SECOND INITIALIZATION TEST

Address	Hex	Dec	Label	Op	Opnd	Comment	Trap	Trap Data
828	027120	010102		MOV	R1,R2	;CONTENTS OF TSSR		
829	027122	042702	176277	BIC	#1C<HIADDR!OFL>,R2	;THESE BITS MAY BE SET		
830	027126	052702	002200	BIS	#SSR!NBA,R2	;READY AND NEW DATA SHOULD BE SET		
831	027132	020102		CMP	R1,R2	;COMPARE EXPECTED TO RECEIVED		
832	027134	001406		BEQ	204	;BRANCH IF COMPARE		
836	027136			ERRDF	ERRNO,SFHERR,SFFMSG	;REPORT A FATAL ERROR		
	027136	104455					TRAP	C4ERDF
	027140	000770					.WORD	504
	027142	003705					.WORD	SFHERR
	027144	012152					.WORD	SFFMSG
837	027146	005237	002220	INC	FATFLG	;SET FATAL ERROR FLAG		
838	027152		204:	CKLOOP		;LOOP ON ERROR IF FLAG SET		
	027152	104406					TRAP	C4CLP1
839	027154			ESCAPE	SUB	;EXIT IF FATAL ERROR DETECTED		
	027154	104410					TRAP	C4ESCAPE
	027156	000170					.WORD	L10051-
840	027160	004737	016416	JSR	PC,CHKTSSR	;WAIT FOR SSR TO SET		
841	027164	105065	000000	CLRB	TSDB(R5)	;PUT BACK INTO MAINTENANCE MODE		
842	027170	004737	016416	JSR	PC,CHKTSSR	;WAIT FOR READY, NON-AMBIGUOUS		
843	027174	005065	000000	CLR	TSDB(R5)	;SET ADDRESS BACK TO 0000		
844	027200	012702	000377	MOV	#377,R2			
845	027204	004737	016416	JSR	PC,CHKTSSR	;WAIT FOR READY, NON-AMBIGUOUS		
846	027210	110265	000000	MOVB	R2,TSDB(R5)	;SHOULD POINT TO RAM 0		
847	027214	004737	016416	JSR	PC,CHKTSSR	;WAIT FOR READY, NON-AMBIGUOUS		
848	027220	005065	000000	CLR	TSDB(R5)	;SELECT LOCATION 0		
849	027224	004737	016416	JSR	PC,CHKTSSR	;WAIT FOR READY, NON-AMBIGUOUS		
850	027230	116501	000000	MOVB	TSBA(R5),R1	;READ RAM LOCATION SPECIFIED		
851	027234	120102		CMPB	R1,R2	;LOCATION SHOULD BE 377 OCTAL		
852	027236	001406		BEQ	254	;BR IF OK		
853	027240			ERRDF	ERRNO,TSADDR,EXPREC	;WASN'T POINTING TO CORRECT LOC.		
	027240	104455					TRAP	C4ERDF
	027242	000770					.WORD	504
	027244	027466					.WORD	TSADDR
	027246	015554					.WORD	EXPREC
854	027250	005237	002220	INC	FATFLG	;SET THE FATAL ERROR FLAG		
855	027254		254:	CKLOOP		;SCOPE LOOP		
	027254	104406					TRAP	C4CLP1
856	027256			ESCAPE	SUB	;NO MORE CHECKS IF FATAL ERROR		
	027256	104410					TRAP	C4ESCAPE
	027260	000066					.WORD	L10051-
857	027262	012704	000310	MOV	#310,R4	;START WITH LOC 310		
858	027266	005002		CLR	R2	;MEMORY EXPECTED SHOULD BE 000000		
859	027270	004737	016416	JSR	PC,CHKTSSR	;WAIT FOR READY, NON-AMBIGUOUS		
860	027274	010465	000000	MOV	R4,TSDB(R5)	;SELECT LOCATION SPECIFIED		
861	027300	004737	016416	JSR	PC,CHKTSSR	;WAIT FOR READY, NON-AMBIGUOUS		
862	027304	116501	000000	MOVB	TSBA(R5),R1	;READ LOC CONTENTS		
863	027310	120102		CMPB	R1,R2	;CHECK MEMORY FOR 000000		
864	027312	001406		BEQ</				

G10

TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-17

SEQ 0123

TEST 5: SECOND INITIALIZATION TEST

```

      027332 104410
      027334 000012
869 027336 005204
870 027340 020427 000400
871 027344 001353
872
873 027346
      027346
      027346 104403
874
875 027350 005737 002220
876 027354 001402
877 027356 004737 017262
878 027362 004737 016536
879 027366 103002
880 027370 000137 026444
881 027374
      027374 104432
      027376 000132
882
883
884
885
886
887 027400 105 170 164 TST5ID: .ASCIZ 'Extended Initialization'
888 027430 111 156 143 TSMEM: .ASCIZ 'Incorrect RAM Data After Init'
889 027466 111 156 143 TSADDR: .ASCIZ 'Incorrect RAM Address After Init'
890
891 027530
      027530
      027530 104401
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

```

INC R4
 CMP R4,#400
 BNE 30\$
 END SUB
 TST FATFLG
 BEQ 50\$
 JSR PC,CKDROP
 JSR PC,TSTLOOP
 BCC 60\$
 JMP TSTLOOP
 EXIT TST

;LOOK AT NEXT RAM LOC.
 ;AT TOP OF RAM ADDRESS SPACE
 ;BRANCH TILL ALL MEMORY TESTED
 ;////////////////// END SUBTEST ////////////////////
 L10051: TRAP C#ESUB
 ;IS FATAL ERROR FLAG SET ?
 ;BRANCH IF NOT
 ;NO LOOP, TRY TO DROP DEVICE
 ;SHOULD WE DO ITERATIONS ?
 ;BRANCH IF NOT
 ;LOOP UNTIL COUNT EXPIRED
 ;ALL DONE THIS TEST
 TRAP C#EXIT
 .WORD L10047-.

;*
 ;LOCAL TEXT MESSAGES FOR TEST
 ;-

L10047: TRAP C#ETST

.SBTTL 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 DATI 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. SUBTEST 1
; SETS UP THE INTERRUPT SERVICE ROUTINE TO FLAG UNEXPECTED
; INTERRUPTS. THE COMMAND WORD IN THE COMMAND BUFFER IS
; INITIALIZED TO 100000 (OCTAL) AND THE REMAINING THREE WORDS IN
; THE COMMAND BUFFER ARE SET TO KNOWN UNIQUE PATTERNS. THEN THE
; FOLLOWING SEQUENCE IS PERFORMED:

```

1. INITIALIZE THE CONTROLLER BY WRITING INTO THE TSSR; PROPER INITIAL CONDITIONS ARE VERIFIED.

TEST 6: COMMAND REJECT

976	027604		ERRDF	ERRNO,SFIERR,SFIMSG	;DEVICE FATAL ERROR DURING INIT		
	027604	104455				TRAP	C#ERDF
	027606	001131				.WORD	601
	027610	003652				.WORD	SFIERR
	027612	012104				.WORD	SFIMSG
977	027614	005037	002220	104:	CLR	FATFLG	;CLEAR FATAL ERROR FLAG
978	027620	005037	002222		CLR	INTRECV	;CLEAR INTERRUPT RECEIVED FLAG
979	027624	004737	016416		JSR	PC,CHKTSSR	;WAIT FOR READY, NON-AMBIGUOUS
980	027630	042714	000200		BIC	#BIT7,(R4)	;DISABLE INTERRUPTS
981	027634	010465	000000		MOV	R4,TSDB(R5)	;SET THE PACKET ADDRESS
982	027640	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET
983	027644	103407			BCS	154	;BR IF CARRY SET (GOOD RETURN)
984	027646	010001			MOV	R0,R1	;SAVE CONTENTS OF TSSR
988	027650				ERRDF	ERRNO,T6SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET
	027650	104455				TRAP	C#ERDF
	027652	001132				.WORD	602
	027654	030475				.WORD	T6SSR
	027656	012116				.WORD	PKTSSR
989	027660	005237	002220	154:	INC	FATFLG	;SET FATAL ERROR FLAG
990	027664				CKLOOP		;LOOP ON ERROR, IF FLAG SET
	027664	104406				TRAP	C#CLP1
991	027666				ESCAPE	SUB	;BY-PASS SUBTEST IF FATAL ERROR
	027666	104410				TRAP	C#ESCAPE
	027670	000170				.WORD	L10053-
992	027672	005737	002222		TST	INTRECV	;DID AN INTERRUPT OCCUR ?
993	027676	001404			BEQ	224	;BRANCH IF NOT
997	027700				ERRHRD	ERRNO,T6INT,PKTSSR	
	027700	104456				TRAP	C#ERHRD
	027702	001133				.WORD	603
	027704	030553				.WORD	T6INT
	027706	012116				.WORD	PKTSSR
998	027710	012702	102206	224:	MOV	#SC!NBA!SSR!TSREJ,R2	;EXPECTED CONTENTS OF TSSR
999	027714	004737	016416		JSR	PC,CHKTSSR	;WAIT FOR READY, NON-AMBIGUOUS
1000	027720	016501	000002		MOV	TSSR(R5),R1	;GET THE CONTENTS OF TSSR
1001	027724	032701	000100		BIT	#OFL,R1	;IS OFF-LINE BIT SET ?
1002	027730	001402			BEQ	254	;BRANCH IF NOT OFF-LINE
1003	027732	052702	000100		BIS	#OFL,R2	;SET OFF-LINE IN EXPECTED DATA
1004	027736	020201		254:	CMP	R2,R1	;DOES EXPECTED MATCH RECEIVED ?
1005	027740	001404			BEQ	304	;OKAY IF MATCH
1009	027742				ERRHRD	ERRNO,T6NBA,PKTSSR	;NBA NOT SET TO REJECT
	027742	104456				TRAP	C#ERHRD
	027744	001134				.WORD	604
	027746	030450				.WORD	T6NBA
	027750	012116				.WORD	PKTSSR
1010	027752			304:	CKLOOP		;LOOP ON ERROR ?
	027752	104406				TRAP	C#CLP1
1011	027754	004737	016416		JSR	PC,CHKTSSR	;WAIT FOR READY, NON-AMBIGUOUS
1012	027760	016501	000000		MOV	TSBA(R5),R1	;GET TSBA REGISTER CONTENTS
1013	027764	010402			MOV	R4,R2	;START OF THE PACKET
1014	027766	062702	000010		ADD	#10,R2	;EXPECT TSBA TO PACKET + 10
1015	027772	020102			CMP	R1,R2	;COMPARE EXPECTED TO RECEIVED
1016	027774	001404			BEQ	354	;ERROR IF NOT EQUAL
1020	027776				ERRHRD	ERRNO,T6TSBA,EXPREC	;PRINT THE ERROR & EXPD/RECV
	027776	104456				TRAP	C#ERHRD
	030000	001135				.WORD	605
	030002	030711				.WORD	T6TSBA
	030004	015554				.WORD	EXPREC

SEQ 0126

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

TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-21

SEQ 0127

TEST 6: COMMAND REJECT

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

L10

TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-22

SEQ 0128

TEST 6: COMMAND REJECT

```

1113 030354 042700 177740      BIC      #177740,R0      ;GET BITS 0-4
1114 030360 020027 000004      CMP      R0,#4        ;DON'T TEST WRITE CHARACTERISTICS
1115 030364 001002              BNE      45$          ;BRANCH IF NOT WRITE CHARACTERISTICS
1116 030366 062703 000002      ADD      #2,R3        ;BY-PASS WRITE CHARACTERISTICS
1117 030372 020327 003062      45$:    CMP      R3,#TBLEND    ;HAVE WE COMPLETED DATA TABLE ?
1118 030376 103002              BHS      50$          ;BRANCH IF ALL TESTED
1119 030400 000137 030114      JMP      5$          ;TEST WITH NEXT DATA
1120
1121 030404              50$:    ENDSUB                ;///////////////// END SUBTEST ///////////////////
1122 030406 005737 002220      TST      FATFLG        ;ANY FATAL ERRORS ?
1123 030412 001402              BEQ      60$          ;BRANCH IF NOT
1124 030414 004737 017262      JSR      PC,CKDROP      ;TRY TO DROP THE UNIT
1125 030420 004737 016536      60$:    JSR      PC,TSTLOOP    ;SHOULD WE DO ITERATIONS ?
1126 030424 103002              BCC      62$          ;BRANCH IF NOT
1127 030426 000137 027550      JMP      T6LOOP      ;LOOP UNTIL COUNT EXPIRED
1128 030432              62$:    EXIT      TST          ;ALL DONE THIS TEST
1129 030432 104432              .WORD    C$EXIT
1130 030434 000346              .WORD    L10052-.
1131
1132      ;* LOCAL STORAGE FOR THIS TEST
1133      ;*
1135 030436      .BLKB      10-<.-TSV2&7>
1137 030440      T6PACKET:  .WORD      0              ;COMMAND PACKET FOR TEST
1138 030440 000000      .WORD      052525            ;WILL CONTAIN VARIABLE COMMANDS
1139 030442 052525      .WORD      125252
1140 030444 125252      .WORD      052525
1141 030446 052525      .WORD
1142
1143
1144      ;* LOCAL TEXT MESSAGES FOR TEST
1145      ;*
1146
1147
1148 030450 103 157 155 T6NBA:  .ASCIZ  'Command Not Rejected'
1149 030475 103 157 156 T6SSR:  .ASCIZ  'Contents of TSSR Incorrect After Write Packet'
1150 030553 125 156 145 T6INT:  .ASCIZ  'Unexpected Interrupt Received On Write Packet'
1151 030631 105 170 160 T6NINT: .ASCIZ  'Expected Interrupt Not Received On Write Packet'
1152 030711 111 156 143 T6TSBA: .ASCIZ  'Incorrect TSBA Address After Packet Write'
1153 030763 103 157 155 TST6ID: .ASCIZ  'Command Reject'
1154
1155 031002      .EVEN
1156 031002      .EVEN
1157 031002 104401      .EVEN
1158
1159      .SBTTL  TEST  7: WRITE CHARACTERISTICS
1160
1161      ; THIS TEST VERIFIES BASIC OPERATION OF THE WRITE CHARACTERISTICS
1162      ; COMMAND. IT VERIFIES THAT THE COMMAND BLOCK AND CHARACTERISTICS
1163      ; DATA BLOCK ARE FETCHED PROPERLY FROM CPU MEMORY, THE NEED BUFFER
1164      ; ADDRESS (NBA) BIT IN TSSR IS HANDLED PROPERLY, AND THAT A PROPER
1165      ; MESSAGE PACKET IS STORED, WHERE APPROPRIATE. THIS TEST DOES NOT
1166      ; CHECK THAT THE VARIOUS FUNCTIONS ENABLED BY CHARACTERISTIC MODE

```


TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-23

TEST 7: WRITE CHARACTERISTICS

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

SEQ 0131

Address	Offset	Label	Instruction	Comment	Trap	Trap Value
1268	031340	001404	BEG			
1272	031342		ERRHRD	354 ERRNO,T7TSBA,EXPREC		
	031342	104456				
	031344	001303				
	031346	034300				
	031350	015554				
1273						
1274						
1275	031352	004737	354: JSR	PC,CKRAM		
1276	031356	103404	BCS	404		
1280	031360		ERRHRD	ERRNO,PKTRAM,RAMERR		
	031360	104456				
	031362	001304				
	031364	004745				
	031366	015570				
1281						
1282	031370		404: ENDSEG			
	031370					
	031370	104405				
1283						
1284	031372	012364	MOV	(R3)+,PKBCNT(R4)		
1285	031376	020327	CMP	R3,#TBLEND		
1286	031402	103002	BHIS	554		
1287	031404	000137	JMP	54		
1288						
1289	031410		554: ENDSUB			
	031410					
	031410	104403				
1290						
1291	031412	005737	TST	FATFLG		
1292	031416	001402	BEG	604		
1293	031420	004737	JSR	PC,CKDROP		
1294	031424		604:			
1295						
1296						
1297						
1298						
1299						
1300						
1301						
1302						
1303						
1304						
1305	031424		BGNSUB			
	031424					
	031424	104402				
1306						
1307	031426		SETPRI	#PRI00		
	031426	012700				
	031432	104441				
1308	031434	012703	MOV	#T72DATA,R3		
1309	031440	012704	MOV	#T7PACKET,R4		
1310	031444	004737	JSR	PC,T7REST		
1311						
1312	031450		BGNSEG			
	031450	104404				
1313						

C11

TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-26

SEQ 0132

TEST 7: WRITE CHARACTERISTICS

1314	031452	004737	016054		JSR	PC,SOFINIT		;DO SOFT INIT OF CONTROLLER	
1315	031456	103405			BCS	104		;BR IF SOFT INIT = OK	
1319	031460	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR	
1320	031462				ERRDF	ERRNO,SFIERR,SFIMSG		;DEVICE FATAL ERROR DURING INIT	
	031462	104455						TRAP	C4ERDF
	031464	001305						.WORD	709
	031466	003652						.WORD	SFIERR
	031470	012104						.WORD	SFIMSG
1321	031472	005037	002222	104:	CLR	INTRECV		;CLEAR INTERRUPT RECEIVED FLAG	
1322	031476	010400			MOV	R4,R0		;START OF THE COMMAND PACKET	
1323	031500	061300			ADD	(R3),R0		;OFFSET TO THE DATA WORD TO TEST	
1324	031502	056310	000002		BIS	2(R3),(R0)		;SET THE DATA BITS TO BE TESTED	
1325	031506	010465	000000		MOV	R4,TSD8(R5)		;SET THE PACKET ADDRESS	
1326	031512	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET	
1327	031516	103405			BCS	154		;BR IF CARRY SET (GOOD RETURN)	
1328	031520	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR	
1332	031522				ERRDF	ERRNO,T7SSR,PKTSSR		;DEVICE FATAL SSR FAILED TO SET	
	031522	104455						TRAP	C4ERDF
	031524	001306						.WORD	710
	031526	034031						.WORD	T7SSR
	031530	012116						.WORD	PKTSSR
1333	031532			154:	CKLOOP			;LOOP ON ERROR, IF FLAG SET	
1334	031532	104406			ESCAPE	SEG		;BY-PASS CHECKS IF FATAL ERROR	
	031534							TRAP	C4CLP1
	031534	104410						TRAP	C4ESCAPE
	031536	000116						.WORD	100004-
1335	031540	005737	002222		TST	INTRECV		;DID AN INTERRUPT OCCUR ?	
1336	031544	001404			BEQ	224		;BRANCH IF NOT	
1340	031546				ERRHRD	ERRNO,T7INT,PKTSSR			
	031546	104456						TRAP	C4ERHRD
	031550	001307						.WORD	711
	031552	034211						.WORD	T7INT
	031554	012116						.WORD	PKTSSR
1341	031556	016501	000002	224:	MOV	TSSR(R5),R1		;GET THE CONTENTS OF TSSR	
1342	031562	012702	102206		MOV	#SC!SSR!TSREJ!NBA,R2		;EXPECTED CONTENTS OF TSSR	
1343	031566	032701	000100		BIT	#OFL,R1		;IS OFF-LINE BIT SET ?	
1344	031572	001402			BEQ	254		;BRANCH IF NOT OFF-LINE	
1345	031574	052702	000100		BIS	#OFL,R2		;SET OFF-LINE IN EXPECTED DATA	
1346	031600	020201		254:	CMP	R2,R1		;DOES EXPECTED MATCH RECEIVED ?	
1347	031602	001414			BEQ	304		;OKAY IF MATCH	
1348	031604	010100			MOV	R1,R0		;DATA FROM TSSR	
1349	031606				XOR	R2,R0		;FIND BITS IN ERROR	
1350	031616	020027	002000		CMP	R0,#NBA		;IS NBA ONLY BIT IN ERROR ?	
1351	031622	001404			BEQ	304		;DON'T PRINT ERROR IF NBA ONLY BAD BIT	
1355	031624				ERRHRD	ERRNO,T72REJ,PKTSSR		;COMMAND NOT REJECTED	
	031624	104456						TRAP	C4ERHRD
	031626	001310						.WORD	712
	031630	033443						.WORD	T72REJ
	031632	012116						.WORD	PKTSSR
1356	031634			304:	CKLOOP			;LOOP ON ERROR ?	
	031634	104406						TRAP	C4CLP1
1357	031636	032701	002000		BIT	#NBA,R1		;IS NBA BIT SET ?	
1358	031642	001004			BNE	354		;OKAY IF NBA SET	
1362	031644				ERRHRD	ERRNO,T72NBA,PKTSSR		;NBA NOT SET	
	031644	104456						TRAP	C4ERHRD
	031646	001311						.WORD	713
	031650	033312						.WORD	T72NBA

D11

TSVSA - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-27

SEQ 0133

TEST 7: WRITE CHARACTERISTICS

```

1363 031652 012116
1364 031654 104405
1365 031656 062703 000004
1366 031662 020327 033312
1367 031666 103002
1368 031670 000137 031440
1369 031674
1370 031674 104403
1371 031676
1372 031676 104402
1373 031700
1374 031700 012700 000000
1375 031704 104441
1376 031706 012703 000001
1377 031712 012704 033230
1378 031716 004737 034412
1379 031722
1380 031722 104404
1381 031724 004737 016054
1382 031730 103405
1383 031732 010001
1384 031734 104455
1385 031736 001312
1386 031740 003652
1387 031742 012104
1388 031744 005037 002222
1389 031750 010364 000006
1390 031754 010465 000000
1391 031760 004737 016330
1392 031764 103405
1393 031766 010001
1394 031770 104455
1395 031772 001313
1396 031774 034031
1397 031776 012116

354:  ENDSEG

ADD    #4,R3
CMP    R3,#T72DONE
BHS    574
JMP     54

574:  ENDSUB

;***** END SEGMENT *****
100004: TRAP    C#ESEG

;POINT TO NEXT DATA PAIR
;COMPARE TO END OF TEST DATA
;BRANCH IF ALL DATA TESTED
;BRANCH TILL BACK TO ZERO

;////////// END SUBTEST //////////
L10057: TRAP    C#ESUB

;*
;
;TEST 7, SUBTEST 3
;
;CHECK THE WRITE CHARACTERISTICS COMMAND IS REJECTED
;IF ISSUED WITH AN INVALID DATA BLOCK BYTE COUNT
;
;-

BGNSUB

;////////// BEGIN SUBTEST //////////
T7.3:  TRAP    C#BSUB

;LOWER PRIORITY TO ALLOW INTERRUPTS
MOV    #PRI00,R0
TRAP    C#SPRI

54:  MOV    #1,R3
MOV    #T7PACKET,R4
JSR    PC,T7REST

;STARTING BYTE COUNT
;GET THE ADDRESS OF COMMAND PACKET
;RESTORE PACKET TO STARTING VALUES

;***** BEGIN SEGMENT *****
TRAP    C#BSEG

JSR    PC,SOFINIT
BCS    104
MOV    R0,R1
ERRDF  ERRNO,SFIERR,SFIMSG

;DO SOFT INIT OF CONTROLLER
;BR IF SOFT INIT = OK
;SAVE CONTENTS OF TSSR
;DEVICE FATAL ERROR DURING INIT
TRAP    C#ERDF
WORD    714
WORD    SFIERR
WORD    SFIMSG

104:  CLR    INTRECV
MOV    R3,PKBCNT(R4)
MOV    R4,TSDB(R5)
JSR    PC,WAITF
BCS    154
MOV    R0,R1
ERRDF  ERRNO,T7SSR,PKTSSR

;CLEAR INTERRUPT RECEIVED FLAG
;INSERT THE BYTE COUNT FOR TEST
;SET THE PACKET ADDRESS
;WAIT FOR SSR TO SET
;BR IF CARRY SET (GOOD RETURN)
;SAVE CONTENTS OF TSSR
;DEVICE FATAL SSR FAILED TO SET
TRAP    C#ERDF
WORD    715
WORD    T7SSR
WORD    PKTSSR

```

TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-28

SEQ 0134

TEST 7: WRITE CHARACTERISTICS

1408	032000		154:	CKLOOP		;LOOP ON ERROR, IF FLAG SET	TRAP	C4CLP1
1409	032002	104406		ESCAPE	SEG	;BY-PASS SUBTEST IF FATAL ERROR	TRAP	C4CLP1
	032002	104410					TRAP	C4ESCAPE
	032004	000116					.WORD	100004-
1410	032006	005737	002222	TST	INTRECV	;DID AN INTERRUPT OCCUR ?		
1411	032012	001404		BEG	224	;BRANCH IF NOT		
1415	032014			ERRHRD	ERRNO,T7INT,PKTSSR			
	032014	104456					TRAP	C4ERHRD
	032016	001314					.WORD	716
	032020	034211					.WORD	T7INT
	032022	012116					.WORD	PKTSSR
1416	032024	016501	000002	224:	MOV TSSR(R5),R1	;GET THE CONTENTS OF TSSR		
1417	032030	012702	102206		MOV #SC!SSR!TSREJ:NBA,R2	;EXPECTED CONTENTS OF TSSR		
1418	032034	032701	000100		BIT #OFL,R1	;IS OFF-LINE BIT SET ?		
1419	032040	001402			BEG 254	;BRANCH IF NOT OFF-LINE		
1420	032042	052702	000100		BIS #OFL,R2	;SET OFF-LINE IN EXPECTED DATA		
1421	032046	020201		254:	CMP R2,R1	;DOES EXPECTED MATCH RECEIVED ?		
1422	032050	001414			BEG 304	;OKAY IF MATCH		
1423	032052	010100			MOV R1,R0	;DATA FROM TSSR		
1424	032054				XOR R2,R0	;FIND BITS IN ERROR		
1425	032064	020027	002000		CMP R0,#NBA	;IS NBA ONLY BIT IN ERROR ?		
1426	032070	001404			BEG 304	;DON'T PRINT ERROR IF NBA ONLY BAD BIT		
1430	032072			ERRHRD	ERRNO,T73REJ,PKTSSR	;COMMAND NOT REJECTED		
	032072	104456					TRAP	C4ERHRD
	032074	001315					.WORD	717
	032076	033542					.WORD	T73REJ
	032100	012116					.WORD	PKTSSR
1431	032102			304:	CKLOOP	;LOOP ON ERROR ?		
	032102	104406					TRAP	C4CLP1
1432	032104	032701	002000		BIT #NBA,R1	;IS NBA BIT SET ?		
1433	032110	001004			BNE 354	;OKAY IF NBA SET		
1437	032112			ERRHRD	ERRNO,T72NBA,PKTSSR	;NBA NOT SET		
	032112	104456					TRAP	C4ERHRD
	032114	001316					.WORD	718
	032116	033312					.WORD	T72NBA
	032120	012116					.WORD	PKTSSR
1438	032122			354:				
1439	032122			ENDSEG		;<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<<		
	032122					100004:		
	032122	104405					TRAP	C4ESEG
1440								
1441	032124	005203		INC	R3	;NEXT BYTE COUNT		
1442	032126	020327	000006	CMP	R3,#6	;TESTED ALL INVALID ?		
1443	032132	002002		BGE	574	;BRANCH IF TEST DONE		
1444	032134	000137	031712	JMP	54	;BRANCH TILL BACK TO ZERO		
1445								
1446	032140			574:	ENDSUB	;\\\\\\\\\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\\\\\\		
	032140					L10060:		
	032140	104403					TRAP	C4ESUB
1447								
1448								
1449								
1450								
1451								
1452								
1453								

F11

TEST 7: WRITE CHARACTERISTICS

```

1454      032142      104402      BGNSUB      ;//////////////// BEGIN SUBTEST //////////////////
1455      032142      104402      T7.4:      TRAP      C4BSUB
1456      032142      104402      1458      032144      SETPRI      #PRI00      ;LOWER PRIORITY TO ALLOW INTERRUPTS
1457      032142      104402      1459      032144      012700      000000      MOV      #PRI00,R0
1458      032144      012700      000000      032150      104441      TRAP      C4SPRI
1459      032152      012703      033276      1460      032152      012703      033276      ;START OF TEST DATA FOR SUBTEST
1460      032156      012704      033230      1461      032156      012704      033230      ;GET THE ADDRESS OF COMMAND PACKET
1461      032162      004737      034412      1462      032162      004737      034412      ;RESTORE PACKET TO STARTING VALUES
1462      032162      004737      034412      1463      032166      004737      016054      JSR      PC,SOFINIT
1463      032166      004737      016054      1464      032172      103405      BCS      104
1464      032172      103405      1465      032174      010001      MOV      R0,R1
1465      032176      010001      1466      032176      104455      ERDF      ERRNO,SFIERR,SFIMSG
1466      032176      104455      1467      032200      001317      TRAP      C4ERDF
1467      032202      003652      1468      032202      003652      .WORD      719
1468      032204      012104      1469      032204      012104      .WORD      SFIERR
1469      032206      005037      002222      1470      032206      005037      002222      .WORD      SFIMSG
1470      032212      052737      000001      1471      032212      052737      000001      033240      104:      CLR      INTRECV
1471      032220      010465      000000      1472      032220      010465      000000      ;CLEAR INTERRUPT RECEIVED FLAG
1472      032224      004737      016330      1473      032224      004737      016330      ;MAKE ADDRESS ODD
1473      032230      103405      1474      032230      103405      ;SET THE PACKET ADDRESS
1474      032232      010001      1475      032232      010001      ;WAIT FOR SSR TO SET
1475      032234      104455      1476      032234      104455      ;BR IF CARRY SET (GOOD RETURN)
1476      032236      001320      1477      032236      001320      ;SAVE CONTENTS OF TSSR
1477      032240      034031      1478      032240      034031      ;DEVICE FATAL SSR FAILED TO SET
1478      032242      012116      1479      032242      012116      TRAP      C4ERDF
1479      032244      104406      1480      032244      104406      .WORD      720
1480      032246      104410      1481      032246      104410      .WORD      T7SSR
1481      032250      000116      1482      032250      000116      .WORD      PKTSSR
1482      032252      005737      002222      1483      032252      005737      002222      154:      CKLOOP
1483      032256      001404      1484      032256      001404      ;LOOP ON ERROR, IF FLAG SET
1484      032260      104456      1485      032260      104456      ;BY-PASS SUBTEST IF FATAL ERROR
1485      032262      001321      1486      032262      001321      TRAP      C4CLP1
1486      032264      034211      1487      032264      034211      ;DID AN INTERRUPT OCCUR ?
1487      032266      012116      1488      032266      012116      .WORD      C4ESCAPE
1488      032270      016501      000002      1489      032270      016501      000002      L10061-.
1489      032274      012702      102206      1490      032274      012702      102206      TST      INTRECV
1490      032300      032701      000100      1491      032300      032701      000100      BEQ      224
1491      032304      001402      1492      032304      001402      ;BRANCH IF NOT OFF-LINE
1492      032306      052702      000100      1493      032306      052702      000100      ;SET OFF-LINE IN EXPECTED DATA
1493      032312      020201      1494      032312      020201      ;DOES EXPECTED MATCH RECEIVED ?
1494      032314      001414      1495      032314      001414      ;OKAY IF MATCH
1495      032316      010100      1496      032316      010100      ;DATA FROM TSSR
1496      032320      020027      002000      1497      032320      020027      002000      ;FIND BITS IN ERROR
1497      032334      001404      1498      032334      001404      ;IS NBA ONLY BIT IN ERROR ?
1498      032334      001404      1499      032334      001404      ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
1499      032334      001404      1500      032334      001404

```


TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-31

SEQ 0137

TEST 7: WRITE CHARACTERISTICS

Address	Offset	Value	Label	Operation	Register/Value	Comment	Trap	Value
1550	032462	104455		ERRDF	ERRNO,T7SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET		
	032462	001325					TRAP	C\$ERDF
	032464	034031					.WORD	725
	032470	012116					.WORD	T7SSR
1551	032472	104406	15\$:	CKLOOP		;LOOP ON ERROR, IF FLAG SET	SET	PKTSSR
	032472	104406					TRAP	C\$CLP1
1552	032474	104410		ESCAPE	SEG	;BY-PASS SUBTEST IF FATAL ERROR		
	032474	000116					TRAP	C\$ESCAPE
	032476	000116					.WORD	10000\$-
1553	032500	005737	002222	TST	INTRECV	;DID AN INTERRUPT OCCUR ?		
1554	032504	001404		BEQ	22\$	;BRANCH IF NOT		
1558	032506	104456		ERRHRD	ERRNO,T7INT,PKTSSR			
	032506	001326					TRAP	C\$ERHRD
	032510	034211					.WORD	726
	032512	012116					.WORD	T7INT
	032514	016501					.WORD	PKTSSR
1559	032516	012702	000002	22\$:	MOV TSSR(R5),R1	;GET THE CONTENTS OF TSSR		
1560	032522	032701	102206		MOV #SC!SSR!TSREJ!NBA,R2	;EXPECTED CONTENTS OF TSSR		
1561	032526	001402	000100		BIT #OFL,R1	;IS OFF-LINE BIT SET ?		
1562	032532	052702	000100		BEQ 25\$	;BRANCH IF NOT OFF-LINE		
1563	032534	020201			BIS #OFL,R2	;SET OFF-LINE IN EXPECTED DATA		
1564	032540	001414		25\$:	CMP R2,R1	;DOES EXPECTED MATCH RECEIVED ?		
1565	032542	010100			BEQ 30\$	;OKAY IF MATCH		
1566	032544	020027	002000		MOV R1,R0	;DATA FROM TSSR		
1567	032546	001404			XOR R2,R0	;FIND BITS IN ERROR		
1568	032556	001404			CMP R0,#NBA	;IS NBA ONLY BIT IN ERROR ?		
1569	032562	104456			BEQ 30\$	;DON'T PRINT ERROR IF NBA ONLY BAD BIT		
1573	032564	001327		ERRHRD	ERRNO,T75REJ,PKTSSR	;COMMAND NOT REJECTED		
	032564	033733					TRAP	C\$ERHRD
	032572	012116					.WORD	727
1574	032574	104406	002000	30\$:	CKLOOP	;LOOP ON ERROR ?		
	032574	032701					TRAP	C\$CLP1
1575	032576	001004			BIT #NBA,R1	;IS NBA BIT SET ?		
1576	032602	104456			BNE 35\$	;OKAY IF NBA SET		
1580	032604	001330		ERRHRD	ERRNO,T72NBA,PKTSSR	;NBA NOT SET		
	032604	033312					TRAP	C\$ERHRD
	032606	012116					.WORD	728
	032610						.WORD	T72NBA
	032612						.WORD	PKTSSR
1581	032614			35\$:	ENDSEG	;<<<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<<<<<<		
1582	032614	104405					10000\$:	
	032614						TRAP	C\$ESEG
1583	032616	005203				;NEXT BUFFER LENGTH		
1584	032620	020327	000016		INC R3	;HAVE ALL BAD VALUES BEEN TESTED ?		
1585	032624	002002			CMP R3,#14.	;BRANCH IF ALL TESTED		
1586	032626	000137	032404		BGE 57\$	;BRANCH TILL BACK TO ZERO		
1587	032632				JMP 54			
1588	032632			57\$:	ENDSUB	;////////// END SUBTEST //////////		
1589	032632	104403					L10062:	
	032632							

TSVSA - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-33

SEQ 0139

TEST 7: WRITE CHARACTERISTICS

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

K11

SEQ 0140

TEST 7: WRITE CHARACTERISTICS

```

1687
1688 033202          55$:  ENDSUB          ;////////// END SUBTEST //////////
      033202          L10063:          TRAP      C$ESUB
      033202 104403
1689
1690 033204 005737 002220          TST      FATFLG          ;ANY FATAL ERRORS ?
1691 033210 001402          BEQ      60$          ;BRANCH IF NOT
1692 033212 004737 017262          JSR      PC,CKDROP        ;TRY TO DROP THE UNIT
1693 033216
1694 033216          60$:  EXIT      TST          ;ALL DONE THIS TEST
      033216 104432          TRAP      C$EXIT
      033220 001240          .WORD     L10055-.

1695
1696
1697 ;*
1698 ;LOCAL STORAGE FOR THIS TEST
1699 ;-
1701 033222          .BLKB      10-<.-TSV2&7>
1703 033230          T7PACKET:
1704 033230 100004          .WORD     100004          ;COMMAND PACKET FOR TEST
1705 033232 033240          .WORD     T7DATA          ;WRITE CHARACTERISTICS COMMAND, WITH ACK
1706 033234 000000          .WORD     0              ;ADDRESS OF CHARACTERISTICS BLOCK
1707 033236 000010          .WORD     8.              ;STARTING VALUE OF BLOCK SIZE
1708
1709 033240          T7DATA:
1710 033240 033256          .WORD     T7BFR          ;CHARACTERISTICS DATA BLOCK
1711 033242 000000          .WORD     0              ;ADDRESS OF MESSAGE BUFFER
1712 033244 000016          .WORD     14.            ;LENGTH OF MESSAGE BUFFER
1713 033246 000000          .WORD     0
1714 033250 000000          T7SP:   .WORD     0          ;EXTFEA EXTRA WORD
1715
1716 033252 000000 000000          T7BFR: .WORD     0,0          ;SPACE
1717 033256          .BLKW      8.              ;MESSAGE BUFFER
1718
1719 ;*
1720 ;TEST DATA FOR SUBTEST TWO
1721 ;
1722 ;DATA HAS FORMAT:
1723 ;
1724 ;          1ST WORD      OFFSET TO TEST WORD IN PACKET
1725 ;          2ND WORD      BITS TO SET FOR TEST
1726 ;
1727 ;-
1728
1729
1730 033276          T72DATA:
1731 033276 000000 037140          .WORD     0,BIT5!BIT6!BIT9!BIT10!BIT11!BIT12!BIT13
1732 033302 000002 000001          .WORD     2,BIT0
1733 033306 000004 100100          .WORD     4,BIT6!BIT15
1734
1735          T72DONE=.
1736
1737 ;*
1738 ;LOCAL TEXT MESSAGES FOR TEST
1739 ;-
1740
1741 033312          116      102      101  T72NBA: .ASCIZ  'NBA Not Set On Rejected WRITE CHARACTERISTICS'

```


L11

SEQ 0141

TEST 7: WRITE CHARACTERISTICS

1742	033370	127	122	111	T7NBA:	.ASCIZ	'WRITE CHARACTERISTICS Command Not Accepted'
1743	033443	127	122	111	T72REJ:	.ASCIZ	'WRITE CHARACTERISTICS Not Rejected With Non-Zero Unused Fields'
1744	033542	127	122	111	T73REJ:	.ASCIZ	'WRITE CHARACTERISTICS Not Rejected With Invalid Data Count'
1745	033635	127	122	111	T74REJ:	.ASCIZ	'WRITE CHARACTERISTICS Not Rejected With Invalid Block Address'
1746	033733	127	122	111	T75REJ:	.ASCIZ	'WRITE CHARACTERISTICS Not Rejected With Invalid Buffer Length'
1747	034031	103	157	156	T7SSR:	.ASCIZ	'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
1748	034120	105	170	160	T7NINT:	.ASCIZ	'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
1749	034211	125	156	145	T7INT:	.ASCIZ	'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
1750	034300	111	156	143	T7TSBA:	.ASCIZ	'Incorrect TSBA Address After WRITE CHARACTERISTICS'
1751	034363	127	162	151	TST7ID:	.ASCIZ	'Write Characteristics'
1752						.EVEN	

1753
1754
1755
1756
1757
1758
1759
1760

```

;
; ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
;
;-

```

1761 034412
1762 034412
1763 034416 012701 033230
1764 034422 012721 100004
1765 034426 012721 033240
1766 034432 005021
1767 034434 012721 000010
1768 034440 012721 033256
1769 034444 005021
1770 034446 012721 000020
1771 034452 005021
1772 034454 005011
1773 034456 000207
1774 034460
034460
034460 104401

```

T7REST: SAVREG          ;SAVE THE REGISTERS
MOV      #T7PACKET,R1    ;START OF THE PACKET
MOV      #100004,(R1)+    ;WRITE CHARACTERISTICS WITH ACK
MOV      #T7DATA,(R1)+    ;ADDRESS OF CHAR DATA BLOCK
CLR      (R1)+            ;EXTENDED ADDRESS
MOV      #8,(R1)+         ;SIZE OF DATA BLOCK IN BYTES
MOV      #T7BFR,(R1)+     ;ADDRESS OF MESSAGE BUFFER
CLR      (R1)+
MOV      #16,(R1)+        ;LENGTH OF MESSAGE BUFFER
CLR      (R1)+
CLR      (R1)
RTS      PC               ;RETURN
ENDTST

```

L10055: TRAP C#ETST

1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796

.SBTTL TEST 8: VOLUME CHECK

```

;
; 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;

M11

TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-36

SEQ 0142

TEST 8: VOLUME CHECK

```

1797                                     : THE VCK BIT SHOULD BE CLEAR (0).
1798                                     :
1799                                     :
1800                                     :
1801                                     :
1802                                     :
1803                                     :
1804                                     :
1805                                     :
1806                                     :
1807                                     :
1808                                     :
1809                                     :
1810                                     :
1811 034462                               BGNTST
1812 034462
1816 034462 012700 035347                MOV    #T8ID,R0
1817 034466 004737 016570                JSR    PC,TSTSETUP
1818 034472 012737 000024 002214        MOV    #20.,LOOPCNT
1819 034500
1820
1821 034500 012704 035070                MOV    #T8PACKET,R4
1822 034504 004737 016054                JSR    PC,SOFINIT
1823 034510 103405                        BCS    10$,
1827 034512 010001                        MOV    R0,R1
1828 034514                        ERDF    ERRNO,SFIERR,SFIMSG
1829 034514 104455                        TRAP    C#ERDF
1830 034516 001441                        .WORD 801
1831 034520 003652                        .WORD SFIERR
1832 034522 012104                        .WORD SFIMSG
1833 034524 042714 040000                10$: BIC    #BIT14,(R4)
1834 034530 010465 000000                MOV    R4,TSDB(R5)
1835 034534 004737 016416                JSR    PC,CHKTSSR
1836 034540 103405                        BCS    15$,
1837 034542 010001                        MOV    R0,R1
1838 034544                        ERDF    ERRNO,T8SSR,PKTSSR
1839 034544 104455                        TRAP    C#ERDF
1840 034546 001442                        .WORD 802
1841 034550 035260                        .WORD T8SSR
1842 034552 012116                        .WORD PKTSSR
1843 034554
1844 034554 104406                15$: CKLOOP
1845 034556                        ESCAPE TST
1846 034556 104410                        TRAP    C#CLP1
1847 034560 000604                        .WORD L10064-.
1848 034562 012702 035112                MOV    #T8BFR,R2
1849 034566 032762 000020 000006        BIT    #XSOVCK,XSTO(R2)
1850 034574 001406                        BEQ    20$,
1851 034576 016501 000002                MOV    TSSR(R5),R1
1852 034602                        ERHRD   ERRNO,T8NVCK,PKTMES
1853 034602 104456                        TRAP    C#ERHRD
1854 034604 001443                        .WORD 803
1855 034606 035167                        .WORD T8NVCK
1856 034610 012160                        .WORD PKTMES
1857 034612
1858 034612 104406                20$: CKLOOP
1859 034614 010465 000000                MOV    R4,TSDB(R5)

```


N11

TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-37

SEQ 0143

TEST 8: VOLUME CHECK

1850	034620	004737	016416		JSR	PC,CHKTSSR		;WAIT FOR SSR TO SET	
1851	034624	103405			BCS	254		;BR IF CARRY SET (GOOD RETURN)	
1852	034626	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR	
1856	034630				ERRDF	ERRNO,T8SSR,PKTSSR		;DEVICE FATAL SSR FAILED TO SET	
	034630	104455						TRAP	C4ERDF
	034632	001444						.WORD	804
	034634	035260						.WORD	T8SSR
	034636	012116						.WORD	PKTSSR
1857	034640			254:	CKLOOP			;LOOP ON ERROR, IF FLAG SET	
	034640	104406			ESCAPE	TST		TRAP	C4CLP1
1858	034642							;EXIT IF FATAL ERROR	
	034642	104410						TRAP	C4ESCAPE
	034644	000520						.WORD	L10064--
1859	034646	032762	000020	000006	BIT	#XSOVCK,XSTO(R2)		;IS VOLUME CHECK SET IN XSTO ?	
1860	034654	001406			BEQ	304		;OKAY IF VOLUME CHECK IS SET	
1864	034656	016501	000002		MOV	TSSR(R5),R1		;CONTENTS OF TSSR FOR ERROR REPORT	
1865	034662				ERRHRD	ERRNO,T8NVCK,PKTMES		;VOLUME CHECK NOT SET	
	034662	104456						TRAP	C4ERHRD
	034664	001445						.WORD	805
	034666	035167						.WORD	T8NVCK
	034670	012160						.WORD	PKTMES
1866	034672			304:	CKLOOP			;LOOP ON ERROR ?	
	034672	104406						TRAP	C4CLP1
1867	034674	052714	040000		BIS	#BIT14,(R4)		;SET THE CVC BIT	
1868	034700	010465	000000		MOV	R4,TSD8(R5)		;SET THE PACKET ADDRESS FOR WRITE CHAR	
1869	034704	004737	016416		JSR	PC,CHKTSSR		;WAIT FOR SSR TO SET	
1870	034710	103405			BCS	354		;BR IF CARRY SET (GOOD RETURN)	
1871	034712	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR	
1875	034714				ERRDF	ERRNO,T8SSR,PKTSSR		;DEVICE FATAL SSR FAILED TO SET	
	034714	104455						TRAP	C4ERDF
	034716	001446						.WORD	806
	034720	035260						.WORD	T8SSR
	034722	012116						.WORD	PKTSSR
1876	034724			354:	CKLOOP			;LOOP ON ERROR, IF FLAG SET	
	034724	104406			ESCAPE	TST		TRAP	C4CLP1
1877	034726							;EXIT IF FATAL ERROR	
	034726	104410						TRAP	C4ESCAPE
	034730	000434						.WORD	L10064--
1878	034732	032762	000020	000006	BIT	#XSOVCK,XSTO(R2)		;IS VOLUME CHECK CLEAR IN XSTO ?	
1879	034740	001406			BEQ	404		;OKAY IF VOLUME CHECK IS CLEARED	
1883	034742	016501	000002		MOV	TSSR(R5),R1		;CONTENTS OF TSSR FOR ERROR REPORT	
1884	034746				ERRHRD	ERRNO,T8VCK,PKTMES		;VOLUME CHECK NOT CLEARED	
	034746	104456						TRAP	C4ERHRD
	034750	001447						.WORD	807
	034752	035132						.WORD	T8VCK
	034754	012160						.WORD	PKTMES
1885	034756			404:	CKLOOP			;LOOP ON ERROR ?	
	034756	104406						TRAP	C4CLP1
1886	034760	042714	040000		BIC	#BIT14,(R4)		;CLEAR THE CVC BIT	
1887	034764	010465	000000		MOV	R4,TSD8(R5)		;SET THE PACKET ADDRESS FOR WRITE CHAR	
1888	034770	004737	016416		JSR	PC,CHKTSSR		;WAIT FOR SSR TO SET	
1889	034774	103405			BCS	454		;BR IF CARRY SET (GOOD RETURN)	
1890	034776	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR	
1894	035000				ERRDF	ERRNO,T8SSR,PKTSSR		;DEVICE FATAL SSR FAILED TO SET	
	035000	104455						TRAP	C4ERDF
	035002	001450						.WORD	808
	035004	035260						.WORD	T8SSR

B12

TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-38

SEQ 0144

TEST 8: VOLUME CHECK

```

1895 035006 012116          454:  CKLOOP          ;LOOP ON ERROR, IF FLAG SET .WORD PKTSSR
035010 104406          ;EXIT IF FATAL ERROR TRAP C4CLP1
1896 035012 104410          ESCAPE TST          ;IS VOLUME CHECK CLEAR IN XST0 ? .WORD C4ESCAPE
035014 000350          ;OKAY IF VOLUME CHECK IS CLEARED L10064-.
1897 035016 032762 000020 000006          BIT #XSOVCK,XST0(R2)
1898 035024 001406          BEQ 504
1902 035026 016501 000002          MOV TSSR(R5),R1
1903 035032 104456          ERRHRD ERRNO,T8VCK,PKTMES ;VOLUME CHECK NOT CLEARED
035034 001451          TRAP C4ERHRD
035036 035132          .WORD 809
035040 012160          .WORD T8VCK
1904 035042 104406          504:  CKLOOP          ;LOOP ON ERROR ? .WORD PKTMES
035044 004737 016536          604:  JSR PC,TSTLOOP          ;SHOULD WE DO ITERATIONS ? TRAP C4CLP1
1905 035050 103002          BCC 624
1906 035052 000137 034500          JMP T8LOOP          ;BRANCH IF NOT
1907 035056 104432          624:  EXIT TST          ;LOOP UNTIL COUNT EXPIRED
035060 000304          ;ALL DONE THIS TEST TRAP C4EXIT
1909          .WORD L10064-.
1910          ;*
1911          ;LOCAL STORAGE FOR THIS TEST
1912          ;-
1913
1915 035062          T8PACKET: .BLKB 10-<.-TSV2E7>
1917 035070          .WORD 100004          ;COMMAND PACKET FOR TEST
1918 035072 100004          .WORD T8DATA          ;WRITE CHARACTERISTICS COMMAND
1919 035074 000000          .WORD 0          ;ADDRESS OF CHARACTERISTICS BLOCK
1920 035076 000010          .WORD 10          ;STARTING VALUE OF COUNTER
1921
1922          T8DATA:          ;CHARACTERISTICS DATA BLOCK
1923 035100          .WORD T8BFR          ;ADDRESS OF MESSAGE BUFFER
1924 035102 035112          .WORD 0
1925 035104 000000          .WORD 16.
1926 035106 000000 000000          .WORD 0,0          ;LENGTH OF MESSAGE BUFFER
1927
1928          T8BFR: .BLKW 8.          ;MESSAGE BUFFER
1929
1930          ;*
1931          ;LOCAL TEXT MESSAGES FOR TEST
1932          ;-
1933
1935 035132 126 157 154 T8VCK: .ASCIZ 'Volume Check Bit Not Cleared'
1936 035167 126 157 154 T8NVCK: .ASCIZ 'Volume Check Bit (VCK) Not Clear After Initialize (XST0)'
1937 035260 103 157 156 T8SSR: .ASCIZ 'Contents of TSSR Incorrect After Write Characteristics'
1938 035347 126 157 154 TST8ID: .ASCIZ 'Volume Check'
1939
1940          .EVEN
1941 035364          .ENDTST
035364 104401          L10064: TRAP C4ETST
1942
1943          .SBTTL TEST 9: COMPLETION INTERRUPT

```


TEST 9: COMPLETION INTERRUPT

```

1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962 035366
      035366
1967 035366 005037 002224
1968 035372 012700 040421
1969 035376 004737 016570
1970 035402 012737 000024 002214
1971 035410
1972 035410
      035410
      035410 104402
1973
1974 035412 004737 040446
1975 035416
      035416 012700 000000
      035422 104441
1976 035424 012703 002764
1977 035430 012704 037350
1978 035434 012764 000010 000006
1979 035442
1980 035442
      035442 104404
1981
1982 035444 004737 016054
1983 035450 103405
1987 035452 010001
1988 035454
      035454 104455
      035456 001605
      035460 003652
      035462 012104
1989 035464 005037 002220
1990 035470 005037 002222
1991 035474 010465 000000
1992 035500 004737 016416
1993 035504 103407
1994 035506 010001
1998 035510
      035510 104455
      035512 001606

      :
      : 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.
      :
      :
      : BGNTST
      :
      : T9::
      : CLEAR EXTENDED FEATURES SWITCH
      : ASCII MESSAGE TO IDENTIFY TEST
      : DO INITIAL TEST SETUP
      : PERFORM 20 ITERATIONS
      :
      : BEGIN SUBTEST
      : T9.1:
      : TRAP C4BSUB
      :
      : SET PACKET TO INITIAL VALUES
      : LOWER PRIORITY TO ALLOW INTERRUPTS
      : MOV #PRI00,RO
      : TRAP C4SPRI
      :
      : START OF TEST DATA
      : GET THE ADDRESS OF COMMAND PACKET
      : START WITH MINIMUM ALLOWABLE VALUE
      :
      : BEGIN SEGMENT
      : TRAP C4BSEG
      :
      : DO SOFT INIT OF CONTROLLER
      : BR IF SOFT INIT = OK
      : SAVE CONTENTS OF TSSR
      : DEVICE FATAL ERROR DURING INIT
      : TRAP C4ERDF
      : .WORD 901
      : .WORD SFIERR
      : .WORD SFIMSG
      :
      : CLEAR FATAL ERROR FLAG
      : CLEAR INTERRUPT RECEIVED FLAG
      : SET THE PACKET ADDRESS
      : WAIT FOR SSR TO SET
      : BR IF CARRY SET (GOOD RETURN)
      : SAVE CONTENTS OF TSSR
      : DEVICE FATAL SSR FAILED TO SET
      : TRAP C4ERDF
      : .WORD 902

```

D12
TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-40

TEST 9: COMPLETION INTERRUPT

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

TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-41

SEQ 0147

TEST 9: COMPLETION INTERRUPT

2045					;WRITE CHARACTERISTICS COMMAND TO BE REJECTED	
2046					;	
2047					;-	
2048						
2049	035666			BGNSUB	;/!!!!!!!!!!!!!! BEGIN SUBTEST !!!!!!!!!!!!!!!	
	035666				T9.2:	
	035666	104402			TRAP C#BSUB	
2050						
2051	035670			SETPRI #PRI00	;LOWER PRIORITY TO ALLOW INTERRUPTS	
	035670	012700	000000		MOV #PRI00,R0	
	035674	104441			TRAP C#SPRI	
2052	035676	012703	037412	MOV #T92DATA,R3	;START OF TEST DATA FOR SUBTEST	
2053	035702	012704	037350	5#: MOV #T9PACKET,R4	;GET THE ADDRESS OF COMMAND PACKET	
2054	035706	004737	040446	JSR PC,T9REST	;RESTORE PACKET TO STARTING VALUES	
2055						
2056	035712			BGNSEG	;>>>>>>>>> BEGIN SEGMENT >>>>>>>>>	
	035712	104404			TRAP C#BSEG	
2057						
2058	035714	004737	016054	JSR PC,SOFINIT	;DO SOFT INIT OF CONTROLLER	
2059	035720	103405		BCS 10#	;BR IF SOFT INIT = OK	
2063	035722	010001		MOV R0,R1	;SAVE CONTENTS OF TSSR	
2064	035724			ERRDF ERRNO,SFIERR,SFIMSG	;DEVICE FATAL ERROR DURING INIT	
	035724	104455			TRAP C#ERDF	
	035726	001611			.WORD 905	
	035730	003652			.WORD SFIERR	
	035732	012104			.WORD SFIMSG	
2065	035734	005037	002222	10#: CLR INTRECV	;CLEAR INTERRUPT RECEIVED FLAG	
2066	035740	010400		MOV R4,R0	;START OF THE COMMAND PACKET	
2067	035742	061300		ADD (R3),R0	;OFFSET TO THE DATA WORD TO TEST	
2068	035744	056310	000002	BIS 2(R3),(R0)	;SET THE DATA BITS TO BE TESTED	
2069	035750	010465	000000	MOV R4,TSDB(R5)	;SET THE PACKET ADDRESS	
2070	035754	004737	016330	JSR PC,WAITF	;WAIT FOR SSR TO SET	
2071	035760	103405		BCS 15#	;BR IF CARRY SET (GOOD RETURN)	
2072	035762	010001		MOV R0,R1	;SAVE CONTENTS OF TSSR	
2076	035764			ERRDF ERRNO,T9SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET	
	035764	104455			TRAP C#ERDF	
	035766	001612			.WORD 906	
	035770	040067			.WORD T9SSR	
	035772	012116			.WORD PKTSSR	
2077	035774			15#: CKLOOP	;LOOP ON ERROR, IF FLAG SET	
	035774	104406			TRAP C#CLP1	
2078	035776			ESCAPE SEG	;BY-PASS CHECKS IF FATAL ERROR	
	035776	104410			TRAP C#ESCAPE	
	036000	000056			.WORD 10000#-	
2079	036002	005737	002222	TST INTRECV	;DID AN INTERRUPT OCCUR ?	
2080	036006	001004		BNE 22#	;BRANCH IF YES	
2084	036010			ERRHRD ERRNO,T9NINT,PKTSSR		
	036010	104456			TRAP C#ERHRD	
	036012	001613			.WORD 907	
	036014	040156			.WORD T9NINT	
	036016	012116			.WORD PKTSSR	
2085	036020	016501	000002	22#: MOV TSSR(R5),R1	;GET THE CONTENTS OF TSSR	
2086	036024	012702	102206	MOV #SC!SSR!TSREJ!NBA,R2	;EXPECTED CONTENTS OF TSSR	
2087	036030	032701	000100	BIT #OFL,R1	;IS OFF-LINE BIT SET ?	
2088	036034	001402		BEQ 25#	;BRANCH IF NOT OFF-LINE	
2089	036036	052702	000100	BIS #OFL,R2	;SET OFF-LINE IN EXPECTED DATA	
2090	036042	020201		25#: CMP R2,R1	;DOES EXPECTED MATCH RECEIVED ?	

SEQ 0149

```

;LOWER PRIORITY TO ALLOW INTERRUPTS

```


TEST 9: COMPLETION INTERRUPT

2228				;*
2229				;
2230				;TEST 9, SUBTEST 5
2231				;
2232				;CHECK THAT WRITE CHARACTERISTICS COMMAND IS REJECTED IF THE
2233				;MESSAGE BUFFER LENGTH SPECIFIES AN INVALID COUNT (LESS THAN 14)
2234				;
2235				;-
2236				;
2237	036472		BGNSUB	;/!!!!!!!!!!!!!! BEGIN SUBTEST !!!!!!!!!!!!!!!
	036472			T9.5:
	036472	104402		TRAP C#BSUB
2238				
2239	036474		SETPRI #PRI00	;LOWER PRIORITY TO ALLOW INTERRUPTS
	036474	012700		MOV #PRI00,R0
	036500	104441		TRAP C#SPRI
2240	036502	012703	000001	;STARTING BUFFER LENGTH
2241	036506	012704	037350	;GET THE ADDRESS OF COMMAND PACKET
2242	036512	004737	040446	;RESTORE PACKET TO STARTING VALUES
2243				
2244	036516		BGNSEG	;>>>>>>>>> BEGIN SEGMENT >>>>>>>>>
	036516	104404		TRAP C#BSEG
2245				
2246	036520	004737	016054	;DO SOFT INIT OF CONTROLLER
2247	036524	103405		;BR IF SOFT INIT = OK
2251	036526	010001		;SAVE CONTENTS OF TSSR
2252	036530			;DEVICE FATAL ERROR DURING INIT
	036530	104455		TRAP C#ERDF
	036532	001625		.WORD 917
	036534	003652		.WORD SFIERR
	036536	012104		.WORD SFIMSG
2253	036540	005037	002222	;CLEAR INTERRUPT RECEIVED FLAG
2254	036544	010337	037364	;INSERT THE BAD MESSAGE LENGTH
2255	036550	010465	000000	;SET THE PACKET ADDRESS
2256	036554	004737	016330	;WAIT FOR SSR TO SET
2257	036560	103405		;BR IF CARRY SET (GOOD RETURN)
2258	036562	010001		;SAVE CONTENTS OF TSSR
2262	036564			;DEVICE FATAL SSR FAILED TO SET
	036564	104455		TRAP C#ERDF
	036566	001626		.WORD 918
	036570	040067		.WORD T9SSR
	036572	012116		.WORD PKTSSR
2263	036574		15#:	;LOOP ON ERROR, IF FLAG SET
	036574	104406		TRAP C#CLP1
2264	036576			;BY-PASS SUBTEST IF FATAL ERROR
	036576	104410		TRAP C#ESCAPE
	036600	000056		.WORD 10000\$-.
2265	036602	005737	002222	;[ID AN INTERRUPT OCCUR ?
2266	036606	001004		;BRANCH IF YES
2270	036610			
	036610	104456		TRAP C#ERHRD
	036612	001627		.WORD 919
	036614	040156		.WORD T9NINT
	036616	012116		.WORD PKTSSR
2271	036620	016501	000002	;GET THE CONTENTS OF TSSR
2272	036624	012702	102206	;EXPECTED CONTENTS OF TSSR
2273	036630	032701	000100	;IS OFF-LINE BIT SET ?
			22#:	
			MOV TSSR(R5),R1	
			MOV #SC!SSR!TSREJ!NBA,R2	
			BIT #OFL,R1	

L12

TEST 9: COMPLETION INTERRUPT

```

2367
2368
2369      ;TEST WRITE CHARACTERISTICS WITH/WITHOUT INTERRUPTS ENABLED
2370      ;
2371      ;-
2372      037130      BGNSUB      ;//////////////// BEGIN SUBTEST //////////////////
      037130      T9.7:      TRAP      C#BSUB
      037130      104402
2373
2374      037132      SETPRI      #PRI00      ;LOWER PRIORITY TO ALLOW INTERRUPTS
      037132      012700      000000      MOV      #PRI00,R0
      037136      104441      TRAP      C#SPRI
2375      037140      012704      037350      MOV      #T9PACKET,R4      ;GET THE ADDRESS OF COMMAND PACKET
2376      037144      004737      040446      JSR      PC,T9REST      ;SET UP A VALID PACKET
2377      037150      004737      016054      JSR      PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
2378      037154      103405      BCS      104      ;BR IF SOFT INIT = OK
2382      037156      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2383      037160      ERRDF      ERRNO,SFIERR,SFIMSG      ;DEVICE FATAL ERROR DURING INIT
      037160      104455      TRAP      C#ERDF
      037162      001635      .WORD      925
      037164      003652      .WORD      SFIERR
      037166      012104      .WORD      SFIMSG
2384      037170      005037      002222      104:      CLR      INTRECV      ;CLEAR INTERRUPT RECEIVED FLAG
2385      037174      052714      000200      BIS      #BIT7,(R4)      ;ENABLE INTERRUPTS
2386      037200      010465      000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS
2387      037204      004737      016416      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
2388      037210      103405      BCS      154      ;BR IF CARRY SET (GOOD RETURN)
2389      037212      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2393      037214      ERRDF      ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      037214      104455      TRAP      C#ERDF
      037216      001636      .WORD      926
      037220      040067      .WORD      T9SSR
      037222      012116      .WORD      PKTSSR
2394      037224      104406      154:      CKLOOP      ;LOOP ON ERROR, IF FLAG SET
      037224      104406      ESCAPE      SUB      ;BY-PASS SUBTEST IF FATAL ERROR
2395      037226      104410      TRAP      C#CLP1
      037226      000102      TRAP      C#ESCAPE
      037230      005737      002222      TST      INTRECV      ;DID AN INTERRUPT OCCUR ?
2396      037232      001004      BNE      224      ;BRANCH IF YES
2397      037236      001004      ERRHRD      ERRNO,T9NINT,PKTSSR
2401      037240      104456      TRAP      C#ERHRD
      037240      001637      .WORD      927
      037244      040156      .WORD      T9NINT
      037246      012116      .WORD      PKTSSR
2402      037250      104406      224:      CKLOOP      ;LOOP ON ERROR ?
      037250      104406      TRAP      C#CLP1
2403
2404      037252      005037      002222      CLR      INTRECV      ;CLEAR INTERRUPT RECEIVED FLAG
2405      037256      042714      000200      BIC      #BIT7,(R4)      ;DISABLE INTERRUPTS
2406      037262      010465      000000      MOV      R4,TSDB(R5)      ;SET THE PACKET ADDRESS
2407      037266      004737      016416      JSR      PC,CHKTSSR      ;WAIT FOR SSR TO SET
2408      037272      103405      BCS      254      ;BR IF CARRY SET (GOOD RETURN)
2409      037274      010001      MOV      R0,R1      ;SAVE CONTENTS OF TSSR
2413      037276      ERRDF      ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      037276      104455      TRAP      C#ERDF
      037300      001640      .WORD      928

```


M12

TSVSA - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-49

SEQ 0155

TEST 9: COMPLETION INTERRUPT

```

037302 040067
037304 012116
2414 037306 104406      254:  CKLOOP      ;LOOP ON ERROR, IF FLAG
037306 104406      ;BY-PASS SUBTEST IF FATAL ERROR
2415 037310 104410      ESCAPE SUB      TRAP C#CLP1
037310 104410      ;DID AN INTERRUPT OCCUR ?
037312 000020      TST INTRECV      TRAP C#ESCAPE
2416 037314 005737 002222 BEQ 304:      ;BRANCH IF NOT
2417 037320 001404      ERRHRD ERRNO,T9INT,PKTSSR
2421 037322      ;
037322 104456      TRAP C#ERHRD
037324 001641      .WORD 929
037326 040247      .WORD T9INT
037330 012116      .WORD PKTSSR
2422 037332      ;
2423 037332      304:  ENDSUB      ;////////////////// END SUBTEST ////////////////////
037332 104403      L10074:      TRAP C#ESUB
2424 037334      EXIT TST      ;ALL DONE THIS TEST
2425 037334 104432      TRAP C#EXIT
037336 001162      .WORD L10065-.

;*
;LOCAL STORAGE FOR THIS TEST
;-
2432 037340      T9PACKET: .BLKB 10-<.-TSV2&7>
2434 037350      .WORD 100204      ;COMMAND PACKET FOR TEST
2435 037350 100204      .WORD T9DATA      ;WRITE CHAR COMMAND, WITH IE, ACK
2436 037352 037360      .WORD 0      ;ADDRESS OF CHARACTERISTICS BLOCK
2437 037354 000000      .WORD 8.      ;STARTING VALUE OF BLOCK SIZE
2438 037356 000010      ;
2439      ;
2440 037360      T9DATA: .WORD T9BFR      ;CHARACTERISTICS DATA BLOCK
2441 037360 037372      .WORD 0      ;ADDRESS OF MESSAGE BUFFER
2442 037362 000000      .WORD 14.      ;LENGTH OF MESSAGE BUFFER
2443 037364 000016      .WORD 0,0
2444 037366 000000 000000      ;
2445      ;
2446 037372      T9BFR: .BLKW 8.      ;MESSAGE BUFFER
2447      ;
2448      ;*
2449      ;TEST DATA FOR SUBTEST TWO
2450      ;
2451      ;DATA HAS FORMAT:
2452      ;
2453      ;      1ST WORD      OFFSET TO TEST WORD IN PACKET
2454      ;      2ND WORD      BITS TO SET FOR TEST
2455      ;
2456      ;
2457      ;-
2458      ;
2459 037412      T92DATA: .WORD 0,BIT5!BIT6!BIT9!BIT10!BIT11!BIT12!BIT13
2460 037412 000000 037140      .WORD 2,BIT0
2461 037416 000002 000001      .WORD 4,BIT6!BIT15
2462 037422 000004 100100

```

TEST 9: COMPLETION INTERRUPT

```

2463          037426          T92DONE=.
2464
2465
2466          ;*
2467          ;LOCAL TEXT MESSAGES FOR TEST
2468          ;*
2469
2470 037426      127      122      111 T9NBA: .ASCIZ 'WRITE CHARACTERISTICS Command Not Accepted'
2471 037501      127      122      111 T92REJ: .ASCIZ 'WRITE CHARACTERISTICS Not Rejected With Non-Zero Unused Fields'
2472 037600      127      122      111 T93REJ: .ASCIZ 'WRITE CHARACTERISTICS Not Rejected With Invalid Data Count'
2473 037673      127      122      111 T94REJ: .ASCIZ 'WRITE CHARACTERISTICS Not Rejected With Invalid Block Address'
2474 037771      127      122      111 T95REJ: .ASCIZ 'WRITE CHARACTERISTICS Not Rejected With Invalid Buffer Length'
2475 040067      103      157      156 T9SSR: .ASCIZ 'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
2476 040156      105      170      160 T9NINT: .ASCIZ 'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
2477 040247      125      156      145 T9INT: .ASCIZ 'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
2478 040336      111      156      143 T9TSBA: .ASCIZ 'Incorrect TSBA Address After WRITE CHARACTERISTICS'
2479 040421      103      157      155 TST9ID: .ASCIZ 'Completion Interrupt'
2480          .EVEN
2481
2482
2483          ;*
2484          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
2485          ;*
2486          ;*
2487          ;*
2488
2489 040446          T9REST:
2490 040446          SAVREG          ;SAVE THE REGISTERS
2491 040452      012701      037350      MOV      #T9PACKET,R1      ;START OF THE PACKET
2492 040456      012721      100204      MOV      #100204,(R1).      ;WRITE CHARACTERISTICS WITH ACK, IE
2493 040462      012721      037360      MOV      #T9DATA,(R1).      ;ADDRESS OF CHAR DATA BLOCK
2494 040466      005021          CLR      (R1).      ;EXTENDED ADDRESS
2495 040470      012721      000010      MOV      #8,(R1).      ;SIZE OF DATA BLOCK IN BYTES
2496 040474      012721      037372      MOV      #T9BFR,(R1).      ;ADDRESS OF MESSAGE BUFFER
2497 040500      005021          CLR      (R1).
2498 040502      012721      000016      MOV      #14,(R1).      ;LENGTH OF MESSAGE BUFFER
2499 040506      005021          CLR      (R1).
2500 040510      005011          CLR      (R1)
2501 040512      005037      037372      CLR      T9BFR          ;CLEAR 1ST LOC IN MESSAGE BUFFER
2502 040516      000207          RTS      PC          ;RETURN
2503 040520          ENDTST
2504          040520      104401          L10065:      TRAP      C#ETST
2505
2506          .SBTTL TEST 10: BASIC PACKET PROTOCOL
2507
2508          ; THIS TEST VERIFIES BASIC OPERATION OF THE MESSAGE BUFFER RELEASE
2509          ; COMMAND, THE FUNCTION OF THE ACK BIT IN THE COMMAND HEADER WORD,
2510          ; AND THE REGISTER MODIFICATION REFUSED (RMR) LOGIC.
2511
2512          ;*
2513          ;TEST 10 SUBTEST 1
2514          ;*
2515          ;CHECKS THAT THE MESSAGE BUFFER RELEASE COMMAND WORKS
2516          ;PROPERLY AND THAT NO INTERRUPT IS GENERATED EVEN
2517          ;IF THE "IE" BIT IS SET IN THE COMMAND PACKET

```


TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-52

SEQ 0158

TEST 10: BASIC PACKET PROTOCOL

Address	Offset	Hex	Dec	Label	Instruction	Comment	Trap	Word
2567	040704	032701	000100		BIT	#OFL,R1		
2568	040710	001402			BEQ	25\$		
2569	040712	052702	000100		BIS	#OFL,R2		
2570	040716	020201		25\$:	CMP	R2,R1		
2571	040720	001404			BEQ	30\$		
2575	040722				ERRHRD	ERRNO,T10NBA,PKTSSR		
	040722	104456					TRAP	C#ERHRD
	040724	001754					.WORD	1004
	040726	043051					.WORD	T10NBA
	040730	012116					.WORD	PKTSSR
2576	040732			30\$:				
2577	040732				ENDSEG			
	040732							
	040732	104405						
2578	040734				BGNSEG			
	040734	104404						
2579	040736	005037	002222		CLR	INTRECV		
2580	040742	012737	025252	042672	MOV	#025252,T10BFR		
2581	040742	012737	025252		MOV	#100212,(R4)		
2582	040750	012714	100212		MOV	R4,TSDB(R5)		
2583	040754	010465	000000		MOV	PC,CHKTSSR		
2584	040760	004737	016416		JSR	PC,CHKTSSR		
2585	040764	103407			BCS	45\$		
2586	040766	010001			MOV	R0,R1		
2590	040770				ERRDF	ERRNO,T10SSR,PKTSSR		
	040770	104455					TRAP	C#ERDF
	040772	001755					.WORD	1005
	040774	043210					.WORD	T10SSR
	040776	012116					.WORD	PKTSSR
2591	041000	005237	002220		INC	FATFLG		
2592	041004			45\$:	CKLOOP			
	041004	104406						
2593	041006	005737	002222		TST	INTRECV		
2594	041012	001404			BEQ	52\$		
2598	041014				ERRHRD	ERRNO,T10INT,PKTSSR		
	041014	104456					TRAP	C#ERHRD
	041016	001756					.WORD	1006
	041020	043370					.WORD	T10INT
	041022	012116					.WORD	PKTSSR
2599	041024	016501	000002	52\$:	MOV	TSSR(R5),R1		
2600	041030	012702	000200		MOV	#SSR,R2		
2601	041034	032701	000100		BIT	#OFL,R1		
2602	041040	001402			BEQ	55\$		
2603	041042	052702	000100		BIS	#OFL,R2		
2604	041046	020201		55\$:	CMP	R2,R1		
2605	041050	001404			BEQ	60\$		
2609	041052				ERRHRD	ERRNO,T10NNBA,PKTSSR		
	041052	104456					TRAP	C#ERHRD
	041054	001757					.WORD	1007
	041056	043133					.WORD	T10NNBA
	041060	012116					.WORD	PKTSSR
2610	041062			60\$:				</

TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-53

SEQ 0159

TEST 10: BASIC PACKET PROTOCOL

Address	Offset	Value	Label	Operation	Comments
041076	104456				
041100	001760				
041102	042754				
041104	015554				
2619	041106		704:	TST	FATFLG
2620	041106	005737		BEG	804
2621	041112	001403		JSR	PC,CKDROP
2622	041114	004737	017262	ENDSEG	
2623	041120				
	041120				
	041120	104405			
2624	041122		804:		
2625	041122			ENDSUB	
	041122				
	041122	104403			
2626					
2627					
2628					
2629					
2630					
2631					
2632					
2633					
2634					
2635					
2636	041124			BGNSUB	
	041124				
	041124	104402			
2637					
2638	041126	004737	043506	JSR	PC,T10RST
2639	041132			SETPRI	#PRI00
	041132	012700	000000		
	041136	104441			
2640	041140	012704	042650	MOV	#T10PACKET,R4
2641	041144	012764	000010	MOV	#8.,PKBCNT(R4)
2642	041152		000006		
2643	041152			54:	
	041152	104404		BGNSEG	
2644					
2645	041154	004737	016054	JSR	PC,SOFINIT
2646	041160	103405		BCS	104
2650	041162	010001		MOV	R0,R1
2651	041164			ERRDF	ERRNO,SFIERR,SFIMSG
	041164	104455			
	041166	001761			
	041170	003652			
	041172	012104			
2652	041174	005037	002220	104:	
2653	041200	005037	002222	CLR	FATFLG
2654	041204	012737	000020	CLR	INTRECV
2655	041212	010465	000000	MOV	#000020,T10DATA+6
2656	041216	004737	016416	MOV	R4,TSDB(R5)
2657	041222	103407		JSR	PC,CHKTSSR
2658	041224	010001		BCS	154
2662	041226			MOV	R0,R1
	041226	104455		ERRDF	ERRNO,T10SSR,PKTSSR
	041230	001762			

E13
TSVSA - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-54

TEST 10: BASIC PACKET PROTOCOL

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

F13

SEQ 0161

TEST 10: BASIC PACKET PROTOCOL

2707	041422	012702	000200		MOV	#SSR,R2	;EXPECTED CONTENTS OF TSSR
2708	041426	032701	000100		BIT	#OFL,R1	;IS OFF-LINE BIT SET ?
2709	041432	001402			BEQ	55:	;BRANCH IF NOT OFF-LINE
2710	041434	052702	000100		BIS	#OFL,R2	;SET OFF-LINE IN EXPECTED DATA
2711	041440	020201			CMP	R2,R1	;DOES EXPECTED MATCH RECEIVED ?
2712	041442	001404			BEQ	60:	;OKAY IF MATCH
2716	041444				ERRHRD	ERRNO,T10NNBA,PKTSSR	;NBA NOT SET
	041444	104456					TRAP C#ERHRD
	041446	001767					.WORD 1015
	041450	043133					.WORD T10NNBA
	041452	012116					.WORD PKTSSR
2717	041454						
2718	041454	013701	042672		MOV	T10BFR,R1	;PICK UP THE 1ST WORD OF MESSAGE BUFFER
2719	041460	012702	025252		MOV	#025252,R2	;SET UP EXPECTED DATA
2720	041464	020102			CMP	R1,R2	;WAS ANY MESSAGE REC'D
2721	041466	001404			BEQ	70:	;BR, IF OK (EQUAL)
2725	041470				ERRHRD	ERRNO,T10MBF,EXPREC	;MESSAGE BUFFER WAS MODIFIED
	041470	104456					TRAP C#ERHRD
	041472	001770					.WORD 1016
	041474	042754					.WORD T10MBF
	041476	015554					.WORD EXPREC
2726							
2727	041500						
2728	041500	005737	002220		TST	FATFLG	;ANY FATAL ERRORS
2729	041504	001402			BEQ	80:	;BR, IF NO FATAL ERRORS
2730	041506	004737	017262		JSR	PC,CKDROP	;TRY TO DROP THE UNIT
2731	041512						
2732	041512				ENDSEG		;<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<<<<
	041512						100014: TRAP C#ESEG
2733	041514				ENDSUB		;////////// END SUBTEST //////////
	041514						L10077: TRAP C#ESUB
	041514	104403					
2734							
2735							
2736							
2737							
2738							
2739							
2740							
2741							
2742							
2743	041516				BGN SUB		;////////// BEGIN SUBTEST //////////
	041516						T10.3: TRAP C#BSUB
	041516	104402					
2744							
2745	041520	004737	043506		JSR	PC,T10RST	;SET PACKET TO INITIAL VALUES
2746	041524				SETPRI	#PRI00	;LOWER PRIORITY TO ALLOW INTERRUPTS
	041524	012700	000000				MOV #PRI00,R0
	041530	104441					TRAP C#SPRI
2747	041532	012704	042650		MOV	#T10PACKET,R4	;GET THE ADDRESS OF COMMAND PACKET
2748	041536	012764	000010	000006	MOV	#8.,PKBCNT(R4)	;START WITH MINIMUM ALLOWABLE VALUE
2749	041544						
2750	041544				BGN SEG		;>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>>>
	041544	104404					TRAP C#BSEG
2751							
2752	041546	004737	016054		JSR	PC,SOFINIT	;DO SOFT INIT OF CONTROLLER

TSVSA - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-56

SEQ 0162

TEST 10: BASIC PACKET PROTOCOL

Address	Offset	Hex	Label	Operation	Comments	Trap	Offset
2753	041552	103405					
2757	041554	010001		BCS	104		
2758	041556			MOV	R0,R1		
	041556	104455		ERRDF	ERRNO,SFIERR,SFIMSG		
	041560	001771				TRAP	C4ERDF
	041562	003652				.WORD	1017
	041564	012104				.WORD	SFIERR
2759	041566	005037	002220	104:	CLR	FATFLG	
2760	041572	005037	002222		CLR	INTRECV	
2761	041576	010465	000000		MOV	R4,TSDB(R5)	
2762	041602	004737	016416		JSR	PC,CHKTSSR	
2763	041606	103407			BCS	154	
2764	041610	010001			MOV	R0,R1	
2768	041612				ERRDF	ERRNO,T10SSR,PKTSSR	
	041612	104455				TRAP	C4ERDF
	041614	001772				.WORD	1018
	041616	043210				.WORD	T10SSR
	041620	012116				.WORD	PKTSSR
2769	041622	005237	002220	154:	INC	FATFLG	
2770	041626				CKLOOP		
2771	041630	104406			ESCAPE	SEG	
	041630	104410					
	041632	000056					
2772	041634	005737	002222		TST	INTRECV	
2773	041640	001004			BNE	224	
2777	041642				ERRHRD	ERRNO,T10NINT,PKTSSR	
	041642	104456					
	041644	001773				TRAP	C4ERHRD
	041646	043277				.WORD	1019
	041650	012116				.WORD	T10NINT
2778	041652	016501	000002	224:	MOV	TSSR(R5),R1	
2779	041656	012702	000200		MOV	#SSR,R2	
2780	041662	032701	000100		BIT	#OFL,R1	
2781	041666	001402			BEQ	254	
2782	041670	052702	000100	254:	BIS	#OFL,R2	
2783	041674	020201			CMP	R2,R1	
2784	041676	001404			BEQ	304	
2788	041700				ERRHRD	ERRNO,T10NBA,PKTSSR	
	041700	104456					
	041702	001774				TRAP	C4ERHRD
	041704	043051				.WORD	1020
	041706	012116				.WORD	T10NBA
2789	041710			304:			PKTSSR
2790	041710				ENDSEG		
	041710	104405					
2791	041712				BGNSEG		
	041712	104404					
2792	041714	004737	043506		JSR	PC,T10RST	
2793	041720	005037	002222		CLR	INTRECV	
2794	041724	012737	025252	042672	MOV	#025252,T10BFR	
2795	041732	012714	100212		MOV	#100212,(R4)	
2796	041736	010465	000000		MOV	R4,TSDB(R5)	
2797	041742	004737	016416		JSR	PC,CHKTSSR	
2798	041746	103407			BCS	454	
2799	041750	010001			MOV	R0,R1	

H13

TSVSA - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-57

SEQ 0163

TEST 10: BASIC PACKET PROTOCOL

2803	041752			ERRDF	ERRNO,T10SSR,PKTSSR	;DEVICE FATAL SSR FAILED TO SET		
	041752	104455					TRAP	C#ERDF
	041754	001775					.WORD	1021
	041756	043210					.WORD	T10SSR
	041760	012116					.WORD	PKTSSR
2804	041762	005237	002220	45#:	INC	FATFLG		
2805	041766				CKLOOP			
	041766	104406				;SET FATAL ERROR FLAG		
2806	041770	005737	002222		TST	INTRECV		
2807	041774	001404			BEQ	52#		
2811	041776				ERRHRD	ERRNO,T10INT,PKTSSR		
	041776	104456					TRAP	C#ERHRD
	042000	001776					.WORD	1022
	042002	043370					.WORD	T10INT
	042004	012116					.WORD	PKTSSR
2812	042006	016501	000002	52#:	MOV	TSSR(R5),R1		
2813	042012	012702	000200		MOV	#SSR,R2		
2814	042016	032701	000100		BIT	#OFL,R1		
2815	042022	001402			BEQ	55#		
2816	042024	052702	000100		BIS	#OFL,R2		
2817	042030	020201		55#:	CMP	R2,R1		
2818	042032	001404			BEQ	60#		
2822	042034				ERRHRD	ERRNO,T10NNBA,PKTSSR		
	042034	104456					TRAP	C#ERHRD
	042036	001777					.WORD	1023
	042040	043133					.WORD	T10NNBA
	042042	012116					.WORD	PKTSSR
2823	042044			60#:				
2824	042044	013701	042672		MOV	T10BFR,R1		
2825	042050	012702	025252		MOV	#025252,R2		
2826	042054	020102			CMP	R1,R2		
2827	042056	001404			BEQ	70#		
2831	042060				ERRHRD	ERRNO,T10MBF,EXPREC		
	042060	104456					TRAP	C#ERHRD
	042062	002000					.WORD	1024
	042064	042754					.WORD	T10MBF
	042066	015554					.WORD	EXPREC
2832	042070			70#:				
2833	042070	104406			CKLOOP			
						;LOOP ON ERROR IF FLAG SET		
2834							TRAP	C#CLP1
2835	042072	005037	002222		CLR	INTRECV		
2836	042076	004737	043506		JSR	PC,T10RST		
2837	042102	042714	100000		BIC	#100000,(R4)		
2838	042106	010465	000000		MOV	R4,TSDB(R5)		
2839	042112	004737	016416		JSR	PC,CHKTSSR		
2840	042116	103407			BCS	75#		
2841	042120	010001			MOV	R0,R1		
2845	042122				ERRDF	ERRNO,T10SSR,PKTSSR		
	042122	104455					TRAP	C#ERDF
	042124	002001					.WORD	1025
	042126	043210					.WORD	T10SSR
	042130	012116					.WORD	PKTSSR
2846	042132	005237	002220	75#:	INC	FATFLG		
2847	042136				CKLOOP			
	042136	104406				;SET FATAL ERROR FLAG		
2848	042140				ESCAPE	SEG		
						;LOOP ON ERROR, IF FLAG SET		
							TRAP	C#CLP1
						;BY-PASS SUBTEST IF FATAL ERROR		

TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-59

SEQ 0165

TEST 10: BASIC PACKET PROTOCOL

2894	042332	016501	000002		112\$:	MOV	TSSR(R5),R1	;GET THE CONTENTS OF TSSR		
2895	042336	012702	000200			MOV	#SSR,R2	;EXPECTED CONTENTS OF TSSR		
2896	042342	032701	000100			BIT	#OFL,R1	;IS OFF-LINE BIT SET ?		
2897	042346	001402				BEQ	125\$	;BRANCH IF NOT OFF-LINE		
2898	042350	052702	000100			BIS	#OFL,R2	;SET OFF-LINE IN EXPECTED DATA		
2899	042354	020201			125\$:	CMP	R2,R1	;DOES EXPECTED MATCH RECEIVED ?		
2900	042356	001405				BEQ	130\$	;OKAY IF MATCH		
2904	042360					ERRHRD	ERRNO,T1OSSR,PKTSSR	;NBA NOT ZERO		
	042360	104456							TRAP	C#ERHRD
	042362	002006							.WORD	1030
	042364	043210							.WORD	T1OSSR
	042366	012116							.WORD	PKTSSR
2905	042370					ENDSEG		;<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<		
	042370							10002\$: TRAP C#ESEG		
2906	042372	005737	002220		130\$:	TST	FATFLG	;CHECK FATAL FLAG FOR SET		
2907	042376	001402				BEQ	140\$	;BR, IF NOT SET (OK)		
2908	042400	004737	017262			JSR	PC,CKDROP	;DROP DEVICE IF INDICATED		
2909	042404				140\$:					
2910	042404					ENDSUB		;////////// END SUBTEST //////////		
	042404							L10100: TRAP C#ESUB		
2911	042406	104403			30\$:					
2912										
2913										
2914										
2915										
2916										
2917										
2918										
2919										
2920										
2921										
2922	042406					BGN SUB		;////////// BEGIN SUBTEST //////////		
	042406							T10.4: TRAP C#BSUB		
2923	042406	104402								
2924	042410	004737	043560			JSR	PC,T1ORT2	;SET SECOND PACKET UP		
2925	042414	004737	043506			JSR	PC,T1ORST	;SET PACKET TO INITIAL VALUES		
2926	042420					SETPRI	#PRI00	;LOWER PRIORITY TO ALLOW INTERRUPTS		
	042420	012700	000000					MOV #PRI00,R0		
	042424	104441						TRAP C#SPRI		
2927	042426	012704	042650			MOV	#T10PACKET,R4	;GET THE ADDRESS OF COMMAND PACKET		
2928	042432	012703	042712			MOV	#T10PKT,R3	;GET THE ADDRESS OF 2ND CMD PACKET		
2929	042436	012764	000010	000006		MOV	#8.,PKBCNT(R4)	;START WITH MINIMUM ALLOWABLE VALUE		
2930	042444	012763	000010	000006		MOV	#8.,PKBCNT(R3)	;START WITH MINIMUM ALLOWABLE VALUE		
2931	042452				5\$:					
2932	042452					BGN SEG		;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>		
	042452	104404						TRAP C#BSEG		
2933	042454	004737	016054			JSR	PC,SOFINIT	;DO SOFT INIT OF CONTROLLER		
2934	042460	103405				BCS	10\$	;BR IF SOFT INIT = OK		
2938	042462	010001				MOV	R0,R1	;SAVE CONTENTS OF TSSR		
2939	042464					ERRDF	ERRNO,SFIERR,SFIMSG	;DEVICE FATAL ERROR DURING INIT		
	042464	104455						TRAP C#ERDF		
	042466	002007						.WORD	1031	
	042470	003652						.WORD	SFIERR	
	042472	012104								

TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-60

SEQ 0166

TEST 10: BASIC PACKET PROTOCOL

Address	Offset	Hex	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	
---------	--------	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

L13

TEST 10: BASIC PACKET PROTOCOL

```

2987 042650 100204          .WORD 100204          ;WRITE CHAR COMMAND, WITH IE, ACK
2988 042652 042660          .WORD T10DATA         ;ADDRESS OF CHARACTERISTICS BLOCK
2989 042654 000000          .WORD 0              ;
2990 042656 000010          .WORD 8              ;STARTING VALUE OF BLOCK SIZE
2991
2992 042660          T10DATA:          ;CHARACTERISTICS DATA BLOCK
2993 042660 042672          .WORD T10BFR         ;ADDRESS OF MESSAGE BUFFER
2994 042662 000000          .WORD 0              ;
2995 042664 000016          .WORD 14             ;LENGTH OF MESSAGE BUFFER
2996 042666 000000 000000    .WORD 0,0           ;
2997
2998 042672          T10BFR: .BLKW 8             ;MESSAGE BUFFER
2999
3000          ;*
3001          ;
3002          ;TEST DATA FOR SUBTEST FOUR
3003          ;
3004 042712          T10PKT:          ;COMMAND PACKET FOR TEST
3005 042712 100204          .WORD 100204         ;WRITE CHAR COMMAND, WITH IE, ACK
3006 042714 042722          .WORD T10DTA         ;ADDRESS OF CHARACTERISTICS BLOCK
3007 042716 000000          .WORD 0              ;
3008 042720 000010          .WORD 8              ;STARTING VALUE OF BLOCK SIZE
3009
3010 042722          T10DTA:          ;CHARACTERISTICS DATA BLOCK
3011 042722 042734          .WORD T10BUFR        ;ADDRESS OF MESSAGE BUFFER
3012 042724 000000          .WORD 0              ;
3013 042726 000016          .WORD 14             ;LENGTH OF MESSAGE BUFFER
3014 042730 000000 000000    .WORD 0,0           ;
3015
3016 042734          T10BUFR: .BLKW 8             ;MESSAGE BUFFER
3017
3018          ;*
3019          ;LOCAL TEXT MESSAGES FOR TEST
3020          ;-
3021
3022
3023 042754 115 145 163 T10MBF: .ASCIZ 'Message Buffer Modified after MESSAGE BUFFER RELEASE Command'
3024 043051 116 102 101 T10NBA: .ASCIZ 'NBA Not Clear After WRITE CHARACTERISTICS Command'
3025 043133 116 102 101 T10NNBA: .ASCIZ 'NBA Set After MESSAGE BUFFER RELEASE Command'
3026 043210 103 157 156 T10SSR: .ASCIZ 'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
3027 043277 105 170 160 T10NINT: .ASCIZ 'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
3028 043370 125 156 145 T10INT: .ASCIZ 'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
3029 043457 102 141 163 TST10ID: .ASCIZ 'Basic Packet Protocol'
3030          .EVEN
3031
3032
3033          ;*
3034          ;
3035          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
3036          ;
3037          ;-
3038
3039
3040 043506          T10RST:          ;SAVE THE REGISTERS
3041 043506          SAVREG          ;START OF THE PACKET
3042 043512 012701 042650      MOV #T10PACKET,R1 ;WRITE CHARACTERISTICS WITH ACK, IE
3043 043516 012721 100204      MOV #100204,(R1)

```

M13
TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-62

TEST 10: BASIC PACKET PROTOCOL

Address	Offset	PC	Instruction	Comment
3044	043522	012721	042660	MOV #T10DATA,(R1). ;ADDRESS OF CHAR DATA BLOCK
3045	043526	005021		CLR (R1). ;EXTENDED ADDRESS
3046	043530	012721	000010	MOV #8.,(R1). ;SIZE OF DATA BLOCK IN BYTES
3047	043534	012721	042672	MOV #T10BFR,(R1). ;ADDRESS OF MESSAGE BUFFER
3048	043540	005021		CLR (R1).
3049	043542	012721	000016	MOV #14.,(R1). ;LENGTH OF MESSAGE BUFFER
3050	043546	005021		CLR (R1).
3051	043550	005011		CLR (R1).
3052	043552	005037	042672	CLR T10BFR ;CLEAR 1ST LOC IN MESSAGE BUFFER
3053	043556	000207		RTS PC ;RETURN
3054				
3055				
3056				
3057				
3058				
3059				
3060	043560			T10RT2:
3061	043560			SAVREG ;SAVE THE REGISTERS
3062	043564	012701	042712	MOV #T10PKT,R1 ;START OF THE PACKET
3063	043570	012721	100204	MOV #100204,(R1). ;WRITE CHARACTERISTICS WITH ACK, IE
3064	043574	012721	042722	MOV #T10DTA,(R1). ;ADDRESS OF CHAR DATA BLOCK
3065	043600	005021		CLR (R1). ;EXTENDED ADDRESS
3066	043602	012721	000010	MOV #8.,(R1). ;SIZE OF DATA BLOCK IN BYTES
3067	043606	012721	042734	MOV #T10BUFR,(R1). ;ADDRESS OF MESSAGE BUFFER
3068	043612	005021		CLR (R1).
3069	043614	012721	000016	MOV #14.,(R1). ;LENGTH OF MESSAGE BUFFER
3070	043620	005021		CLR (R1).
3071	043622	005011		CLR (R1).
3072	043624	005037	042734	CLR T10BUFR ;CLEAR 1ST LOC IN MESSAGE BUFFER
3073	043630	000207		RTS PC ;RETURN
3074	043632			ENDTST
3075	043632			
3076	043632	104401		
3077				
3078				
3079				
3080				
3081				
3082				
3083				
3084				
3085				
3086				
3087				
3088	043634			BGNTST
3089	043634			
3090	043634	012700	045642	
3091	043640	004737	016570	
3092	043644	012737	000024	002214
3093	043652			
3094	043652			
3095	043652			
3096	043652			
3097	043652			
3098	043652	104402		
3099	043654			

N13

SEQ 0169

TEST 11: NON-TAPE MOTION COMMANDS

PC	Address	Offset	Label	Instruction	Comment	Trap	Priority
	043654	012700	000000				#P100,RO
	043660	104441				TRAP	C4SPRI
3100	043662	004737	016054	JSR	PC,SOFINIT		
3101	043666	103405		BCS	34		
3105	043670	010001		MOV	RO,R1		
3106	043672			ERRDF	ERRNO,SFIERR,SFIMSG		
	043672	104455				TRAP	C4ERDF
	043674	002115				.WORD	1101
	043676	003652				.WORD	SFIERR
	043700	012104				.WORD	SFIMSG
3107	043702						
3108	043702	012704	045070	MOV	#T11PK2,R4		
3109	043706	004737	010662	JSR	PC,WRTCHR		
3110	043712	103404		BCS	44		
3114	043714			ERRHRD	ERRNO,WRTMSG,SFIMSG		
	043714	104456				TRAP	C4ERHRD
	043716	002116				.WORD	1102
	043720	005056				.WORD	WRTMSG
	043722	012104				.WORD	SFIMSG
3115	043724						
3116	043724	004737	045674	JSR	PC,T11REST		
3117	043730	012704	045020	MOV	#T11PACKET,R4		
3118	043734						
3119	043734			BGNSEG			
	043734	104404				TRAP	C4BSEG
3120							
3121	043736	005037	002220	CLR	FATFLG		
3122	043742	005037	002222	CLR	INTRECV		
3123	043746	010465	000000	MOV	R4,TSDB(R5)		
3124	043752	004737	016416	JSR	PC,CHKTSSR		
3125	043756	103407		BCS	154		
3126	043760	010001		MOV	RO,R1		
3130	043762			ERRDF	ERRNO,T11SSR,PKTSSR		
	043762	104455				TRAP	C4ERDF
	043764	002117				.WORD	1103
	043766	045364				.WORD	T11SSR
	043770	012116				.WORD	PKTSSR
3131	043772	005237	002220	INC	FATFLG		
3132	043776			CKLOOP			
	043776	104406				TRAP	C4CLP1
3133	044000			ESCAPE	SEG		
	044000	104410				TRAP	C4ESCAPE
	044002	000074				.WORD	100004-
3134	044004	005737	002222	TST	INTRECV		
3135	044010	001004		BNE	224		
3139	044012			ERRHRD	ERRNO,T11NINT,PKTSSR		
	044012	104456				TRAP	C4ERHRD
	044014	002120				.WORD	1104
	044016	045514				.WORD	T11NINT
	044020	012116				.WORD	PKTSSR
3140	044022	016501	000002	MOV	TSSR(R5),R1		
3141	044026	012702	000200	MOV	#SSR,R2		
3142	044032	032701	000100	BIT	#OFL,R1		
3143	044036	001402		BEQ	254		
3144	044040	052702	000100	BIS	#OFL,R2		
3145	044044	020201		CMP	R2,R1		
3146							

C14
TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-65

TEST 11: NON-TAPE MOTION COMMANDS

Address	Offset	Hex	Label	Instruction	Comment	Trap	Code
3192	044152	004737	010662				
3193	044156	103404					
3197	044160			JSR BCS ERRHRD	PC, WRTCHR 4#		
	044160	104456			ERRNO, WRTMSG, SFMSG		
	044162	002124				TRAP	C#ERRHRD
	044164	005056				.WORD	1108
	044166	012104				.WORD	WRTMSG
						.WORD	SFMSG
3198	044170			4#:			
3199	044170	004737	045674	JSR	PC, T11REST		
3200	044174	012704	045020	MOV	#T11PACKET, R4		
3201	044200						
3202	044200	005037	002222	5#:			
3203	044204	052714	007400	10#:	CLR INTRECV		
3204	044210	010465	000000	BIS	#P.MODE, (R4)		
3205	044214	004737	016416	MOV	R4, TSDB(R5)		
3206	044220	103405		JSR	PC, CHKTSSR		
3207	044222	010001		BCS	15#		
3211	044224			MOV	R0, R1		
	044224	104455		ERRDF	ERRNO, T11SSR, PKTSSR		
	044226	002125				TRAP	C#ERDF
	044230	045364				.WORD	1109
	044232	012116				.WORD	T11SSR
3212	044234			15#:	CKLOOP		
3213	044236	104406		ESCAPE	SEG		
	044236	104410				TRAP	C#CLP1
	044240	000074				TRAP	C#ESCAPE
3214	044242	005737	002222	TST	INTRECV		
3215	044246	001004		BNE	22#		
3219	044250			ERRHRD	ERRNO, T11NINT, PKTSSR		
	044250	104456				TRAP	C#ERRHRD
	044252	002126				.WORD	1110
	044254	045514				.WORD	T11NINT
	044256	012116				.WORD	PKTSSR
3220	044260	016501	000002	22#:	MOV TSSR(R5), R1		
3221	044264	012702	100206	MOV	#SC!SSR!TSREJ, R2		
3222	044270	032701	000100	BIT	#OFL, R1		
3223	044274	001402		BEQ	25#		
3224	044276	052702	000100	BIS	#OFL, R2		
3225	044302	020201		25#:	CMP R2, R1		
3226	044304	001404		BEQ	30#		
3230	044306			ERRHRD	ERRNO, T112REJ, PKTSSR		
	044306	104456				TRAP	C#ERRHRD
	044310	002127				.WORD	1111
	044312	045172				.WORD	T112REJ
	044314	012116				.WORD	PKTSSR
3231	044316			30#:			
3232	044316	004737	011154	35#:	JSR PC, CKRAM		
3233	044322	103405		BCS	59#		
3237	044324			ERRHRD	ERRNO, PKTRAM, RAMERR		
	044324	104456				TRAP	C#ERRHRD
	044326	002130				.WORD	1112
	044330	004745				.WORD	PKTRAM
	044332	015570				.WORD	RAMERR
3238	044334			ENDSEG			
	044334						

F14

TSV5A - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-68

SEQ 0174

TEST 11: NON-TAPE MOTION COMMANDS

3331	044600	010001		MOV	R0,R1		;SAVE CONTENTS OF TSSR		
3332	044602			ERRDF	ERRNO,SFIERR,SFIMSG		;DEVICE FATAL ERROR DURING INIT		
	044602	104455					TRAP	C#ERDF	
	044604	002137					.WORD	1119	
	044606	003652					.WORD	SFIERR	
	044610	012104					.WORD	SFIMSG	
3333	044612								
3334	044612	012704	045070	3#: MOV	#T11PK2,R4		;WRITE CHARACTERISTICS PACKET		
3335	044616	004737	010662	JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS		
3336	044622	103404		BCS	4#		;BR, IF COMMAND ISSUED OK		
3340	044624			ERRHRD	ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTICS FAILED		
	044624	104456					TRAP	C#ERHRD	
	044626	002140					.WORD	1120	
	044630	005056					.WORD	WRTMSG	
	044632	012104					.WORD	SFIMSG	
3341	044634								
3342	044634	004737	045674	4#: JSR	PC,T11REST		;SET UP PACKET FOR COMMAND		
3343	044640	012704	045020	MOV	#T11PACKET,R4		;GET THE ADDRESS OF COMMAND PACKET		
3344	044644								
3345	044644	005037	002222	5#: CLR	INTRECV		;CLEAR INTERRUPT RECEIVED FLAG		
3346	044650	052714	007000	10#: BIS	#007000,(R4)		;SET TO NON-ZERO MODE		
3347	044654	010465	000000	MOV	R4,TSD8(R5)		;SET THE PACKET ADDRESS		
3348	044660	004737	016416	JSR	PC,CHKTSSR		;WAIT FOR SSR TO SET		
3349	044664	103405		BCS	15#		;BR IF CARRY SET (GOOD RETURN)		
3350	044666	010001		MOV	R0,R1		;SAVE CONTENTS OF TSSR		
3354	044670			ERRDF	ERRNO,T11SR2,PKTSSR		;DEVICE FATAL SSR FAILED TO SET		
	044670	104455					TRAP	C#ERDF	
	044672	002141					.WORD	1121	
	044674	045440					.WORD	T11SR2	
	044676	012116					.WORD	PKTSSR	
3355	044700			15#: CKLOOP			;LOOP ON ERROR, IF FLAG SET		
	044700	104406					TRAP	C#CLP1	
3356	044702			ESCAPE	SUB		;BY-PASS SUBTEST IF FATAL ERROR		
	044702	104410					TRAP	C#ESCAPE	
	044704	000076					.WORD	L10106--	
3357	044706	005737	002222	TST	INTRECV		;DID AN INTERRUPT OCCUR ?		
3358	044712	001004		BNE	22#		;BRANCH IF YES		
3362	044714			ERRHRD	ERRNO,T11NINT,PKTSSR				
	044714	104456					TRAP	C#ERHRD	
	044716	002142					.WORD	1122	
	044720	045514					.WORD	T11NINT	
	044722	012116					.WORD	PKTSSR	
3363	044724	016501	000002	22#: MOV	TSSR(R5),R1		;GET THE CONTENTS OF TSSR		
3364	044730	012702	100206	MOV	#SC!SSR!TSREJ,R2		;EXPECTED CONTENTS OF TSSR		
3365	044734	032701	000100	BIT	#0FL,R1		;IS OFF-LINE BIT SET ?		
3366	044740	001402		BEQ	25#		;BRANCH IF NOT OFF-LINE		
3367	044742	052702	000100	BIS	#0FL,R2		;SET OFF-LINE IN EXPECTED DATA		
3368	044746	020201		25#: CMP	R2,R1		;DOES EXPECTED MATCH RECEIVED ?		
3369	044750	001404		BEQ	30#		;OKAY IF MATCH		
3373	044752			ERRHRD	ERRNO,T114REJ,PKTSSR		;COMMAND NOT REJECTED		
	044752	104456					TRAP	C#ERHRD	
	044754	002143					.WORD	1123	
	044756	045303					.WORD	T114REJ	
	044760	012116					.WORD	PKTSSR	
3374	044762			30#: JSR	PC,CKRAM				
3375									
3376	044762	004737	011154	35#: JSR	PC,CKRAM		;CHECK RAM TO MEMORY		

G14

SEQ 0175

TEST 11: NON-TAPE MOTION COMMANDS

Address	Hex	Dec	Label	Value	Comment
3377	044766	103404			
3381	044770		BCS	594	;RAM OK GO ON
	044770	104456	ERRHRD	ERRNO,PKTRAM,RAMERR	;THEY DON'T MATCH
	044772	002144			TRAP
	044774	004745			.WORD
	044776	015570			.WORD
3382	045000		594:		C#ERHRD
3383	045000		ENDSEG		1124
	045000	104405			PKTRAM
3384	045002		ENDSUB		RAMERR
	045002	104403			
3385	045004				
3386	045004	104432	EXIT	TST	
	045006	000772			;ALL DONE THIS TEST
3387					TRAP
3388					.WORD
3389					C#EXIT
3390					L10102-
3391					
3393	045010				
3395	045020		T11PACKET:		
3396	045020	100204	.BLKB	10-<.-TSV2&7>	
3397	045022	045030	.WORD	100204	;COMMAND PACKET FOR TEST
3398	045024	000000	.WORD	T11DATA	;WRITE CHAR COMMAND, WITH IE, ACK
3399	045026	000010	.WORD	0	;ADDRESS OF CHARACTERISTICS BLOCK
3400			.WORD	8.	;STARTING VALUE OF BLOCK SIZE
3401	045030		T11DATA:		
3402	045030	045042	.WORD	T11BFR	;CHARACTERISTICS DATA BLOCK
3403	045032	000000	.WORD	0	;ADDRESS OF MESSAGE BUFFER
3404	045034	000016	.WORD	14.	;LENGTH OF MESSAGE BUFFER
3405	045036	000000	.WORD	0,0	
3406					
3407	045042		T11BFR:	.BLKW 8.	;MESSAGE BUFFER
3408					
3409					
3411	045062				
3413	045070		T11PK2:	.BLKB 10-<.-TSV2&7>	
3414	045070	100204	.WORD	100204	;COMMAND PACKET FOR TEST
3415	045072	045100	.WORD	T11DTA	;WRITE CHAR COMMAND, WITH IE, ACK
3416	045074	000000	.WORD	0	;ADDRESS OF CHARACTERISTICS BLOCK
3417	045076	000010	.WORD	8.	;STARTING VALUE OF BLOCK SIZE
3418					
3419	045100		T11DTA:		
3420	045100	045112	.WORD	T11BF2	;CHARACTERISTICS DATA BLOCK
3421	045102	000000	.WORD	0	;ADDRESS OF MESSAGE BUFFER
3422	045104	000016	.WORD	14.	;LENGTH OF MESSAGE BUFFER
3423	045106	000000	.WORD	0,0	
3424					
3425	045112		T11BF2:	.BLKW 8.	;MESSAGE BUFFER
3426					
3427					
3428					
3429					
3430					

I14

TSVSA - HARDWARE TESTS MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 85-71

SEQ 0177

TEST 11: NON-TAPE MOTION COMMANDS

046000
046000 104401
3488 046002

ENDMOD

L10102: TRAP C\$ETST

J14

TEST 11: NON-TAPE MOTION COMMANDS

```

1      .TITLE  TSV6 - PARAMETER CODING
7
12
18
19 046002      BGNMOD  TSV6
    046002      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 046002      BGNHRD
    046002      .WORD  L10107-L$HARD/2
    046004      L$HARD::
32
33 046004      GPRMA  HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
    046004      .WORD  T$CODE
    046006      .WORD  HPM1
    046010      .WORD  T$LLOLM
    046012      .WORD  T$HILIM
34 046014      GPRMA  HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
    046014      .WORD  T$CODE
    046016      .WORD  HPM2
    046020      .WORD  T$LLOLM
    046022      .WORD  T$HILIM
35      .GPRMD  HPM3,4,0,340,0,7,YES                  ;GET INTERRUPT PRIORITY.
36 046024      ENDRD
    046024      .EVEN
37 046024      L10107:
    104      105      126  HPM1:  .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
38 046060      111      116      124  HPM2:  .ASCIZ  'INTERRUPT VECTOR '
39 046104      111      116      124  HPM3:  .ASCIZ  'INTERRUPT PRIORITY '
40      .EVEN

```


K14

SOFTWARE PARAMETER CODING SECTION

```

42                                     .SBTTL  SOFTWARE PARAMETER CODING SECTION
43
44
45      ;**
46      ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
47      ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
48      ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
49      ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
50      ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
51      ; WITH THE OPERATOR.
52      ;--
53      BGNSFT
54      .WORD L10110-L$SOFT/2
55      L$SOFT::
56      ; GPRM1  SPM1,0,-1,YES          ; GET TRANSPORT TEST FLAG.
57      ; GPRM1  SPM4,2,-1,YES          ; GET ITERATION CONTROL.
58      .WORD   T$CODE
59      .WORD   SPM4
60      .WORD   -1
61      ; GPRM6  SPM6,4,D,7777,0,7777,YES  ; GET LOCAL ERROR LIMIT
62      ; GPRM7  SPM7,6,D,7777,0,7777,YES  ; GET GLOBAL ERROR LIMIT
63      ENDSFT
64      .EVEN
65      L10110:
66      SPM1:  .ASCIZ  'ENABLE TRANSPORT TESTS '
67      SPM4:  .ASCIZ  'INHIBIT ITERATIONS '
68      SPM6:  .ASCIZ  'PER TEST ERROR LIMIT '
69      SPM7:  .ASCIZ  'PER UNIT ERROR LIMIT '
70      .SBTTL  PATCH AREA
71
72      ;
73      ; FINALLY A GENEROUS PATCH AREA.
74      ;
75      ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
76      ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
77      ;
78      PATCH::
79      .BLKW  32.
80      . = !377*1
81      LASTAD          ;SET LAST USED ADDRESS.
82      .EVEN
83      .WORD  0
84      .WORD  0
85      L$LAST::
86      ENDMOD
87      .END

```

Symbol table

ADDSSR	012176	G	C#AU	=	000052	DEVDR0	023412	FRESIZ	003126	G	INTFLA	016225					
ADR	=	000020	G	C#AUTO	=	000061	DEVNRD	023331	FUSI	004117	INTMAS	016224					
AMBTSS	006635		C#BRK	=	000022	DEVNXR	023247	F#AU	=	000015	INTR	016276					
ASSEMB	=	000010	C#BSEG	=	000004	DEVONL	023177	F#AUTO	=	000020	INTREC	002222					
A1716	=	000003	C#BSUB	=	000002	DEVSUM	023142	F#BGN	=	000040	INTVEC	016226					
BADDAT	003156	G	C#CEFG	=	000045	DFPTBL	002154	F#CLEA	=	000007	INTX	004300					
BADSSR	015760	G	C#CLCK	=	000062	DIAGMC	=	000000	F#DU	=	000016	INVERT	021222				
BDVPCR	=	177520	G	C#CLEA	=	000012	DICEA	=	000001	F#END	=	000041	IOKCKI	=	000200		
BENBSW	002230	G	C#CLOS	=	000035	DSBINT	016264	F#HARD	=	000004	IOKSTP	=	000001				
BIE	=	040000	C#CLP1	=	000006	DUAD12	004643	F#HW	=	000013	IPRI	=	002210	G			
BIT0	=	000001	G	C#CVEC	=	000036	DUFLG	003112	F#INIT	=	000006	ISR	=	000100	G		
BIT00	=	000001	G	C#DCLN	=	000044	DUMMY	003062	F#JMP	=	000050	IVEC	=	002206	G		
BIT01	=	000002	G	C#DODU	=	000051	EF.CON	=	000036	F#MOD	=	000000	IXE	=	004000	G	
BIT02	=	000004	G	C#DRPT	=	000024	EF.NEW	=	000035	F#MSG	=	000011	I#AU	=	000041		
BIT03	=	000010	G	C#DU	=	000053	EF.PWR	=	000034	F#PROT	=	000021	I#AUTO	=	000041		
BIT04	=	000020	G	C#EDIT	=	000003	EF.RES	=	000037	F#PWR	=	000017	I#CLN	=	000041		
BIT05	=	000040	G	C#ERDF	=	000055	EF.STA	=	000040	F#RPT	=	000012	I#DU	=	000041		
BIT06	=	000100	G	C#ERHR	=	000056	EMAXDU	017057	F#SEG	=	000003	I#HRD	=	000041			
BIT07	=	000200	G	C#ERRO	=	000060	EN	=	000000	F#SOFT	=	000005	I#INIT	=	000041		
BIT08	=	000400	G	C#ERSF	=	000054	ENAIN	016232	F#SRV	=	000010	I#MOD	=	000041			
BIT09	=	001000	G	C#ERSO	=	000057	ENVIRN	020710	F#SUB	=	000002	I#MSG	=	000041			
BIT1	=	000002	G	C#ESCA	=	000010	EPRTSW	002176	F#SW	=	000014	I#PROT	=	000040			
BIT10	=	002000	G	C#ESEG	=	000005	EPRT1	006360	F#TEST	=	000001	I#PTAB	=	000041			
BIT11	=	004000	G	C#ESUB	=	000003	EPRT2	006360	GDDAT	003160	G	I#PWR	=	000041			
BIT12	=	010000	G	C#ETST	=	000001	ERCM	012003	GERRMA	002172	G	I#RPT	=	000241			
BIT13	=	020000	G	C#EXIT	=	000032	ERRHI	002236	GETPAT	020254	G	I#SEG	=	000041			
BIT14	=	040000	G	C#GETB	=	000026	ERRK	017036	GETSEL	020336	G	I#SETU	=	000041			
BIT15	=	100000	G	C#GETW	=	000027	ERRLO	002240	G#CNT0	=	000200	I#SFT	=	000041			
BIT2	=	000004	G	C#GHAN	=	000043	ERRNO	=	002144	G#DELM	=	000372	I#SRV	=	000041		
BIT3	=	000010	G	C#GPHR	=	000042	ERRVEC	=	000004	G#DISP	=	000003	I#SUB	=	000041		
BIT4	=	000020	G	C#GPLO	=	000030	ERTABE	003376	G#EXCP	=	000400	I#TST	=	000041			
BIT5	=	000040	G	C#GPRI	=	000040	ERTABL	003176	G#HILI	=	000002	J#JMP	=	000167			
BIT6	=	000100	G	C#INIT	=	000011	ESUM	017040	G#LOLI	=	000001	KIPAR0	=	172340			
BIT7	=	000200	G	C#INLP	=	000020	EVL	=	000004	G#NO	=	000000	KIPAR1	=	172342		
BIT8	=	000400	G	C#MANI	=	000050	EXBCNT	=	000010	G#OFFS	=	000400	KIPAR2	=	172344		
BIT9	=	001000	G	C#MEM	=	000031	EXPBRE	015562	G#FSI	=	000376	KIPAR3	=	172346			
BOE	=	000400	G	C#MSG	=	000023	EXPD	002232	G#PRMA	=	000001	KIPAR4	=	172350			
BRINIT	004457		C#OPEN	=	000034	EXPGET	004533	G#PRMD	=	000002	KIPAR5	=	172352				
BSEL0	=	000000	C#PNTB	=	000014	EXPGET2	004567	G#PRML	=	000000	KIPAR6	=	172354				
BSEL1	=	000001	C#PNTF	=	000017	EXPMMSG	002322	G#RADA	=	000140	KIPAR7	=	172356				
CHKAMB	016124		C#PNTS	=	000016	EXPREC	015554	G#RAD8	=	000000	KIPDR0	=	172300				
CHKMAN	020560	G	C#PNTX	=	000015	EXTA	005772	G#RADD	=	000040	KIPDR1	=	172302				
CHKTSS	016416		C#QIO	=	000377	EXTEND	005770	G#RADL	=	000120	KIPDR2	=	172304				
CKDROP	017262		C#RDBU	=	000007	EXTFEA	002224	G#RAD0	=	000020	KIPDR3	=	172306				
CKEMAX	017162		C#REFG	=	000047	E#END	=	002100	G#XFER	=	000004	KIPDR4	=	172310			
CKMSG	011420	G	C#RESE	=	000033	E#LOAD	=	000035	G#YES	=	000010	KIPDR5	=	172312			
CKMSG2	011550	G	C#REVI	=	000003	FATERR	=	000060	HIADDR	=	001400	KIPDR6	=	172314			
CKRAM	011154	G	C#RFLA	=	000021	FATFLG	002220	HOE	=	100000	G	KIPDR7	=	172316			
CKRAM2	011264	G	C#RPT	=	000025	FERCM	011772	HPM1	=	046024		KTENAB	=	003134	G		
CHDPKT	021274	G	C#SEFG	=	000046	FIFEXP	012240	HPM2	=	046060		KTFLG	=	003132	G		
CHPMEM	017740		C#SPRI	=	000041	FIF1MS	012312	HPM3	=	046104		KTINIT	=	021070			
CONFIG	017330		C#SVEC	=	000037	FIF2MS	012361	IBE	=	010000	G	KTOFF	=	017354			
COUNT	002310	G	C#TPRI	=	000013	FILLME	017502	IDU	=	000040	G	KTON	=	017336			
CSRADD	002204	G	DATA	=	002312	G	FNOINT	004215	IER	=	020000	G	LERRMA	=	002170	G	
CTAB	003164	G	DATASC	=	020312		FORCER	002174	IFALT	=	004256		LISTAL	=	000001		
CTABE	003176	G	DEBUGM	=	011702		FREE	003124	INCERK	=	017124		LOE	=	040000	G	
CTABM	003164	G	DEVcnt	=	002216	G	FREEHI	003130	INTCPC	=	016230		LOOPCN	=	002214	G	

M14

TSV6 - PARAMETER CODING MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 87-2

SEQ 0181

Symbol table

LOOPCO	013176	L10001	002174	L10073	037126	NXR	003740	PRI05	=	000240	G
LOOPFL	003162	L10002	005766	L10074	037332	NXRERR	005736	PRI06	=	000300	G
LOT	= 000010	L10003	012114	L10075	043632	NXR	003777	PRI07	=	000340	G
L\$ACP	002110	L10004	012132	L10076	041122	NXTU	022036	PRMESS	=	014312	
L\$APT	002036	L10005	012150	L10077	041514	OFL	= 000100	PRMNO	=	002320	G
L\$AU	022366	L10006	012156	L10100	042404	ONEFIL	= 000000	PRMSG	=	014632	G
L\$AUT	002070	L10007	012174	L10101	042632	O\$APTS	= 000000	PRMSG0	=	015012	
L\$AUTO	022572	L10010	012212	L10102	046000	O\$AU	= 000001	PRMSG1	=	015057	
L\$CCP	002106	L10011	012236	L10103	044100	O\$BGNR	= 000001	PRMSG2	=	015115	
L\$CLEA	022652	L10012	012310	L10104	044336	O\$BGNS	= 000001	PROASC	=	014500	
L\$CO	002032	L10013	012460	L10105	044556	O\$DU	= 000001	PR1ASC	=	014545	
L\$DEPO	002011	L10014	013174	L10106	045002	O\$ERRT	= 000000	PST32W	=	003152	G
L\$DESC	003410	L10015	014022	L10107	046024	O\$GNSW	= 000001	PUNIT	=	022320	
L\$DESP	002076	L10016	014044	L10110	046144	O\$POIN	= 000001	PW.D11	=	000021	
L\$DEVP	002060	L10017	015560	MEMADD	014024	O\$SETU	= 000000	PW.D13	=	000022	
L\$DISP	002124	L10020	015566	MEMCK	021312	PASRPT	022070	PW.D22	=	000020	
L\$DLY	002116	L10021	015574	MENASC	020527	PATCH	046304	PW.NOP	=	000000	
L\$DTP	002040	L10022	015606	MENERR	020454	PATDAT	020310	PW.NO1	=	000023	
L\$DTYP	002034	L10023	015630	MENRES	020556	PC.ERA	= 002400	PW.RDE	=	000024	
L\$DU	022464	L10024	015656	MIVEC	= 000250	PC.IER	= 002000	PW.RDR	=	000001	
L\$DUT	002072	L10025	016016	MSA.FR	= 000006	PC.NOO	= 001000	PW.RDS	=	000005	
L\$DVTY	003402	L10026	016326	MSA.NO	= 000000	PC.REL	= 000000	PW.RFI	=	000003	
L\$EF	002052	L10030	022316	MSA.NR	= 000004	PC.REW	= 000400	PW.WCT	=	000006	
L\$ENVI	002044	L10031	022462	MSA.VO	= 000002	PKBCNT	= 000006	PW.WFI	=	000004	
L\$ETP	002102	L10032	022570	MSGEXP	012214	PKHI	= 000004	PW.WFM	=	000007	
L\$EXP1	002046	L10033	022650	MSGLOO	013134	PKLOW	= 000002	PW.WHI	=	000010	
L\$EXP4	002064	L10034	022676	MSGSTA	012420	PKTADD	007554	PW.WNP	=	000011	
L\$EXP5	002066	L10035	023140	MSGSUB	014012	PKTFRM	007516	PW.WTR	=	000002	
L\$HARD	046004	L10036	023700	MS.ATT	= 000006	PKTGET	012134	P.ACK	=	100000	
L\$HIME	002120	L10037	023546	MS.EXT	= 000200	PKTMES	012160	P.CMD	=	000037	
L\$HPCP	002016	L10040	023630	MS.RSD	= 000001	PKTRAM	004745	P.CONT	=	000012	
L\$HPTP	002022	L10041	024376	MS.RSF	= 000020	PKTSSR	012116	P.CVC	=	040000	
L\$HM	002154	L10042	025070	MS.RST	= 000010	PNT	= 001000	P.FMT	=	000140	
L\$ICP	002104	L10043	026424	M2901	026150	PRAMPK	014046	P.FORM	=	000011	
L\$INIT	021572	L10044	025332	M8186	005554	PRASC	014603	P.GETS	=	000017	
L\$LADP	002026	L10045	025632	M8189	005645	PRBEXP	015550	P.IE	=	000200	
L\$LAST	047004	L10046	026130	NBA	= 002000	PRBMSG	015416	P.INIT	=	000013	
L\$LOAD	002100	L10047	027530	NEWPAS	022024	PRBREC	015552	P.MODE	=	007400	
L\$LUN	002074	L10050	026776	NODEV	003114	PRBTOT	015503	P.OPP	=	020000	
L\$PREV	002050	L10051	027346	NOINIT	004335	PRBYTE	015202	P.POSI	=	000010	
L\$NAME	002000	L10052	031002	NOINTR	004221	PRI	= 002000	P.READ	=	000001	
L\$PRIO	002042	L10053	030060	NOITS	002166	PRIADD	010160	P.SMB	=	010000	
L\$PROT	021562	L10054	030404	NOMAN	020614	PRIAO	010230	P.WRIT	=	000005	
L\$PRT	002112	L10055	034460	NOMEM	005460	PRIBXO	007612	P.WRTC	=	000004	
L\$REPP	002062	L10056	031410	NP.IR	= 000200	PRIEQU	010060	P.WRTS	=	000006	
L\$REV	002010	L10057	031674	NP.LOO	= 000040	PRIPKT	007370	QVP	=	002202	G
L\$RPT	022700	L10060	032140	NP.OUT	= 000100	PRIRAM	010066	RAMASC	=	014226	
L\$SOFT	046136	L10061	032366	NP.WRP	= 000020	PRITAD	010274	RAMDAT	=	002242	G
L\$SPC	002056	L10062	032632	NSI	004152	PRITSS	006024	RAMERR	=	015570	G
L\$SPCP	002020	L10063	033202	NSINIT	004407	PRITO	010356	RAMEXP	=	015610	G
L\$SPTP	002024	L10064	035364	NUL	004527	PRIT1	010421	RAMFOR	=	010116	
L\$STA	002030	L10065	040520	NULCR	004530	PRI XOR	007742	RAMSIZ	=	002302	G
L\$SW	002164	L10066	035626	NXM	= 004000	PRI00	= 000000	RAMTAD	=	015576	G
L\$TEST	002114	L10067	036076	NXMFLG	003136	PRI01	= 000040	RCVHIA	=	002304	G
L\$TIM	002014	L10070	036302	NXMI	003142	PRI02	= 000100	RCVLOA	=	002306	G
L\$UNIT	002012	L10071	036470	NXMLO	003140	PRI03	= 000140	RDERR	=	005206	
L10000	002162	L10072	036674	NXMTST	021466	PRI04	= 000200	RECMG	=	002466	G

Symbol table

RECV	002234	G	S1.ICE=	002000	TST5ID	027400	T1LOOP	023500	T3TSSR	024726	
REGSAV	020220		S1.IEO=	010000	TST6ID	030763	T1.1	023502	T4	025072	G
RETERR	005372		S1.IFM=	001000	TST7ID	034363	T1.2	023562	T4LOOP	025112	
REV	002226	G	S1.IHE=	000400	TST8ID	035347	T10	040522	T4.1	025072	
REWIND	011054	G	S1.IID=	004000	TST9ID	040421	T10BFR	042672	T4.2	025334	
RMCHBE=	000167		S1.IIR=	020000	TSV2	002000	T10BUF	042734	T4.3	025634	
RMCHEN=	000200		S1.I2R=	040000	TSV3	002174	T10DAT	042660	T5	026426	G
RMMSGB=	000215		S1.PAR=	100000	TSV4	021562	T10DTA	042722	T5ADDR	027466	
RMMSGC=	000234		S2.ATI=	000010	TSV5	023462	T10INT	043370	T5LOOP	026444	
RMPKTB=	000201		S2.BTI=	000004	TSV6	046002	T10LOO	040540	T5MEM	027430	
RMPKTE=	000210		S2.DIM=	000200	TTIBFR=	177562	T10MBF	042754	T5.1	026450	
RMR	010000		S2.ILW=	000100	TTICSR=	177560	T10NBA	043051	T5.2	027016	
RMPACK	011150		S2.INR=	000020	TTIVEC=	000060	T10NIN	043277	T6	027532	G
SC	100000		S2.OUT=	000040	T#ARGC=	000003	T10NNB	043133	T6INT	030553	
SCE	020000		S2.UND=	000003	T#CODE=	001130	T10PAC	042650	T6LOOP	027550	
SCERR	005300		TBEND=	003062	T#ERRN=	002144	T10PKT	042712	T6NBA	030450	
SCME	005013		TCOASC	006476	T#EXCP=	000000	T10RST	043506	T6NINT	030631	
SDELAY	010660		TCOCOD	006676	T#FLAG=	000040	T10RT2	043560	T6PACK	030440	
SELASC	020522		TEMP1	003116	T#GMAN=	000000	T10SSR	043210	T6SSR	030475	
SELDAT=	000004		TEMP2	003120	T#HILI=	000776	T10.1	040540	T6TSBA	030711	
SEL2	000002		TERCLS=	000016	T#LAST=	000001	T10.2	041124	T6.1	027550	
SETMAP	017376		TESTNO=	000013	T#LOLI=	000000	T10.3	041516	T6.2	030074	
SETU	022122		TEXASC	006435	T#LSYM=	010000	T10.4	042406	T7	031004	G
SFFMSG	012152	G	TFCASC	006537	T#LTNO=	000013	T11	043634	T7BFR	033256	
SFHERR	003705		TIMEXP	015632	T#NEST=	177777	T11BFR	045042	T7DATA	033240	
SFIERR	003652		TIMSGO	015660	T#NSO =	000000	T11BF2	045112	T7INT	034211	
SFIMSG	012104	G	TINERR	012071	T#NS1 =	000005	T11DAT	045030	T7LOOP	031022	
SFPTBL	002164	G	TMPBFR	002632	T#NS2 =	000002	T11DTA	045100	T7NBA	033370	
SIFLAG	003154	G	TNAM	016764	T#NS3 =	000003	T11LOO	043652	T7NINT	034120	
SIMSG	012036		TRANST	002164	T#PTNU=	000000	T11NBA	045132	T7PACK	033230	
SKIPT	003400		TSBA	000000	T#SAVL=	177777	T11NIN	045514	T7REST	034412	
SOFINI	016054	G	TSBAH =	000001	T#SEGL=	177777	T11PAC	045020	T7SP	033250	
SPACE	010466	G	TSBAH2	026240	T#SEKO=	010000	T11PK2	045070	T7SSR	034031	
SPM1	046144		TSBAMS	026322	T#SUBN=	000004	T11RES	045674	T7TSBA	034300	
SPM4	046174		TSDB	000000	T#TAGL=	177777	T11RT2	045736	T7.1	031022	
SPM6	046224		TSDBH =	000001	T#TAGN=	010111	T11SR2	045440	T7.2	031424	
SPM7	046254		TSFCOD	007236	T#TEMP=	000000	T11SSR	045364	T7.3	031676	
SRO	177572		TSREJ =	000006	T#TEST=	000013	T11TSB	045572	T7.4	032142	
SR1	177574		TSSDEF	006606	T#TSTM=	177777	T11.1	043652	T7.5	032370	
SR2	177576		TSSR	000002	T#TSTS=	000001	T11.2	044114	T7.6	032634	
SR3	172516		TSSRBI	003502	T#AU =	010031	T11.3	044340	T72DAT	033276	
SSR	000200		TSSRFO	006415	T#AUT=	010033	T11.4	044560	T72DON=	033312	
STATCO	012462		TSSRM =	000003	T#CLE=	010034	T112RE	045172	T72NBA	033312	
SVCGBL=	000000		TSSX	004020	T#DU =	010032	T113RE	045253	T72REJ	033443	
SVCINS=	000000		TSTBLK	002752	T#HAR=	010107	T114RE	045303	T73REJ	033542	
SVCSUB=	000001		TSTCNT	002212	T#HMJ =	010000	T2	023702	T74REJ	033635	
SVCTAG=	000000		TSTEND	017000	T#INI=	010030	T2LOOP	023720	T75REJ	033733	
SVCTST=	000001		TSTFLA	002314	T#MSG=	010025	T2SSR	024276	T8	034462	G
S#LSYM=	010000		TSTL00	016536	T#PRO=	010027	T2TSBA	024164	T8BFR	035112	
SO.IDB=	000010		TSTPTR	002316	T#RPT=	010035	T2TSSR	024231	T8DATA	035100	
SO.IFB=	000002		TSTSET	016570	T#SEG=	010000	T23A	003144	T8LOOP	034500	
SO.IFP=	000001		TST1ID	023660	T#SOF=	010110	T23B	003146	T8NVCK	035167	
SO.ILD=	000020		TST1OI	043457	T#SRV=	010026	T3	024400	T8PACK	035070	
SO.ION=	000040		TST1II	045642	T#SUB=	010106	T3BFLG	003150	T8SSR	035260	
SO.IRD=	000100		TST2ID	024350	T#SW =	010001	T3LOOP	024416	T8VCK	035132	
SO.IRW=	000004		TST3ID	025043	T#TES=	010102	T3SSR	024772	T9	035366	G
SO.ISP=	000200		TST4ID	026402	T1	023462	T3TSBA	024662	T9BFR	037372	

B15

TSV6 - PARAMETER CODING MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 87-4

SEQ 0183

Symbol table

T9DATA 037360	T95REJ 037771	WF.I4R= 000001	XSONEF= 002000	X2.EXT= 000200
T9INT 040247	UAM = 000200 G	WRTCHR 010662 G	XSOONL= 000100	X2.OPM= 100000
T9LOOP 035410	UNITN 002200 G	WRTERR 005113	XSOPED= 000010	X2.RCE= 040000
T9NBA 037426	UNREC = 000006	WRTMSG 005056	XSORLL= 010000	X2.REV= 000077
T9NINT 040156	USI 004123	WSMBK 021304 G	XSORLS= 040000	X2.SPA= 035400
T9PACK 037350	WAITF 016330 G	XFERAS 016020	XSOYMK= 100000	X2.UNI= 000007
T9REST 040446	WC.IFA= 000200	XNXM 016456	XSOVCK= 000020	X2.WCF= 002000
T9SSR 040067	WC.IFE= 000002	XORBF0 007674	XSOWLE= 004000	X3.DCK= 000010
T9TSBA 040336	WC.IGO= 000001	XORFOR 010012	XSOWLK= 000004	X3.MBZ= 000006
T9.1 035410	WC.IRE= 000010	XST0 = 000006 G	XXCOMM 003122 G	X3.MDE= 177400
T9.2 035666	WC.IRW= 000004	XST1 = 000010 G	X#ALWA= 000000	X3.OPI= 000100
T9.3 036100	WC.IOT= 000100	XST2 = 000012 G	X#FALS= 000040	X3.REV= 000040
T9.4 036304	WC.IIT= 000040	XST3 = 000014 G	X#OFFS= 000400	X3.RIB= 000001
T9.5 036472	WC.ISR= 000020	XST4 = 000016 G	X#TRUE= 000020	X3.SPA= 000200
T9.6 036676	WF.IED= 000010	XSOBOT= 000002	X1.COR= 020000	X3.TRF= 000020
T9.7 037130	WF.IER= 000004	XSOEOT= 000001	X1.DLT= 100000	X4.HSP= 100000
T92DAT 037412	WF.IHI= 000200	XSOIE = 000040	X1.MBZ= 017375	X4.MBZ= 017400
T92DON= 037426	WF.IRE= 000040	XSOILA= 000400	X1.RBP= 000400	X4.RCE= 040000
T92REJ 037501	WF.IWF= 000020	XSOILC= 001000	X1.SPA= 040000	X4.TSM= 020000
T93REJ 037600	WF.IWR= 000100	XSOLET= 020000	X1.UNC= 000002	X4.WRC= 000377
T94REJ 037673	WF.I3R= 000002	XSOMOT= 000200	X2.BUF= 000100	

. ABS. 047004 000 (RW,I,GBL,ABS,OVR)
 000000 001 (RW,I,LCL,REL,CON)
 ABS 000000 002 (RW,I,LCL,REL,CON)

Errors detected: 0

*** Assembler statistics

Work file reads: 273
 Work file writes: 268
 Size of work file: 28912 Words (113 Pages)
 Size of core pool: 19684 Words (75 Pages)
 Operating system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:04:47.69

CVTSAC,CVTSAC/-SP-SVC/PL,TSV1A,TSV22A,TSV3B,TSV4,TSV5A,TSV6