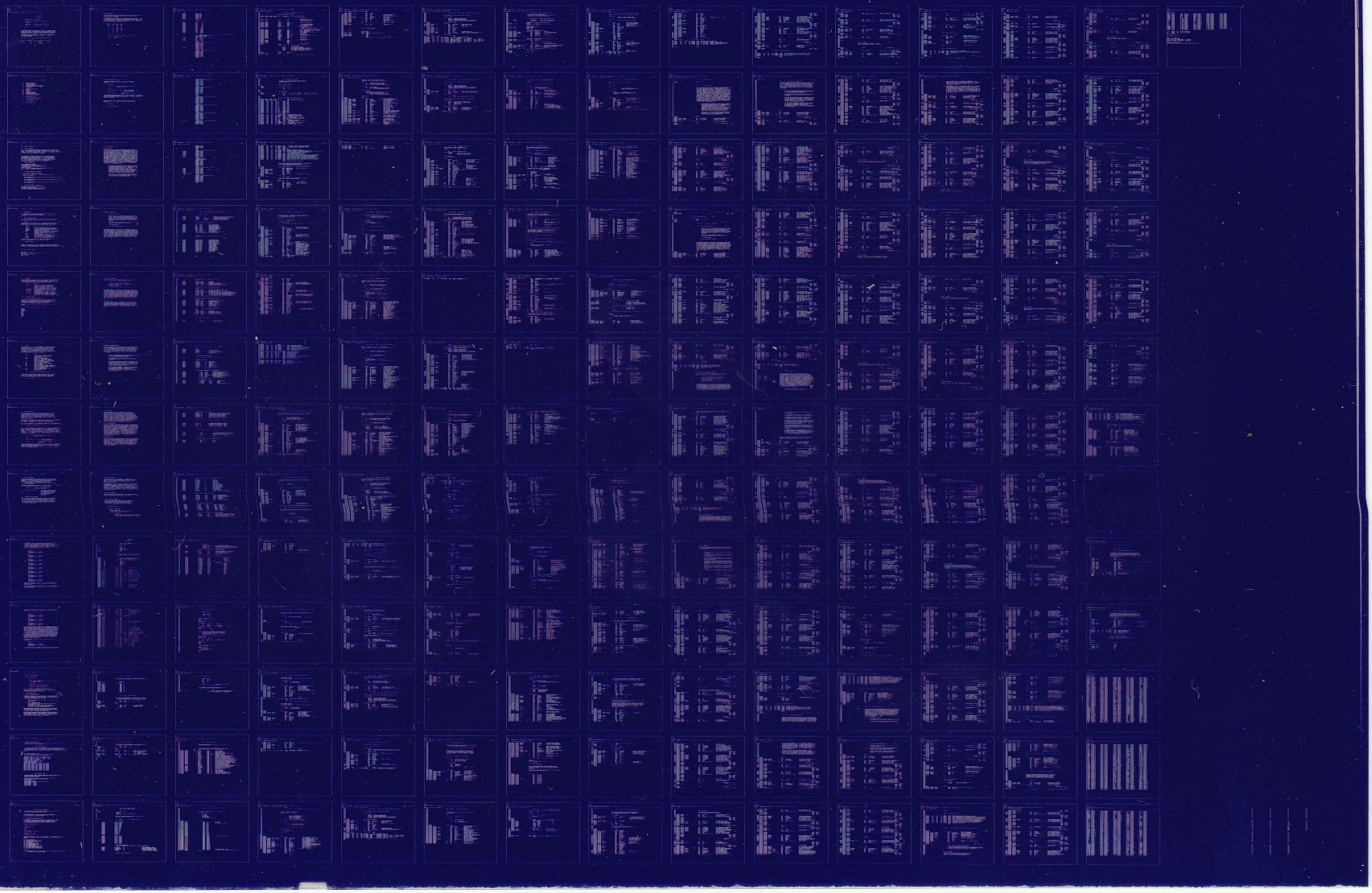
TSV05

TSV05 CTRL PT 1
CVTSAC0

AH-T094C-MC
1 OF 1 SEP 1987
 MADE IN USA

digital



.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

TABLE OF CONTENTS

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 (HSAAA.SYS VERSION 34 OR LATER)
PDP-11 DIAGNOSTIC LOADER/MONITOR (XXDP+)

1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

1. CHQUS XXDP+ USERS GUIDE; DOCUMENT NUMBER AC-F348E-MC
   DATE: 14 JULY 1980.

2. TSV05 TRANSPORT SUBSYSTEM USER'S GUIDE; DOCUMENT NUMBER EK-TSV05-UG-001
   DATE: AUGUST 1983

3. TSV05 TRANSPORT SUBSYSTEM TECHNICAL MANUAL; DOCUMENT NUMBER EK-TSV05-TM-001
   DATE: AUGUST 1983

4. TSV05 TRANSPORT SUBSYSTEM INSTALLATION MANUAL; DOCUMENT NUMBER EK-TSV05-IN-001
   DATE: AUGUST 1983

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

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

1.5 ASSUMPTIONS

ALL HARDWARE EXCEPT THE HARDWARE UNDER TEST IS ASSUMED TO WORK PROPERLY OR FALSE ERRORS CAN BE REPORTED.
THE TAPE BEING USED ON THE 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.

```
0 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 "0 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 (O) ?  160000<CR>
SUB-DEVICE # (O) ?  0<CR>
Q-FACTOR (O) 0 ?  1<CR>

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

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

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

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

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

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

UNIT 8
CSR ADDRESS (O)  160000<CR>
SUB-DEVICE # (O) ?  7<CR>
Q-FACTOR (O) 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 (O) ?  160000<CR>
        SUB-DEVICE # (O) ?  0,1<CR>
        Q-FACTOR (O) 0 ?  1.0<CR>

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

        UNIT 7
        CSR ADDRESS (O) ?  160000<CR>
        SUB-DEVICE # (O) ?  6,7<CR>
        Q-FACTOR (O) 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 (O) ?  160000<CR>
        SUB-DEVICE # (O) ?  0-7<CR>
        Q-FACTOR (O) 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 PRIOVIDES 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 SIGINALS IN WORD #9:
DATMIS<7>  ILW<6>  OUTRDY<5>  INRDY<4>
 MESSAGE BUFFER ADDRESS = 047352
 MESSAGE BUFFER CONTENTS:
WORD #0 EXPD: 100020  RECV: 100C20  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 (XSTO) 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 CHARACTREISTICS 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 CONTINUOSLY 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 ERPOR 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 DC005 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 CAPABLITY 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 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.

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 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.

TEST 9: COMPLETION INTERRUPT

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

THE SEQUENCES OF TEST 7 ARE REPEATED, EXCEPT THAT THE INTERRUPT SERVICE ROUTINE IS SET UP TO EXPECT INTERRUPTS AND EACH WRITE CHARACTERISTICS COMMAND IS ISSUED WITH THE IE BIT SET (1). IT IS VERIFIED, WHERE APPROPRIATE, THAT THE IE STATUS BIT IN XST0 OF ANY MESSAGE PACKET IS SET AND THAT A COMPLETION INTERRUPT IS GENERATED. FINALLY, A SEQUENCE OF TWO COMMANDS ARE ISSUED, THE FIRST WITH IE=1 AND THE SECOND WITH IE=0. IT IS VERIFIED THAT NO INTERRUPT IS GENERATED AFTER THE SECOND COMMAND AND THAT THE IE BIT IN XST0 IS 0.

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


```
                                .TITLE  TSV2 - PROGRAM HEADER
                                .SBTTL  PROGRAM HEADER
000000                          .PSECT  ABS

                                .MCALL  SVC
000000                          SVC                     ; INITIALIZE SUPERVISOR MACROS
                                .ENABLE LC
                                .NLIST  BEX,CND
000000                          .ENABL  ABS,AMA
        002000                  .=2000
002000                          BGNMOD  TSV2
002000                  TSV2::

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


002000                          POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
002000                          HEADER  CVTSA,C,0,655.,0
002000                  L$NAME::                ;DIAGNOSTIC NAME
002000  103                     .ASCII /C/
002001  126                     .ASCII /V/
002002  124                     .ASCII /T/
002003  123                     .ASCII /S/
002004  101                     .ASCII /A/
002005  000                     .BYTE   0
002006  000                     .BYTE   0
002007  000                     .BYTE   0
002010                  L$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$CO::
002032  000000                  .WORD   0
002034                  L$DTYP::                ;DIAGNOSTIC TYPE
002034  000000                  .WORD   0
002036                  L$APT::                 ;APT EXPANSION
002036  000000                  .WORD   0
002040                  L$DTP::                 ;PTR. TO DISPATCH TABLE
```

K2

PROGRAM HEADER

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

L2

PROGRAM HEADER

```
                                        .SBTTL  DISPATCH TABLE

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

002122                                  DISPATCH 11
002122  000013                         .WORD   11
002124                          L$DISPATCH::
002124  023462                          .WORD   T1
002126  023702                          .WORD   T2
002130  024400                          .WORD   T3
002132  025072                          .WORD   T4
002134  026426                          .WORD   T5
002136  027532                          .WORD   T6
002140  031004                          .WORD   T7
002142  034462                          .WORD   T8
002144  035366                          .WORD   T9
002146  040522                          .WORD   T10
002150  043634                          .WORD   T11


                                        .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.
                                ;--
002152                                  BGNHW   DFPTBL          ;DEFAULT HARD-P-TABLE
002152  000003                         .WORD   L10000-L$HW/2
002154                          L$HW::
002154                          DFPTBL::

002154  172520                          .WORD   172520          ; 1ST (OF 2) REGISTERS.
002156  000224                          .WORD   224             ; INTERRUPT VECTOR
002160  000200                          .WORD   PRI04           ; INTERRUPT PRIORITY.
002162                                  ENDHW
002162                          L10000:

```

M2

SOFTWARE P-TABLE

```
                                        .SBTTL  SOFTWARE P-TABLE

                                ;**
                                ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
                                ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
                                ;--
002162                                  BGNSW   SFPTBL
002162  000004                         .WORD   L10001-L$SW/2
002164                          L$SW::
002164                          SFPTBL::

002164  000000                  TRANSTST::      .WORD   0       ; ENABLE TEST OF TRANSPORT(S) IF =1
002166  000000                  NOITS::         .WORD   0       ; INHIBIT ITERATION OPTION.
                                                                ; ... 0 = ITERATE
                                                                ; ...NZ = INHIBIT ITERATE.
002170  000017                  LERRMAX::       .WORD   15.     ; LOCAL (PER TEST) ERROR LIMIT
002172  000310                  GERRMAX::       .WORD   200.    ; GLOBAL (PER UNIT) ERROR LIMIT
002174                                  ENDSW
002174                          L10001:

002174                                  ENDMOD
```

N2

SOFTWARE P-TABLE

```
                                .TITLE  TSV3 - GLOBAL AREAS
                                .SBTTL  GLOBAL EQUATES SECTION


002174                          BGNMOD  TSV3
002174                  TSV3.:
                                .SBTTL  GLOBAL EQUATES SECTION

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

002174                          EQUALS          ; GET STANDARD EQUATES.
                                ;
                                ; BIT DIFINITIONS
                                ;
100000                          BIT15== 100000
040000                          BIT14== 40000
020000                          BIT13== 20000
010000                          BIT12== 10000
004000                          BIT11== 4000
002000                          BIT10== 2000
001000                          BIT09== 1000
000400                          BIT08== 400
000200                          BIT07== 200
000100                          BIT06== 100
000040                          BIT05== 40
000020                          BIT04== 20
000010                          BIT03== 10
000004                          BIT02== 4
000002                          BIT01== 2
000001                          BIT00== 1
                                ;
001000                          BIT9==  BIT09
000400                          BIT8==  BIT08
000200                          BIT7==  BIT07
000100                          BIT6==  BIT06
000040                          BIT5==  BIT05
000020                          BIT4==  BIT04
000010                          BIT3==  BIT03
000004                          BIT2==  BIT02
000002                          BIT1==  BIT01
000001                          BIT0==  BIT00
                                ;
                                ; EVENT FLAG DEFINITIONS
                                ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                ;
000040                          EF.START==      32.             ; START COMMAND WAS ISSUED
000037                          EF.RESTART==    31.             ; RESTART COMMAND WAS ISSUED
000036                          EF.CONTINUE==   30.             ; CONTINUE COMMAND WAS ISSUED
000035                          EF.NEW==        29.             ; A NEW PASS HAS BEEN STARTED
000034                          EF.PWR==        28.             ; A POWER-FAIL/POWER-UP OCCURRED
                                ;
                                ;
                                ; PRIORITY LEVEL DEFINITIONS
```

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                   KT11    ;;                  ;DEFINE MEMORY MANAGEMENT REGISTERS
                        .SBTTL  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 REGISTORS
                        UDPDR0= 177620
                        UDPDR1= 177622
                        UDPDR2= 177624
                        UDPDR3= 177626
                        UDPDR4= 177630
                        UDPDR5= 177632
                        UDPDR6= 177634
                        UDPDR7= 177636
                        .ENDC
```

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
```

```
                        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


```
                                .SBTTL  TSV05 REGISTER AND PACKET DEFINITIONS

                        ;
                        ; SOME GENERAL EQUATES.
                        ;

000004                  ERRVEC==        4               ; POINTER TO ERROR VECTOR FOR BUS TIME OUT.
000060                  TTIVEC==        60              ; INTERRUPT VECTOR FOR CONSOLE INPUT
177560                  TTICSR==        177560          ; BUS ADDRESS OF CONSOLE INPUT
177562                  TTIBFR==        177562          ; CONSOLE INPUT DATA BUFFER
177520                  BDVPCR==        177520          ; BDV11 PAGE CONTROL REGISTER

                        ;+
                        ;BIT DEFINITIONS FOR TSSR REGISTER
                        ;-

100000                  SC=     BIT15           ;SPECIAL CONDITION
040000                  BIE=    BIT14           ;BUS INTERFACE ERROR
020000                  SCE=    BIT13           ;SANITY CHECK ERROR
010000                  RMR=    BIT12           ;MODIFICATION REFUSED
004000                  NXM=    BIT11           ;NONEXISTANT MEMORY ERROR
002000                  NBA=    BIT10           ;NEED BUFFER ADDRESS
001400                  HIADDR= BIT9!BIT8       ;EXTENDED ADDRESS BITS
000200                  SSR=    BIT7            ;SUB SYSTEM READY
000100                  OFL=    BIT6            ;OFF LINE BIT
000060                  FATERR= BIT4!BIT5       ;FATAL TERMINATION ERROR CODES
000016                  TERCLS= BIT3!BIT2!BIT1  ;TERMINATION CODES

                        ;+
                        ;
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
                        ;(XST0)
                        ;
                        ;-

100000                  XSOTMK= BIT15           ;TAPE MARK DETECTED
040000                  XSORLS= BIT14           ;RECORD LENGTH SHORT
020000                  XSOLET= BIT13           ;LOGICAL END OF TAPE
010000                  XSORLL= BIT12           ;RECORD LENGTH LONG
004000                  XSOWLE= BIT11           ;WRITE LOCK ERROR
002000                  XSONEF= BIT10           ;NON EXECUTABLE FUNCTION
001000                  XSOILC= BIT9            ;ILLEGAL COMMAND
000400                  XSOILA= BIT8            ;ILLEGAL ADDRESS
000200                  XSOMOT= BIT7            ;TAPE IN MOTION
000100                  XSOONL= BIT6            ;TRANSPORT ON LINE
000040                  XSOIE=  BIT5            ;INTERRUPT ENABLE
000020                  XSOVCK= BIT4            ;VOLUME CHECK BIT
000010                  XSOPED= BIT3            ;PHASE ENCODED DRIVE
000004                  XSOWLK= BIT2            ;WRITE LOCKED
000002                  XSOBOT= BIT1            ;BEGINNING OF TAPE
000001                  XSOEOT= BIT0            ;END OF TAPE
```

TSV05 REGISTER AND PACKET DEFINITIONS

```
                                ;*
                                ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
                                ;(XST1)
                                ;-
        100000                  X1.DLT  = BIT15         ;DATA LATE
        040000                  X1.SPARE= BIT14         ;NOT USED
        020000                  X1.COR  = BIT13         ;CORRECTABLE DATA ERROR
        017375                  X1.MBZ  = BIT12+BIT11+BIT10+BIT9+BIT7+BIT6+BIT5+BIT4+BIT3+BIT2+BIT0 ;ALWAYS 0
        000400                  X1.RBP  = BIT8          ;READ BUS PARITY ERROR
        000002                  X1.UNC  = BIT1          ;UNCORRECTABLE DATA OR HARD ERROR

                                ;*
                                ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
                                ;(XST2)
                                ;-
        100000                  X2.OPM  = BIT15         ;OPERATION IN PROGRESS (TAPE MOVING)
        040000                  X2.RCE  = BIT14         ;RAM CHECKSUM ERROR
        035400                  X2.SPARE= BIT13+BIT12+BIT11+BIT9+BIT8   ;NOT USED BY TSV05 (ALWAYS=0)
        002000                  X2.WCF  = BIT10         ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
        000200                  X2.EXTF = BIT7          ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
        000100                  X2.BUFE = BIT6          ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
        000077                  X2.REV  = 000077        ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
        000007                  X2.UNIT = BIT2+BIT1+BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.

                                ;*
                                ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
                                ;(XST3)
                                ;-
        177400                  X3.MDE  = 177400        ;MICRO DIAGNOSTIC ERROR CODE
        000200                  X3.SPARE= BIT7          ;NOT USED BY TSV05
        000100                  X3.OPI  = BIT6          ;OPERATION INCOMPLETE
        000040                  X3.REV  = BIT5          ;REVERSE
        000020                  X3.TRF  = BIT4          ;TRANSPORT RESPONSE FAILURE
        000010                  X3.DCK  = BIT3          ;DENSITY CHECK
        000006                  X3.MBZ  =BIT2+BIT1      ;NOT USED ALWAYS 0
        000001                  X3.RIB  = BIT0          ;REVERSE INTO BOT

                                ;*
                                ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
                                ;(XST4)
                                ;-
        100000                  X4.HSP  = BIT15         ;HIGH SPEED
        040000                  X4.RCE  = BIT14         ;RETRY COUNT EXCEEDED
        020000                  X4.TSM  = BIT13         ;TRANSPORT SPECIAL MODE
        017400                  X4.MBZ  = BIT12+BIT11+BIT10+BIT9+BIT8   ;NOT USED ALWAYS 0
        000377                  X4.WRC  = 000377        ;WRITE RETRY COUNT FIELD

                                ;*
                                ;
                                ;TSSR TERMINATION CODES (BIT 0-2)
                                ;
                                ;-

                                        TSREJ=  3*2             ;COMMAND REJECTED
        000006                          UNREC=  6               ;UNRECOVERABLE ERROR
```

TSV05 REGISTER AND PACKET DEFINITIONS

```
                                ;*
                                ;
                                ;DEVICE REGISTER OFFSETS
                                ;
                                ;-

000000                          TSBA== 0
000000                          TSDB== 0                ;TSDB/TSBA REGISTER
000001                          TSBAH== 1
000001                          TSDBH== 1               ;TSDB/TSBA REGISTER HIGH BYTE
000002                          TSSR== 2                ;TSSR REGISTER
000003                          TSSRH== 3               ;TSSR REGISTER HIGH BYTE

                                ;*
                                ; TSDB ADDRESS BIT DEFINITIONS
                                ;-
000003                          A1716   = BIT1!BIT0     ;ADDRESS BITS 17:16 ARE IN 1:0

                                ;*
                                ; COMMAND DEFINITIONS
                                ;-
000017                          P.GETSTAT       = 17    ;GET STATUS
000013                          P.INIT          = 13    ;INITIALIZE
000012                          P.CONTROL       = 12    ;CONTROL COMMANDS
000011                          P.FORMAT        = 11    ;FORMAT
000010                          P.POSITION      = 10    ;POSITION
000006                          P.WRTSUB        = 6     ;SUBSYSTEM WRITE
000005                          P.WRITE         = 5     ;WRITE
000004                          P.WRTCHAR       = 4     ;WRITE CHARACTERISTICS
000001                          P.READ          = 1     ;READ

                                ;*
                                ; COMMAND PACKET HEADER WORD BIT DEFINITIONS
                                ;-
100000                                  P.ACK   = BIT15         ;BUFFER AVAIL FOR CONTROLLER
040000                                  P.CVC   = BIT14         ;CLEAR VOLUME CHECK
020000                                  P.OPP   = BIT13         ;REVERSE SEQUENCE OF DATA BITS
010000                                  P.SWB   = BIT12         ;SWAP BYTES IN MEMORY
007400                                  P.MODE  = BIT11!BIT10!BIT9!BIT8 ;EXTENDED COMMAND MODE FIELD
000200                                  P.IE    = BIT7          ;INTERRUPT ENABLE
000140                                  P.FMT= BIT6!BIT5        ;PACKET HEADER TYPE (ALWAYS=0)
000037                                  P.CMD   = 37            ;MAJOR COMMAND FIELD

                                ;*
                                ; CONTROL COMMAND MODE CODES
                                ;-
000000                                  PC.RELEASE      = 0*256.        ;RELEASE BUFFER
000400                                  PC.REWIND       = 1*256.        ;REWIND
001000                                  PC.NOOP         = 2*256.        ;NO-OP
002000                                  PC.IEREW        = 4*256.        ;REWIND IMMEDIATE INTERRUPT
002400                                  PC.ERASE        = 5*256.        ;SECURITY ERASE
```

TSV05 REGISTER AND PACKET DEFINITIONS

```
                        ;*
                        ; CONTROLLER RAM DEFINITIONS
                        ;-
000167                  RMCHBEG = 167           ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
000200                  RMCHEND = 200           ;CHARACTERISTICS IO DATA END RAM ADDRESS
000201                  RMPKTBEG= 201           ;COMMAND PACKET BEGIN RAM ADDRESS
000210                  RMPKTEND= 210           ;COMMAND PACKET END RAM ADDRESS
000215                  RMMSGBEG= 215           ;MESSAGE BUFFER BEGIN RAM ADDRESS
000234                  RMMSGEND= 234           ;MESSAGE BUFFER END RAM ADDRESS
                        ;*
                        ;
                        ;REGISTER DEFINITIONS IN THE MESSAGE BUFFER
                        ;
                        ;-

000006                  XST0==  6               ;EXTENDED STATUS REGISTER 0 (WORD 4)
000010                  XST1==  8.              ;EXTENDED STATUS REGISTER 1 (WORD 5)
000012                  XST2==  10.             ;EXTENDED STATUS REGISTER 2 (WORD 6)
000014                  XST3==  12.             ;EXTENDED STATUS REGISTER 3 (WORD 7)
000016                  XST4==  14.             ;EXTENDED STATUS REGISTER 4 (WORD 8)

                        ;*
                        ;
                        ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
                        ;
                        ;-

000002                          PKLOW   = 2             ;LOW ORDER CHARACTERISTIC DATA POINTER
000004                          PKHI    = 4             ;HIGH ORDER CHARACTERISTIC DATA POINTER
000006                          PKBCNT    6             ;NUMBER OF BYTES IN DATA PACKET

000010                          EXBCNT=10               ;NUMBER OF BYTES IN EXTENDED DATA PACKET

                        ;*
                        ;DATA PACKET OFFSETS FOR WRITE SUBSYTEM COMMAND
                        ;-
000000                          BSEL0   = 0             ;BYTE 0
000001                          BSEL1   = 1             ;BYTE 1
000002                          SEL2    = 2             ;WORD 2
000004                          SELDATA = 4             ;WORD 3
```

```
                                ;*
                                ;BSEL0 SELECT CODES FOR WRITE SUBSYSTEM COMMAND
                                ;-
000000                          PW.NOP          = 0             ;NO-OP
000001                          PW.RDRAM        = 1             ;READ RAM
000002                          PW.WTRAM        = 2             ;WRITE RAM
000003                          PW.RFIFO        = 3             ;READ FIFO
000004                          PW.WFIFO        = 4             ;WRITE FIFO
000005                          PW.RDSTAT       = 5             ;READ STATUS
000006                          PW.WCTL         = 6             ;WRITE TAPE CONTROL
000007                          PW.WFMT         = 7             ;WRITE TAPE FORMAT
000010                          PW.WMISC        = 10            ;WRITE MISCELLANEOUS
000011                          PW.WNPP         = 11            ;WRITE NPR CONTROL
000020                          PW.D22          = 20            ;DO MICROTEST 22
000021                          PW.D11          = 21            ;DO MICROTEST 11
000022                          PW.D13          = 22            ;DO MICROTEST 13
000023                          PW.NO1311       = 23            ;DISABLE MICROTEST 11 AND 13
000024                          PW.RDEXT        = 24            ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS

                                ;*
                                ;BSEL1 CODES FOR WRITE TAPE CONTROL
                                ;-
000200                          WC.IFAD         = BIT7          ;IFAD - FORMATTER ADDRESS
000100                          WC.IOTAD        = BIT6          ;ITADO  - TRANSPORT ADDRESS BIT 0
000040                          WC.I1TAD        = BIT5          ;ITAD1  - TRANSPORT ADDRESS BIT 1
000020                          WC.ISRESV       = BIT4          ;IRESV5 - RESERVED #5
000010                          WC.IREW         = BIT3          ;IREW   - REWIND
000004                          WC.IRWU         = BIT2          ;IRWU   - REWIND AND UNLOAD
000002                          WC.IFEN         = BIT1          ;IFEN   - FORMATTER ENABLE
000001                          WC.IGO          = BIT0          ;GO

                                ;*
                                ;BSEL1 CODES FOR WRITE FORMAT
                                ;-
000200                          WF.IHISP        = BIT7          ;IHISP  - HIGH SPEED
000100                          WF.IWRT         = BIT6          ;IWRT   - WRITE
000040                          WF.IREV         = BIT5          ;IREV   - REVERSE
000020                          WF.IWFM         = BIT4          ;IWFM   - WRITE FILE MARK
000010                          WF.IEDIT        = BIT3          ;IEDIT  - EDIT
000004                          WF.IERASE       = BIT2          ;IERASE - ERASE
000002                          WF.I3RESV       = BIT1          ;IRESV3 - RESERVED #3
000001                          WF.I4RESV       = BIT0          ;IRESV4 - RESERVED #4

                                ;*
                                ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
                                ;-
000200                          MS.EXT          = BIT7          ;INVERT SENSE OF EXTENDED FEATURES SWITCH
000020                          MS.RSFIFO       = BIT4          ;RESET FIFO AND INPUT PARITY ERRORR
000010                          MS.RSTAPE       = BIT3          ;RESET TAPE STATUS IN 2 FLIP-FLOPS
000006                          MS.ATTN         = BIT2!BIT1     ;ATTENTION TRIGGER FIELD
000001                          MS.RSD          = BIT0          ;RESET TIMER A,B THEN DELAY TIMES IN SEL2
```

TSV05 REGISTER AND PACKET DEFINITIONS

```
                                ;*
                                ; MS.ATTN SUBCODES
                                ;-
000000                                  MSA.NOP = 0*2           ;NO-OP (NOTHING TRIGGERED)
000002                                  MSA.VOL = 1*2           ;SIMULATE ON-LINE/OFF LINE TRANSISTION
000004                                  MSA.NRAM= 2*2           ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
000006                                  MSA.FRAM= 3*2           ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
                                ;*
                                ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
                                ;-
000200                          NP.IR           = BIT7          ;INTERRUPT REQUEST (0-1 TRANSITION)
000100                          NP.OUT          = BIT6          ;TAPE DATA DIRECTION OUT (0= IN)
000040                          NP.LOOP         = BIT5          ;ENABLE TRANSPORT LOOPBACK
000020                          NP.WRP          = BIT4          ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
                                ;*
                                ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
                                ;-

000200                          S2.DIM          = BIT7          ;WORD #9 BYTE 2 DATA IN MISS
000100                          S2.ILW          = BIT6          ;               ILW H
000040                          S2.OUTRDY       = BIT5          ;               OUT RDY H
000020                          S2.INRDY        = BIT4          ;               IN RDY H
000010                          S2.ATIMR        = BIT3          ;               TIMER A FLAG H
000004                          S2.BTIMR        = BIT2          ;               TIMER B FLAG H
000003                          S2.UNDEF        = BIT1+BIT0     ;(UNDEFINED)
100000                          S1.PARIN        = BIT15         ;WORD #8 BYTE 1 PARIN H
040000                          S1.I2RESV       = BIT14         ;               IRESV2
020000                          S1.I1RESV       = BIT13         ;               IRESV1
010000                          S1.IEOT         = BIT12         ;               IEOT L
004000                          S1.IIDENT       = BIT11         ;               IIDENT H
002000                          S1.ICER         = BIT10         ;               ICER H
001000                          S1.IFMK         = BIT9          ;               IFMK H
000400                          S1.IHER         = BIT8          ;               IHER H
000200                          S0.ISPEED       = BIT7          ;WORD #8 BYTE 0 ISPEED H
000100                          S0.IRDY         = BIT6          ;               IRDY L
000040                          S0.IONL         = BIT5          ;               IONL L
000020                          S0.ILDP         = BIT4          ;               ILDP L
000010                          S0.IDBY         = BIT3          ;               IDBY L
000004                          S0.IRWD         = BIT2          ;               IRWD L
000002                          S0.IFBY         = BIT1          ;               IFBY L
000001                          S0.IFPT         = BIT0          ;               IFPT L
```

```
                        .SBTTL  SPECIAL MACROS AND OPDEFS.

                ;+
                ;SAVE GENERAL REGS 1 TO 5
                ;-

                        .MACRO  SAVREG
                        JSR     R5,REGSAV
                        .ENDM

                ;+
                ; MACRO TO FORCE AN ERROR
                ;-
                .MACRO  FORCERROR       TAG,NOTSSR
                .NLIST
                .IIF NDF LISTALL, .NLIST
                .LIST
                .IF B NOTSSR
                        MOV     TSSR(R5),R1     ;READ TSSR
                .ENDC
                        MOV     FORCER,FORCER   ;IS FORCER SET? (LEAVE C BIT ALONE)
                        BNE     TAG             ;BR IF YES
                .NLIST
                .IIF NDF LISTALL, .LIST
                .LIST
                .ENDM

                ;+
                ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
                ;       WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
                ;       SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
                ;       FORCER TO 177777
                ;       TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
                ;-
                .MACRO  FORCEXIT        TAG
                .NLIST
                .IIF NDF LISTALL, .NLIST
                .LIST
                        MOV     FORCER,FORCER   ;IS FORCER NEGATIVE?
                        BMI     TAG             ;BR IF YES
                .NLIST
                .IIF NDF LISTALL, .LIST
                .LIST
                .ENDM
                ;+
                ; MACRO TO INCREMENT ERROR COUNTS
                ;-
                .MACRO  NEXT.ERRNO
                .NLIST
                ;;;.IIF NDF LISTALL, .NLIST
                        ERRNO=ERRNO+1
                ;;;.IIF NDF LISTALL, .LIST
                .LIST
                .ENDM
```

SPECIAL MACROS AND OPDEFS.

```
                                ;+
                                ;MACRO TO PERFORM XOR
                                ;-

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

000000                                  EN=0                    ; INITIALIZE ERROR NUMBER
                                        .SBTTL  FORCER  - FORCE ERROR FLAG


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

002174  000000                  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

GLOBAL DATA SECTION

```
                                        .SBTTL  GLOBAL DATA SECTION

                                ;++
                                ;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.
                                ;
002176  000000                  EPRTSW::        .WORD   0       ;PRINT SWITCH
002200  000000                  UNITN::         .WORD   0       ;UNIT # UNDER TEST.
002202  000000                  QVP::           .WORD   0       ;QUICK VERIFY FLAG.
002204  000000                  CSRADDR::       .WORD   0       ;ADDRESS OF CSR FOR CURRENT DEVICE
002206  000224                  IVEC::          .WORD   224     ;INTERRUPT VECTOR
002210  000200                  IPRI::          .WORD   PRI04   ;INTERRUPT PRIORITY.
002212  000000                  TSTCNT::        .WORD   0       ;NUMBER OF TESTS RUN IN THIS PASS
002214  000000                  LOOPCNT::       .WORD   0       ;REMAINING ITERATION COUNT FOR TEST
002216  000000                  DEVCNT::        .WORD   0       ;NUMBER OF DEVICE UNDER TEST
002220  000000                  FATFLG::        .WORD   0       ;SET IF FATAL ERROR IS DETECTED IN TEST
002222  000000                  INTRECV::       .WORD   0       ;SET IF TAPE INTERRUPT WAS RECEIVED
002224  000000                  EXTFEA::        .WORD   0       ;EXTENDED FEATURES SOFTWARE SW 0=OFF;1=ON
002226  000000                  REV::           .WORD   0       ;REV LEVEL (old microcode=1,new micro=2
002230  000000                  BENBSW::        .WORD   0       ;BUFFER ENABLE SWITCH SW 0=OFF;1=ON
002232  000000                  EXPD::          .WORD   0       ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
002234  000000                  RECV::          .WORD   0       ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
002236  000000                  ERRHI::         .WORD   0       ;HIGH ADDRESS MEMORY ERROR
002240  000000                  ERRLO::         .WORD   0       ;LOW ADDRESS MEMORY ERROR
002242                          RAMDATA::       .BLKW   16.     ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
002302  000000                  RAMSIZ::        .WORD   0       ;RAM DATA SIZE FOR PRAMPKT ROUTINE
002304  000000                  RCVHIADD::      .WORD   0       ;RECEIVED BUFFER HIGH ADDRESS
002306  000000                  RCVLOADD::      .WORD   0       ;RECEIVED BUFFER LOW ADDRESS
002310  000000                  COUNT::         .WORD   0       ;TEST COUNT PATTERN
002312  000000                  DATA::          .WORD   0       ;TEST DATA
002314  000000                  TSTFLAG::       .WORD   0       ;TEST FLAG WORD
002316  000000                  TSTPTR::        .WORD   0       ;TSTBLK POINTER
002320  000000                  PRMNO::         .WORD   0       ;PRINT ROUTINE TEMP
002322                          EXPMSG::        .BLKB   100.    ;EXPECTED MESSAGE BUFFER DATA
002466                          RECMSG::        .BLKB   100.    ;RECEIVED MESSAGE BUFFER DATA
002632                          TMPBFR::        .BLKB   80.     ;TEMPORARY STORAGE FOR PRINT
```

N3


```
                                        .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
                                ;
                                ;-

002752                          TSTBLK::
002752  000000                          .WORD   0               ;ALL ZEROS
002754  177777                          .WORD   177777          ;ALL ONES
002756  000001                          .WORD   BIT0            ;DATA FOR WALKING ONES
002760  000002                          .WORD   BIT1
002762  000004                          .WORD   BIT2
002764  000010                          .WORD   BIT3
002766  000020                          .WORD   BIT4
002770  000040                          .WORD   BIT5
002772  000100                          .WORD   BIT6
002774  000200                          .WORD   BIT7
002776  000400                          .WORD   BIT8
003000  001000                          .WORD   BIT9
003002  002000                          .WORD   BIT10
003004  004000                          .WORD   BIT11
003006  010000                          .WORD   BIT12
003010  020000                          .WORD   BIT13
003012  040000                          .WORD   BIT14
003014  100000                          .WORD   BIT15
003016  177776                          .WORD   ^CBIT0          ;DATA FOR WALKING ZEROS
003020  177775                          .WORD   ^CBIT1
003022  177773                          .WORD   ^CBIT2
003024  177767                          .WORD   ^CBIT3
003026  177757                          .WORD   ^CBIT4
003030  177737                          .WORD   ^CBIT5
003032  177677                          .WORD   ^CBIT6
003034  177577                          .WORD   ^CBIT7
003036  177377                          .WORD   ^CBIT8
003040  176777                          .WORD   ^CBIT9
003042  175777                          .WORD   ^CBIT10
003044  173777                          .WORD   ^CBIT11
003046  167777                          .WORD   ^CBIT12
003050  157777                          .WORD   ^CBIT13
003052  137777                          .WORD   ^CBIT14
003054  077777                          .WORD   ^CBIT15
003056  125252                          .WORD   125252          ;ALTERNATING ONES, ZEROS
003060  052525                          .WORD   052525          ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
003062                          TBLEND--.
```

GLOBAL ENVIRONMENT STORAGE

```
                                        .SBTTL  GLOBAL ENVIRONMENT STORAGE
                                ;
                                ;STORAGE FOR DEVICE REGISTERS
                                ;
003062  000000  100000  000000  DUMMY:  0,100000,0,0     ;DUMMY DEVICE REGISTERS...
003072  000000  000000  000000          0,0,0,0,0,0,0,0
                                                        ;...FOR MULTI-UNIT CHECKOUT.

003112  000000                  DUFLG::         .WORD   0               ;"DROPPED UNIT" FLAG.
                                                                        ;INHIBITS CODE IN "CLEAN-UP".
003114  000000                  NODEV::         .WORD   0               ;FLAG TO SAY NO DEVICE.

003116  000000                  TEMP1::         .WORD   0               ;SOME TEMP LOCATIONS.
003120  000000                  TEMP2::         .WORD   0
003122  000000                  XXCOMM::        .WORD   0               ;XXDP+ COMM BLOCK POINTER.
003124  000000                  FREE::          .WORD   0               ;1ST FREE MEMORY ADDRESS...
003126  000000                  FRESIZ::        .WORD   0               ;...AND SIZE (IN WORDS).
003130  000000                  FREEHI: .WORD   0               ;LAST WORD IN FREE SPACE
003132  000000                  KTFLG::         .WORD   0               ;KT11, MEM AVAIL FLAG -
                                                                        ;- .WORD        0 = <24K OR NO KT -
                                                                        ;- NZ = >24K AND KT.
003134  000000                  KTENABLE::      .WORD   0               ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
003136  000000                  NXMFLG::        .WORD   0               ;SET IF WE CAN TEST CLEARED OTHERWISE
003140  000000                  NXMLO::         .WORD   0               ;NXM LO ADDRESS BITS
003142  000000                  NXMHI::         .WORD   0               ;NXM HI ADDRESS BITS FOR DAL'S 16-21
003144  000000                  T23A::          .WORD   0               ;11/23A FLAG
003146  000000                  T23B::          .WORD   0               ;11/23B FLAG
003150  000000                  T38FLG::        .WORD   0               ;TEST 38 FLAG TO
003152  002000                  PST32W::        .WORD   2000            ;32W BLOCK ADDRESS FOR 32K START
003154  000000                  SIFLAG::        .WORD   0               ;
003156  000000                  BADDAT::        .WORD   0               ;ACTUAL DATA
003160  000000                  GDDAT::         .WORD   0               ;EXPECTED DATA
003162  000000                  LOOPFL::        .WORD   0
003164                          CTAB::                                  ;CONFIGURATION TABLES.
003164  000000                  CTABM::         .WORD   0               ;CONFIG WORK.
003166  000000                                  .WORD   0
003170  000000                                  .WORD   0
003172  000000                                  .WORD   0
003174  177777                                  .WORD   -1              ;END OF MEM TABLE.
003176                          CTABE::
                                ;ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
                                ;
                                ;       0       =       UNIT NOT TESTED
                                ;       100000  =       UNIT ONLINE, NO ERRORS
                                ;       10XXXX  =       UNIT ONLINE, ENCOUNTERED XXXX ERRORS
                                ;       160000  =       UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
                                ;       160001  =       UNIT DROPPED, NOT IDLE AT START
                                ;       14XXXX  =       UNIT DROPPED, ENCOUNTERED XXXX ERRORS
                                ;
003176                          ERTABL:         .BLKW   64.
003376  000000                  ERTABE:         .WORD   0

003400  000000                  SKIPT:  .WORD   0                ;1=SKIP SUBTEST 0=NO SKIP OF SUBTEST
```

GLOBAL TEXT MESSAGES

```
                                        .SBTTL  GLOBAL TEXT MESSAGES
                                ;**
                                ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
                                ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
                                ; MORE THAN ONE TEST.
                                ;--
                                ;*
                                ;NAMES OF DEVICES SUPPORTED
                                ;-
003402                                  DEVTYP  <TSV05>
003402                          L$DVTYP::
003402  124     123     126             .ASCIZ  /TSV05/
                                        .EVEN

                                ;*
                                ;TEST DESCRIPTION
                                ;-
003410                                  DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****>
003410                          L$DESC::
003410  052     052     052             .ASCIZ  /**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****/
                                        .EVEN

                                ;*
                                ;BIT TO ASCII CONVERSION FOR TSSR REGISTER
                                ;-
003502  003542  003545  003551  TSSRBIT::       .WORD   1$,2$,3$,4$,5$,6$,7$,8$
003522  003603  003607  003613                  .WORD   9$,10$,11$,12$,13$,14$,15$,16$
003542  123     103     000     1$:     .ASCIZ  'SC'
003545  102     111     105     2$:     .ASCIZ  'BIE'
003551  123     103     105     3$:     .ASCIZ  'SCE'
003555  122     115     122     4$:     .ASCIZ  'RMR'
003561  116     130     115     5$:     .ASCIZ  'NXM'
003565  116     102     101     6$:     .ASCIZ  'NBA'
003571  102     111     124     7$:     .ASCIZ  'BIT9'
003576  102     111     124     8$:     .ASCIZ  'BIT8'
003603  123     123     122     9$:     .ASCIZ  'SSR'
003607  117     106     114     10$:    .ASCIZ  'OFL'
003613  102     111     124     11$:    .ASCIZ  'BIT5'
003620  102     111     124     12$:    .ASCIZ  'BIT4'
003625  102     111     124     13$:    .ASCIZ  'BIT3'
003632  102     111     124     14$:    .ASCIZ  'BIT2'
003637  102     111     124     15$:    .ASCIZ  'BIT1'
003644  102     111     124     16$:    .ASCIZ  'BIT0'
                                .EVEN
003652  124     123     123     SFIERR: .ASCIZ  'TSSR ERROR AFTER SOFT INIT'
003705  124     123     123     SFHERR: .ASCIZ  'TSSR ERROR AFTER BUS RESET'
003740  040     040     116     NXR:    .ASCIZ  /  NON-EXISTANT DEVICE REGISTER/
003777  045     101     040     NXRX:   .ASCIZ  /%A  ADDRESS: %O6/
004020  045     101     040     TSSX:   .ASCII  /%A  TSBA,TSSR EXP'D: %O6%A,%O6%N/
004060  045     101     040             .ASCIZ  /%A  TSBA,TSSR REC'D: %O6%A,%O6/
004117  045     116     045     FUSI:   .ASCII  /%N%A/
004123  040     040     125     USI:    .ASCIZ  /  UNEXPECTED INTERRUPT/
004152  040     040     111     NSI:    .ASCIZ  /  INTERRUPT EXPECTED, NOT RECEIVED/
004215  045     116     045     FNOINTR:        .ASCII  /%N%A/
004221  040     040     116     NOINTR: .ASCIZ  /  NO INTERRUPT WAS GENERATED/
004256  040     040     111     IFAULT: .ASCIZ  /  INTERRUPT FAULT/
004300  045     101     040     INTX:   .ASCIZ  /%A  CPU PC: %O6%A  TSBA: %O6/
```

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               000  NUL:    .ASCIZ  //
    638 004530      045      116      000  NULCR:  .ASCIZ  /≤N/
    639 004533      045      101      040  EXPGOT: .ASCIZ  /≤A EXP'D: ≤O6≤A, REC'D: ≤O6/
    640 004567      045      116      045  EXPGT2: .ASCIZ  /≤N≤A EXP'D: ≤O6≤A, ≤O6≤N≤A REC'D: ≤O≤A, ≤O6/
    641 004643      045      101      040  DUAD12: .ASCIZ  /≤A REG(W) WRITTEN TO: ≤O6≤A REG(R) READ; EXP'D: ≤O6≤A, REC'D: ≤O6/
    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                                              .EVEN
    653                                              .SBTTL  GLOBAL ERROR REPORT SECTION
    654
    655                                      ;++
    656                                      ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
    657                                      ; CALLS THAT ARE USED IN MORE THAN ONE TEST.
    658                                      ; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
    659                                      ;--
    660 005736                                       BGNMSG  NXRERR          ;NON-EXISTANT DEVICE REGISTER.
        005736                               NXRERR::
    661 005736                                       PRINTX  #NXRX,NODEV     ;NODEV = NEXM ADDRESS.
        005736  013746  003114                       MOV     NODEV,-(SP)
        005742  012746  003777                       MOV     #NXRX,-(SP)
        005746  012746  000002                       MOV     #2,-(SP)
        005752  010600                               MOV     SP,R0
        005754  104415                               TRAP    C$PNTX
        005756  062706  000006                       ADD     #6,SP
    662 005762  004737  005770                       JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
    663 005766                                       ENDMSG
        005766                               L10002:
        005766  104423                               TRAP    C$MSG
    664                                      ;
    665                                      ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
    666                                      ; TO ANY OF THE ABOVE ERROR SIGNATURES.
    667                                      ;
    668 005770  005727                       EXTEND: TST     (PC)+
    669 005772  000000                       EXTA:   0                       ; 0 = NO EXTENSION.
    670 005774  001402                               BEQ     1$
    671 005776  004777  177770                       JSR     PC,@EXTA        ; APPEND EXTENSION TEXT.
    672 006002                               1$:     PRINTX  #NULCR          ; PRINT A BLANK LINE
        006002  012746  004530                       MOV     #NULCR,-(SP)
        006006  012746  000001                       MOV     #1,-(SP)
        006012  010600                               MOV     SP,R0
        006014  104415                               TRAP    C$PNTX
        006016  062706  000004                       ADD     #4,SP
    673 006022  000207                               RTS     PC
```

```
                                        .SBTTL  PRITSSR - PRINT TSSR CONTENTS

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

006024                          PRITSSR:
006024                                  SAVREG                  ;SAVE GENERAL REGISTERS
006030  010104                          MOV     R1,R4           ;SAVE THE TSSR CONTENTS
006032                                  PRINTB  #TSSRFOR,R4     ;PRINT THE CONTENTS OF TSSR
006032  010446                          MOV     R4,-(SP)
006034  012746  006415                  MOV     #TSSRFOR,-(SP)
006040  012746  000002                  MOV     #2,-(SP)
006044  010600                          MOV     SP,R0
006046  104414                          TRAP    C$PNTB
006050  062706  000006                  ADD     #6,SP
006054  010400                          MOV     R4,R0           ;GET TSSR BACK FOR CHKAMB
006056  004737  016124                  JSR     PC,CHKAMB       ;ARE CONTENTS AMBIGUOUS ?
006062  103410                          BCS     5$              ;BRANCH IF NOT
006064                                  PRINTX  #AMBTSSR        ;SHOW CONTENTS ARE AMBIGUOUS
006064  012746  006635                  MOV     #AMBTSSR,-(SP)
006070  012746  000001                  MOV     #1,-(SP)
006074  010600                          MOV     SP,R0
006076  104415                          TRAP    C$PNTX
006100  062706  000004                  ADD     #4,SP
006104  010403                  5$:     MOV     R4,R3           ;CONTENTS OF TSSR
006106  042703  001476                  BIC     #HIADDR!FATERR!TERCLS,R3        ;CLEAR ALL MULTIPLE BIT FIELDS
006112  001434                          BEQ     20$             ;NO BITS ARE SET
006114  012702  002632                  MOV     #TMPBFR,R2      ;TEMPORARY ASCII BUFFER
006120  012701  003502                  MOV     #TSSRBIT,R1     ;ASCII EQUIVALENT OF BITS
006124  005703                  10$:    TST     R3              ;REMAINING BITS TO CONVERT
006126  001413                          BEQ     15$             ;BRANCH WHEN ALL ARE DONE
006130  000241                          CLC                     ;CLEAR CARRY FOR SHIFT
006132  006103                          ROL     R3              ;SHIFT NEXT BIT TO CARRY
006134  103006                          BCC     13$             ;BRANCH IF BIT NOT SET
006136  011100                          MOV     (R1),R0         ;POINTER TO BIT DEFINITION
006140  112022                  11$:    MOVB    (R0)+,(R2)+     ;MOVE ASCIZ TO BUFFER
006142  001376                          BNE     11$             ;MOVE ALL BITS
006144  112762  000054  177777          MOVB    #',,-1(R2)      ;INSERT A COMMA TO TERMINATE
006152  005721                  13$:    TST     (R1)+           ;POINT TO NEXT DESCRIPTION
006154  000763                          BR      10$             ;GET THE REMAINING BITS
006156  105042                  15$:    CLRB    -(R2)           ;TERMINATE THE LINE
006160                                  PRINTX  #TSSDEF,#TMPBFR ;PRINT THE BIT DEFINITIONS
006160  012746  002632                  MOV     #TMPBFR,-(SP)
006164  012746  006606                  MOV     #TSSDEF,-(SP)
```

F4

PRITSSR - PRINT TSSR CONTENTS

```
        006170  012746  000002                  MOV     #2.,-(SP)
        006174  010600                           MOV     SP,R0
        006176  104415                           TRAP    C$PNTX
        006200  062706  000006                  ADD     #6.SP
   719
   720  006204  010403                  20$:    MOV     R4,R3           ;GET THE TSSR CONTENTS
   721  006206  042703  177761                  BIC     #^CTERCLS,R3    ;CLEAR ALL BUT TERMINATION
   722  006212  016303  006676                  MOV     TCOCOD(R3),R3   ;GET THE TERMINATION CODE MEANING
   723  006216                                   PRINTX  #TCOASC,R3      ;PRINT THE TERMINATION CODE
        006216  010346                           MOV     R3,-(SP)
        006220  012746  006476                   MOV     #TCOASC,-(SP)
        006224  012746  000002                   MOV     #2.,-(SP)
        006230  010600                           MOV     SP,R0
        006232  104415                           TRAP    C$PNTX
        006234  062706  000006                   ADD     #6.SP
   724  006240  010403                           MOV     R4,R3           ;TSSR CONTENTS AGAIN
   725  006242  042703  177717                  BIC     #^CFATERR,R3    ;CLEAR ALL BUT FATAL TERMINATION
   726  006246  001416                           BEQ     25$             ;DON'T PRINT IF ZERO
   727  006250  006203                           ASR     R3
   728  006252  006203                           ASR     R3
   729  006254  006203                           ASR     R3              ;ALINE TERMINATION CODE FOR INDEX
   730  006256  016303  007236                  MOV     TSFCOD(R3),R3   ;GET THE FATAL TERMINATION CODE
   731  006262                                   PRINTX  #TFCASC,R3      ;PRINT THE FATAL TERMINATION CODE
        006262  010346                           MOV     R3,-(SP)
        006264  012746  006537                   MOV     #TFCASC,-(SP)
        006270  012746  000002                   MOV     #2.,-(SP)
        006274  010600                           MOV     SP,R0
        006276  104415                           TRAP    C$PNTX
        006300  062706  000006                   ADD     #6,SP
   732  006304  042704  176377          25$:    BIC     #^CHIADDR,R4    ;CLEAR ALL BUT EXTENDED ADDRESS
   733  006310  001411                           BEQ     30$             ;DON'T PRINT IF ZERO
   734  006312                                   PRINTX  #TEXASC,R4      ;PRINT THE EXTENDED ADDRESS BITS
        006312  010446                           MOV     R4,-(SP)
        006314  012746  006435                   MOV     #TEXASC,-(SP)
        006320  012746  000002                   MOV     #2.,-(SP)
        006324  010600                           MOV     SP,R0
        006326  104415                           TRAP    C$PNTX
        006330  062706  000006                   ADD     #6,SP
   735  006334  013703  002176          30$:    MOV     EPRTSW,R3               ;PRINT MEAASGE BUFFER ADDRESS
   736  006340                                   PRINTX  R3                      ;PRINT PROPER MESSAGE
        006340  010346                           MOV     R3,-(SP)
        006342  012746  000001                   MOV     #1.,-(SP)
        006346  010600                           MOV     SP,R0
        006350  104415                           TRAP    C$PNTX
        006352  062706  000004                   ADD     #4,SP
   737  006356  000207                           RTS     PC              ;RETURN TO CALLER
```

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 = %O6'
    757 006435      045     116     045 TEXASC:         .ASCIZ  '%N%A Extended Address Bits = %O6'
    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
```

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

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

007370                          PRIPKT::
007370                                  SAVREG                          ;SAVE THE REGISTERS
007374  010005                          MOV     R0,R5                   ;SAVE NO. OF WORDS IN PACKET
007376  005737  003134                  TST     KTENABLE                ;ABOVE 28K UNDER TEST?
007402  001001                          BNE     10$                     ;BR IF YES
007404  005003                          CLR     R3                      ;SET HIGH ORDER ADDRESS TO 0
007406  010301                  10$:    MOV     R3,R1                   ;COPY HIGH ORDER ADDRESS
007410  010400                          MOV     R4,R0                   ;GET LOWER ADDRESS
007412  006100                          ROL     R0                      ;SHIFT BIT 15 INTO C BIT
007414  006101                          ROL     R1                      ;AND INTO HIGH ORDER.
007416                                  PRINTB  #PKTADD,R1,R4           ;PRINT PACKET ADDRESS
007416  010446                          MOV     R4,-(SP)
007420  010146                          MOV     R1,-(SP)
007422  012746  007554                  MOV     #PKTADD,-(SP)
007426  012746  000003                  MOV     #3,-(SP)
007432  010600                          MOV     SP,R0
007434  104414                          TRAP    C$PNTB
007436  062706  000010                  ADD     #10,SP
007442  010300                  15$:    MOV     R3,R0                   ;GET HIGH ORDER ADDRESS
007444  001404                          BEQ     20$                     ;BR IF NOT ABOVE 28K.
007446  010401                          MOV     R4,R1                   ;GET LOW ORDER ADDRESS
007450  004737  017376                  JSR     PC,SETMAP               ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
007454  010004                          MOV     R0,R4                   ;GET RETURNED PAR6 ADDRESS BIAS
007456  005001                  20$:    CLR     R1                      ;SAVE WORD NUMBER
007460  012402                  25$:    MOV     (R4)+,R2                ;GET PACKET CONTENTS
007462                                  PRINTB  #PKTFRM,R1,R2           ;PRINT THE DATA
007462  010246                          MOV     R2,-(SP)
007464  010146                          MOV     R1,-(SP)
007466  012746  007516                  MOV     #PKTFRM,-(SP)
007472  012746  000003                  MOV     #3,-(SP)
007476  010600                          MOV     SP,R0
007500  104414                          TRAP    C$PNTB
007502  062706  000010                  ADD     #10,SP
007506  005201                          INC     R1                      ;NEXT WORD NUMBER
007510  020105                          CMP     R1,R5                   ;DONE ALL PACKET WORDS?
007512  002762                          BLT     25$                     ;LOOP TILL ALL DONE
007514  000207                          RTS     PC                      ;RETURN

007516      045     116     045 PKTFRM: .ASCIZ  '%N%A Packet Word #%D1%A = %O6'
007554      045     116     045 PKTADD: .ASCIZ  '%N%A Packet Address = %O1%O5'
                                        .EVEN
```

```
                                        .SBTTL  PRIBXOR - PRINT EXPD, RECV AND XOR BYTE

                                ;+
                                ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA
                                ;       R2      EXPECTED DATA
                                ;
                                ;OUTPUT:
                                ;
                                ;       R0      XOR OF EXPECTED/RECEIVED DATA
                                ;-
007612                          PRIBXOR::
007612                                  SAVREG                          ;SAVE THE REGISTERS
007616  010203                          MOV     R2,R3                   ;EXPECTED DATA
007620                                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007630  012700  177400                  MOV     #^C<377>,R0             ;BYTE MASK
007634  040001                          BIC     R0,R1                   ;SAVE LOW BYTE RECV
007636  040002                          BIC     R0,R2                   ;SAVE LOW BYTE EXPD
007640  040003                          BIC     R0,R3                   ;SAVE LOW BYTE XOR
007642                                  PRINTB  #XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
007642  010346                          MOV     R3,-(SP)
007644  010146                          MOV     R1,-(SP)
007646  010246                          MOV     R2,-(SP)
007650  012746  007674                  MOV     #XORBFOR,-(SP)
007654  012746  000004                  MOV     #4,-(SP)
007660  010600                          MOV     SP,R0
007662  104414                          TRAP    C$PNTB
007664  062706  000012                  ADD     #12,SP
007670  010300                          MOV     R3,R0                   ;R0 HAS XOR ON RETURN
007672  000207                          RTS     PC                      ;RETURN TO CALLER

007674      045     116     045 XORBFOR:        .ASCIZ  '%N%A EXPD: %03%A RECV: %03%A XOR: %03'
                                        .EVEN
                                        .SBTTL  PRIXOR  - PRINT EXPD, RECV AND XOR
                                ;+
                                ;
                                ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA
                                ;       R2      EXPECTED DATA
                                ;
                                ;OUTPUT:
                                ;
                                ;       R0      XOR OF EXPECTED/RECEIVED DATA
                                ;-
007742                          PRIXOR::
007742                                  SAVREG                          ;SAVE THE REGISTERS
007746  010203                          MOV     R2,R3                   ;EXPECTED DATA
007750                                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007760                                  PRINTB  #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
```

J4

SEQ 0048

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    C$PNTB
        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
```

```
                                        .SBTTL  PRIEQU    PRINT BIT NUMBERS AS ASCII EQUIVALENT

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

010060                          PRIEQU:
010060                                  SAVREG                          ;SAVE THE REGISTERS
010064  000207                          RTS     PC                      ;RETURN TO CALLER

                                        .SBTTL  PRIRAM  - PRINT RAM ADDRESS
                                ;*
                                ;
                                ;PRINT CONTROLLER RAM ADDRESS.
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                                ;
                                ;INPUTS:
                                ;
                                ;       R4      RAM ADDRESS
                                ;
                                ;-
010066                          PRIRAM:
010066                                  SAVREG                  ;SAVE R1 R5 UNTIL NEXT RETURN
010072                                  PRINTB  #RAMFOR,R4      ;PRINT RAM ADDRESS IN ERROR
010072  010446                          MOV     R4,-(SP)
010074  012746  010116                  MOV     #RAMFOR,-(SP)
010100  012746  000002                  MOV     #2,-(SP)
010104  010600                          MOV     SP,R0
010106  104414                          TRAP    C$PNTB
010110  062706  000006                  ADD     #6,SP
010114  000207                          RTS     PC              ;RETURN

010116      045     116     045 RAMFOR: .ASCIZ  '%N%A CONTROLLER RAM ADDRESS = %O6'
                                        .EVEN
```

L4


```
                                        .SBTTL  PRIADD  - PRINT MEMORY ERROR ADDRESS
                                ;+
                                ;
                                ;PRINT MEMORY ADDRESS
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                                ;
                                ; IMPLICIT INPUTS
                                ;
                                ;       ERRHI   - HIGH ORDER ADDRESS
                                ;       ERRLO   - LOW ORDER ADDRESS
                                ;
                                ;-
010160                          PRIADD:
010160                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
010164  013700  002236                  MOV     ERRHI,R0                ;GET HIGH ADDRESSS
010170  013701  002240                  MOV     ERRLO,R1                ;GET LOW ADDRESS
010174  010102                          MOV     R1,R2                   ;COPY LOW ADDRESS
010176  006101                          ROL     R1                      ;SHIFT BIT 15 TO C BIT
010200  006100                          ROL     R0                      ;SHIFT INTO HIGH ORDER
010202                                  PRINTB  #PRIA0,R0,R2            ;PRINT MEMORY ADDRESS IN ERROR
010202  010246                          MOV     R2,-(SP)
010204  010046                          MOV     R0,-(SP)
010206  012746  010230                  MOV     #PRIA0,-(SP)
010212  012746  000003                  MOV     #3,-(SP)
010216  010600                          MOV     SP,R0
010220  104414                          TRAP    C$PNTB
010222  062706  000010                  ADD     #10,SP
010226  000207                          RTS     PC                      ;RETURN

010230      045     116     045 PRIA0:  .ASCIZ  '%N%A MEMORY ERROR ADDRESS = %O1%O5'
                                        .EVEN

                                        .SBTTL  PRITADD - PRINT MEMORY TEST ADDRESS
                                ;+
                                ;
                                ;PRINT MEMORY ADDRESS
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                                ;
                                ; IMPLICIT INPUTS
                                ;
                                ;       ERRHI   - HIGH ORDER ADDRESS
                                ;       ERRLO   - LOW ORDER ADDRESS
                                ;
                                ;-
010274                          PRITADD:
010274                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
010300  013702  002236                  MOV     ERRHI,R2                ;GET HIGH ADDRESSS
010304  013701  002240                  MOV     ERRLO,R1                ;GET LOW ADDRESS
                                        ;MOV    R1,R2                   ;COPY LOW ADDRESS
                                        ;ROL    R1                      ;SHIFT BIT 15 TO C BIT
                                        ;ROL    R0                      ;SHIFT INTO HIGH ORDER
010310                                  PRINTB  #PRIT0,R1               ;PRINT MEMORY ADDRESS LOW IN ERROR
010310  010146                          MOV     R1,-(SP)
010312  012746  010356                  MOV     #PRIT0,-(SP)
010316  012746  000002                  MOV     #2,-(SP)
010322  010600                          MOV     SP,R0
010324  104414                          TRAP    C$PNTB
```

M4

PRITADD - PRINT MEMORY TEST ADDRESS

```
        010326  062706  000006                  ADD     #6,SP
    961 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
        010350  062706  000006                  ADD     #6,SP
    962 010354  000207                          RTS     PC              ;RETURN
    963
    964 010356     045     116     045  PRIT0:  .ASCIZ  '#N#A MEMORY TEST ADDRESS LOW = #O6'
    965 010421     045     116     045  PRIT1:  .ASCIZ  '#N#A MEMORY TEST ADDRESS HIGH = #O6'
    966                                         .EVEN
```

```
                        .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:
                        ;
                        ;
                        ;-

010466                  SPACE::
010466                          SAVREG                          ;SAVE THE GENERAL REGISTERS
010472  012737  000764  010660          MOV     #500.,SDELAY    ;SET UP DELAY
010500  012737  140010  010650          MOV     #140010,80$     ;SET UP COMMAND, SPACE FORWARD
010506  005703                          TST     R3              ;CHECK FOR DIRECTION
010510  100403                          BMI     5$              ;BR, IF REVERSE INDICATED
010512  010337  010652                  MOV     R3,90$          ;LOAD UP NUMBER OF RECORDS TO SPACE
010516  000407                          BR      10$             ;GO DO COMMAND
010520  042703  100000          5$:     BIC     #BIT15,R3       ;CLEAR DIRECTION BIT
010524  010337  010652                  MOV     R3,90$          ;LOAD UP NUMBER OF RECORDS TO SPACE
010530  052737  000400  010650          BIS     #BIT8,80$       ;SET REVERSE BIT IN COMMAND PACKET
010536  012704  010650          10$:    MOV     #80$,R4         ;SET UP R4 WITH PACKET ADDRESS
010542  010465  000000                  MOV     R4,TSDB(R5)     ;SEND OUT COMMAND
010546  004737  016330          15$:    JSR     PC,WAITF        ;WAIT FOR SSR
010552  103420                          BCS     20$             ;BR, IF SSR IS SET AND OK
010554                                  DELAY   250             ;DELAY ABOUT .25 SECONDS
010554  012727  000250                  MOV     #250,(PC)+
010560  000000                          .WORD   0
010562  013727  002116                  MOV     L$DLY,(PC)+
010566  000000                          .WORD   0
010570  005367  177772                  DEC     -6(PC)
010574  001375                          BNE     .-4
```

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

```
        010576  005367  177756                  DEC     -22(PC)
        010602  001367                          BNE     .-20
  1019  010604  005337  010660                  DEC     $DELAY                  ;BUMP DELAY COUNTER DOWN
  1020  010610  001356                          BNE     15$                     ;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$:    CLC                             ;CARRY CLEAR = ERROR
  1030  010640                          70$:
  1031  010640  010400                          MOV     R4,R0                   ;PASS PACKET ADDRESS
  1032  010642  000207                          RTS     PC                      ;RETURN
  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
```

WRTCHR - WRITE CHARACTERISTICS COMMAND

```
                                ;+
                                ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
                                ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
                                ;
                                ;INPUT:
                                ;       R4      ADDRESS OF PACKET FROM TEST
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;       REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
                                ;
                                ;OUTPUT:
                                ;       R0      TSSR CONTENTS
                                ;       CARRY   SET - WRITE CHARACTERISTICS COMMAND OK
                                ;               CLR - WRITE CHARACTERISTICS FAILED
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
                                ;       SOFTWARE SWITCHES SET AS FOLLOWS:
                                ;               EXTFEA = EXTENDED FEATURES PRESENT
                                ;               BENBSW = BUFFER ENABLE SWITCH ON OR OFF
                                ;
                                ;SIDE EFFECTS:
                                ;-
010662                          WRTCHR::
010662                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
010666  005037  002230                  CLR     BENBSW          ;CLEAR BUFFER ENABLE SWITCH
010672  005037  002224                  CLR     EXTFEA                  ;CLEAR EXTENDED FEATURES SW SWITCH
010676  010465  000000          10$:    MOV     R4,TSDB(R5)             ;SEND OUT COMMAND
010702  004737  016416                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
010706  103401                          BCS     20$                     ;BR, IF SSR IS SET AND OK
010710  000455                          BR      60$                     ;BR IF TROUBLE CARRY = CLEAR
010712  016501  000002          20$:    MOV     TSSR(R5),R1             ;READ TSSR
010716  012702  000200                  MOV     #SSR,R2                 ;SET UP EXPECTED
010722  032701  000100                  BIT     #OFL,R1                 ;WAS OFF LINE SET IN TSSR
010726  001402                          BEQ     25$                     ;BR, IF NO OFL SET
010730  052702  000100                  BIS     #OFL,R2                 ;MAKE THEM LOOK ALIKE
010734  020201          25$:    CMP     R2,R1                   ;ARE THEY OK
010736  001401                          BEQ     40$                     ;BR, IF EQUAL = OK
010740  000441                          BR      60$                     ;TROUBLE EXIT
010742  062704  000010          40$:    ADD     #8.,R4                  ;POINT TO WRT CHARA DATA PACKET
010746  011403                          MOV     (R4),R3                 ;GET ADDRESS OF MESSAGE BUFFER
010750  032763  000200  000012          BIT     #X2.EXTF,XST2(R3)       ;EXTENDED FEATURES BIT SET?
010756  001402                          BEQ     45$                     ;BR IF NO
010760  005237  002224                  INC     EXTFEA                  ;SET EXTENDED FEATURES SW SWITCH
010764                          45$:
010764  032763  000100  000012          BIT     #X2.BUFE,XST2(R3)       ;BUFFER ENABLE SWITCH SET
010772  001402                          BEQ     50$                     ;BR, IF SWITCH NOT SET
010774  005237  002230                  INC     BENBSW                  ;SET SOFTWARE SWITCH FOR ENABLED
011000                          50$:
011000  016337  000012  002226          MOV     XST2(R3),REV            ;MICROCODE REV LEVEL
011006  042737  017700  002226          BIC     #17700,REV              ;CLEAR UNWANTED BITS
011014  022737  000001  002226          CMP     #1,REV                  ;IS IT A NEW MICROCODE
011022  001406                          BEQ     55$                     ;NO BR
011024  012737  000001  002224          MOV     #1,EXTFEA               ;ALWAY EXTENDED FEATURE FOR NEW
                                                                        ;MICROCODE
011032  052763  000200  000012          BIS     #X2.EXTF,XST2(R3)       ;EXTENDED FEATURE ALWAYS SET IN
                                                                        ;MICROCODE
```

WRTCHR - WRITE CHARACTERISTICS COMMAND

```
1109 011040  000261                  55$:   SEC                          ;SET CARRY NO TROUBLE
1110 011042  000401                         BR      70$                   ;EXIT
1111 011044  000241                  60$:   CLC                          ;CARRY CLEAR = ERROR
1112 011046  016500  000002          70$:   MOV     TSSR(R5),R0           ;RETURN TSSR CONTENTS
1113 011052  000207                         RTS     PC                    ;RETURN
```

```
                                        .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::
011054                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011054  012704  011150                  MOV     #RWPACK,R4              ;GET PACKET ADDRESS
011060  010465  000000                  MOV     R4,TSDB(R5)             ;SEND PACKET ADDRESS TO EXECUTE
011064  012703  000550                  MOV     #360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
011070  004737  016330          10$:    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
011074  103417                          BCS     20$                     ;LEAVE WHEN SSR IS SET
011076                                  DELAY   250.                    ;WAIT FOR .25 SECONDS
011076  012727  000372                  MOV     #250.,(PC)+
011102  000000                          .WORD   0
011104  013727  002116                  MOV     L$DLY,(PC)+
011110  000000                          .WORD   0
011112  005367  177772                  DEC     -6(PC)
011116  001375                          BNE     .-4
011120  005367  177756                  DEC     -22(PC)
011124  001367                          BNE     .-20
011126  005303                          DEC     R3                      ;BUMP COUNTER DOWN
011130  001357                          BNE     10$                     ;KEEP GOING
011132  000241                          CLC                             ;CLEAR CARRY TO SET ERROR
011134  010400                  20$:    MOV     R4,R0                   ;PASS THE PACKET ADDRESS
011136  000207                          RTS     PC                      ;RETURN
                                
011150                                  .=<.+10>&177770
011150                          RWPACK:
011150  102010                          .WORD   102010                  ;POSTION COMMAND (REWIND)
011152  000000                          .WORD   0                       ;NOT USED
```

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

011154                  CKRAM::
011154                          SAVREG                          ;SAVE THE GENERAL REGISTERS
011160  012701  002242          MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
011164  012702  000201          MOV     #RMPKTBEG,R2            ;BYTE ADDRESS OF FIRST RAM DATA
011170  005003                  CLR     R3                      ;CLEAR THE ERROR FLAG
011172  004737  016416          JSR     PC,CHKTSSR              ;WAIT FOR SSR
011176  112765  000000  000000  MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
011204  004737  016416  10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011210  010265  000000          MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
011214  004737  016416          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011220  116511  000000          MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
011224  122124                  CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
011226  001401                  BEQ     20$                     ;BRANCH IF OK
011230  005203                  INC     R3                      ;SET ERROR FLAG
011232  005202          20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
011234  020227  000210          CMP     R2,#RMPKTEND            ;REACHED END YET ?
011240  003761                  BLE     10$                     ;BRANCH TILL ALL READ
011242  005703                  TST     R3                      ;WAS AN ERROR FOUND ?
011244  001402                  BEQ     30$                     ;BRANCH IF NOT
011246  000241                  CLC                             ;CLEAR CARRY TO SHOW ERROR
011250  000401                  BR      50$                     ;AND EXIT
011252  000261          30$:    SEC                             ;SHOW GOOD COMPARE
011254  012737  000010  002302  50$:    MOV     #8.,RAMSIZ      ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
011262  000207                  RTS     PC                      ;RETURN
```

```
                                .SBTTL  CKRAM2  - COMPARE RAM TO I/O CHARACTERISTICS DATA
                              ;*
                              ;
                              ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
                              ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
                              ;
                              ;INPUT:
                              ;
                              ;       R4      ADDRESS OF THE CHARACTERISTICS DATA
                              ;       R5      FIRST DEVICE UNIBUS ADDRESS
                              ;
                              ;OUTPUT:
                              ;
                              ;       CARRY   SET - RAM MATCHES PACKET
                              ;               CLR - RAM DOES NOT MATCH PACKET
                              ;
                              ;IMPLICIT OUTPUT:
                              ;
                              ;       THE TABLE RAMDATA IS FILLED WITH THE
                              ;       DATA HELD IN RAM.
                              ;       RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
                              ;
                              ;SIDE EFFECTS:
                              ;
                              ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
                              ;-
011264                        CKRAM2::
011264                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
011270  012701  002242                  MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
011274  012702  000167                  MOV     #RMCHBEG,R2             ;BYTE ADDRESS OF FIRST RAM DATA
011300  005003                          CLR     R3                      ;CLEAR THE ERROR FLAG
011302  004737  016416                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
011306  112765  000000  000000          MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
011314  004737  016416          10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011320  010265  000000                  MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
011324  004737  016416                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011330  116511  000000                  MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
011334  122124                          CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
011336  001401                          BEQ     20$                     ;BRANCH IF OK
011340  005203                          INC     R3                      ;SET ERROR FLAG
011342  005202                  20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
011344  012737  000010  002302          MOV     #8.,RAMSIZ              ;ASSUME EXTFEA NOT SET
011352  005737  002224                  TST     EXTFEA                  ;IS THE SOFTWARE EXTENDED FEATURES SET
011356  001407                          BEQ     25$                     ;BR, IF NOT SET
011360  012737  000012  002302          MOV     #10.,RAMSIZ             ;SET RAMSIZ FOR EXTEND FEATURES
011366  020227  000200                  CMP     R2,#RMCHEND             ;AT END OF EXTENDED BUFFER
011372  003750                          BLE     10$                     ;BR, IF NOT AT END YET
011374  000403                          BR      27$                     ;AT END BRANCH
011376  020227  000176          25$:    CMP     R2,#RMCHEND-2           ;REACHED END YET ?
011402  003744                          BLE     10$                     ;BRANCH TILL ALL READ
011404  005703                  27$:    TST     R3                      ;WAS AN ERROR FOUND ?
011406  001402                          BEQ     30$                     ;BRANCH IF NOT
011410  000241                          CLC                             ;CLEAR CARRY TO SHOW ERROR
011412  000401                          BR      50$                     ;AND EXIT
011414  000261                  30$:    SEC                             ;SHOW GOOD COMPARE
011416  000207                  50$:    RTS     PC                      ;RETURN
```

```
                                        .SBTTL  CKMSG   - COMPARE WRITE CHAR. MESSAGE BUFFERS
                                ;*
                                ;
                                ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
                                ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
                                ;ERROR PRINT ROUTINES.
                                ;
                                ;INPUT:
                                ;
                                ;       R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
                                ;       R2      EXPD MESSAGE BUFFER ADDRESS
                                ;OUTPUT:
                                ;
                                ;       CARRY   SET - MESSAGE BUFFERS MATCH
                                ;               CLR -MESSAGE BUFFERS DON'T MATCH
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       EXPMSG          BUFFER IS SET TO EXPD DATA
                                ;       RECMSG          BUFFER IS SET TO RECV DATA
                                ;       RCVHIADD        SET TO HIGH ORDER ADDRESS OF RECV
                                ;       RCVLOADD        SET TO LOW ORDER ADDRESS OF RECV
                                ;
                                ;-
011420                          CKMSG::
011420                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011424  010037  002304                  MOV     R0,RCVHIADD             ;SAVE RECV HIGH ADDRESS
011430  010137  002306                  MOV     R1,RCVLOAD              ;SAVE RECV LOW ADDRESS
011434  005737  003134                  TST     KTENABLE                ;TESTING ABOVE 28K?
011440  001403                          BEQ     10$                     ;BR IF NO
011442  004737  017376                  JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
011446  010001                          MOV     R0,R1                   ;GET RETURNED ADDRESS BIASED TO PAR6
011450  005004                  10$:    CLR     R4                      ;WORD IN BUFFER
011452  005003                          CLR     R3                      ;CLEAR ERROR SEEN FLAG
011454  010205                          MOV     R2,R5                   ;GET EXPD BUFFER ADDRESS
011456  011264  002322          15$:    MOV     (R2),EXPMSG(R4)         ;SAVE EXPD FOR ERROR REPORT
011462  011164  002466                  MOV     (R1),RECMSG(R4)         ;SAVE RECV FOR ERROR REPORT
011466  022221                          CMP     (R2)+,(R1)+             ;EXPD EQUAL RECV?
011470  001401                          BEQ     25$                     ;BR IF YES
011472  005203                          INC     R3                      ;SET ERROR SEEN FLAG
011474  062704  000002          25$:    ADD     #2,R4                   ;POINT TO NEXT WORD ADDRESS
011500  020427  000014                  CMP     R4,#14                  ;DONE FIRST 7 WORDS?
011504  003764                          BLE     15$                     ;BR IF NO
011506  022737  000001  002226          CMP     #1,REV                  ;IS IT A NEW MICROCODE
011514  001004                          BNE     30$                     ;NO BR
011516  032765  000200  000012          BIT     #X2.EXTF,XST2(R5)       ;IS EXTENDED FEATURES SET IN EXPD?
011524  001403                          BEQ     50$                     ;BR IF NO
011526  020427  000016          30$:    CMP     R4,#16                  ;DONE EXTENDED FEATURES WORD?
011532  003751                          BLE     15$                     ;BR IF NO
011534  005703                  50$:    TST     R3                      ;ANY ERRORS SEEN?
011536  001402                          BEQ     55$                     ;BR IF NO
011540  000241                          CLC                             ;SET FAILURE
011542  000401                          BR      60$                     ;
011544  000261                  55$:    SEC                             ;SET SUCCESS
011546  000207                  60$:    RTS     PC                      ;RETURN
```

```
                                        .SBTTL  CKMSG2  - COMPARE EXPD RECV MESSAGE BUFFERS
                                ;+
                                ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
                                ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
                                ;ERROR PRINT ROUTINES.
                                ;
                                ;INPUT:
                                ;       R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
                                ;       R2      EXPD MESSAGE BUFFER ADDRESS
                                ;       R3      NUMBER OF BYTES TO COMPARE
                                ;
                                ;OUTPUT:
                                ;       CARRY   SET - MESSAGE BUFFERS MATCH
                                ;               CLR - MESSAGE BUFFERS DON'T MATCH
                                ;
                                ;IMPLICIT OUTPUT:
                                ;       EXPMSG          BUFFER IS SET TO EXPD DATA
                                ;       RECMSG          BUFFER IS SET TO RECV DATA
                                ;       RCVHIADD        SET TO HIGH ORDER ADDRESS OF RECV
                                ;       RCVLOADD        SET TO LOW ORDER ADDRESS OF RECV
                                ;-
011550                          CKMSG2::
                                        SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011554  020327  000144                  CMP     R3,#RECMSG-EXPMSG;@@D IS COUNT ABOVE MAX ALLOWED?
011560  003412                          BLE     5$              ;@@D BR IF NO
011562  012703  000144                  MOV     #RECMSG-EXPMSG,R3;@@D
011566                                  PRINTF  #DEBUGMSG       ;@@D
011566  012746  011702                  MOV     #DEBUGMSG,-(SP)
011572  012746  000001                  MOV     #1,-(SP)
011576  010600                          MOV     SP,R0
011600  104417                          TRAP    C$PNTF
011602  062706  000004                  ADD     #4,SP
011606  010037  002304          5$:     MOV     R0,RCVHIADD     ;SAVE RECV HIGH ADDRESS
011612  010137  002306                  MOV     R1,RCVLOAD      ;SAVE RECV LOW ADDRESS
011616  005737  003134                  TST     KTENABLE        ;TESTING ABOVE 28K?
011622  001403                          BEQ     10$             ;BR IF NO
011624  004737  017376                  JSR     PC,SETMAP       ;RETURN ADDRESS BIASED TO PAR6 IN R0
011630  010001                          MOV     R0,R1           ;GET RETURNED ADDRESS BIASED TO PAR6
011632  005004          10$:    CLR     R4                      ;WORD IN BUFFER
011634  005005                  CLR     R5                      ;CLEAR ERROR SEEN FLAG
011636  111264  002322  15$:    MOVB    (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
011642  111164  002466          MOVB    (R1),RECMSG(R4) ;SAVE RECV FOR ERROR REPORT
011646  122221                  CMPB    (R2)+,(R1)+     ;EXPD EQUAL RECV?
011650  001401                  BEQ     25$             ;BR IF YES
011652  005205                  INC     R5              ;SET ERROR SEEN FLAG
011654  062704  000001  25$:    ADD     #1,R4           ;POINT TO NEXT BYTE
011660  020403                  CMP     R4,R3           ;DONE ALL BYTES?
011662  002001                  BGE     50$             ;BR IF YES
011664  000764                  BR      15$             ;DO NEXT BYTE
011666  005705          50$:    TST     R5              ;ANY ERRORS SEEN?
011670  001402                  BEQ     55$             ;BR IF NO
011672  000241                  CLC                     ;SET FAILURE
011674  000401                  BR      60$             ;
011676  000261          55$:    SEC                     ;SET SUCCESS
011700  000207          60$:    RTS     PC              ;RETURN
```

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```
                                        ;
                                        ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
                                        ;
                                        ;INPUT:
                                        ;
                                        ;       R1      CONTENTS OF TSSR AT ERROR
                                        ;
                                        ;SIDE EFFECTS:
                                        ;
                                        ;       EXECUTES DROP UNIT TO CEASE TESTING
                                        ;
                                        ;-
                                                BGNMSG  SFIMSG
                                        SFIMSG::
                                                JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
                                                JSR     PC,CKDROP       ;DROP UNIT, IF ALLOWED
                                                ENDMSG
                                        L10003:
                                                TRAP    C$MSG
                                        ;
                                        ;PRINT ROUTINE TO PRINT THE CONTENTS OF
                                        ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      TSSR CONTENTS
                                        ;       R4      ADDRESS OF COMMAND PACKET
                                        ;
                                        ;-
                                                BGNMSG  PKTSSR
                                        PKTSSR::
                                                JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
                                                MOV     #4,R0           ;NO. OF WORDS IN PACKET
                                                JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
                                                ENDMSG
                                        L10004:
                                                TRAP    C$MSG
```

```
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 012134                            BGNMSG  PKTGETS
     012134                            PKTGETS::
1434 012134  004737  006024            JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1435 012140  012700  000002            MOV     #2,R0           ;NO. OF WORDS IN GET STATUS PACKET
1436 012144  004737  007370            JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
1437 012150                            ENDMSG
     012150                            L10005:
     012150  104423                    TRAP    C$MSG
1438                                    ;+
1439                                    ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
1440                                    ;
1441                                    ;INPUTS:
1442                                    ;       R1      TSSR CONTENTS
1443                                    ;       R4      ADDRESS OF COMMAND PACKET
1444                                    ;-
1445 012152                            BGNMSG  SFFMSG
     012152                            SFFMSG::
1446 012152  004737  006024            JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1447 012156                            ENDMSG
     012156                            L10006:
     012156  104423                    TRAP    C$MSG
1448                                    .SBTTL  PKTMES  - PRINT TSSR AND MESSAGE BUFFER
1449                                    ;+
1450                                    ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
1451                                    ;BUFFER FOR ERROR REPORTS
1452                                    ;
1453                                    ;INPUTS:
1454                                    ;
1455                                    ;       R1      CONTENTS OF TSSR
1456                                    ;       R2      LOW ORDER MESSAGE BUFFER
1457                                    ;       R3      HIGH ORDER MESSAGE BUFFER ADDRESS
1458                                    ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
1459                                    ;-
1460 012160                            BGNMSG  PKTMES
     012160                            PKTMES::
1461 012160  004737  006024            JSR     PC,PRITSSR              ;PRINT CONTENTS OF TSSR
1462 012164  010200                    MOV     R2,R0                   ;LOW ORDER ADDRESS
1463 012166  010301                    MOV     R3,R1                   ;HIGH ORDER ADDRESS
1464 012170  004737  014312            JSR     PC,PRMESS               ;PRINT THE MESSAGE BUFFER
1465 012174                            ENDMSG
     012174                            L10007:
     012174  104423                    TRAP    C$MSG
```

```
                                        .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
012176                          ADDSSR::
012176  004737  010274                  JSR     PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
012202  016501  000002                  MOV     TSSR(R5),R1     ;GET CURRENT TSSR
012206  004737  006024                  JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
012212                                  ENDMSG
012212                          L10010:
012212  104423                          TRAP    C$MSG

                                        .SBTTL  MSGEXP - PRINT WRITE CHAR. EXPD-RECV MESSAGE BUFFERS
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
                                        BGNMSG  MSGEXP
012214                          MSGEXP::
012214  012700  000007                  MOV     #7,R0           ;ASSUME NO EXT FEATURES
012220  005737  002224                  TST     EXTFEA          ;EXT FEATURES SET?
012224  001402                          BEQ     5$              ;BR IF NO
012226  012700  000010                  MOV     #8.,R0          ;EXT FEATURE BUFFER IS 8 WORDS
012232  004737  014632          5$:     JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012236                                  ENDMSG
012236                          L10011:
012236  104423                          TRAP    C$MSG
```

M5

FIFEXP - PRINT FIFO EXP/RECV DATA

```
                                        .SBTTL  FIFEXP  - PRINT FIFO EXP/RECV DATA
                                ;*
                                ;
                                ;PRINT ROUTINE TO PRINT FIFO EXP/RECV DATA
                                ;
                                ;       R1      - BYTE COUNT
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
                                ;-
                                        BGNMSG  FIFEXP
012240                          FIFEXP::
012240                                  PRINTX  #FIF1MSG,R1     ;PRINT BYTES TRANSFERRED
012240  010146                          MOV     R1,-(SP)
012242  012746  012312                  MOV     #FIF1MSG,-(SP)
012246  012746  000002                  MOV     #2,-(SP)
012252  010600                          MOV     SP,R0
012254  104415                          TRAP    C$PNTX
012256  062706  000006                  ADD     #6,SP
012262                                  PRINTX  #FIF2MSG        ;PRINT HEADER MSG
012262  012746  012361                  MOV     #FIF2MSG,-(SP)
012266  012746  000001                  MOV     #1,-(SP)
012272  010600                          MOV     SP,R0
012274  104415                          TRAP    C$PNTX
012276  062706  000004                  ADD     #4,SP
012302  010100                          MOV     R1,R0           ;GET BYTE COUNT
012304  004737  015202                  JSR     PC,PRBYTEXP     ;PRINT FIFO BYTES IN ERROR
012310                                  ENDMSG
012310                          L10012:
012310  104423                          TRAP    C$MSG
012312     045     116     045  FIF1MSG:        .ASCIZ  '<CR><LF>A NUMBER OF BYTES TRANSFERRED = <D2>'
012361     045     116     045  FIF2MSG:        .ASCIZ  '<CR><LF>A FIFO DATA BYTES IN ERROR:'
                                        .EVEN
```

```
                                        .SBTTL  MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
                                ;*
                                ;
                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
                                ;
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
012420                                  BGNMSG  MSGSTAT
012420                          MSGSTAT::
012420  012701  012462                   MOV     #STATCOD,R1     ;ASCII ADDRESS TABLE
012424  012100                  10$:     MOV     (R1)+,R0        ;DONE ALL MSG LINES?
012426  001410                           BEQ     20$             ;BR IF YES
012430                                   PRINTX  R0              ;PRINT STATUS BIT NAMES
012430  010046                           MOV     R0,-(SP)
012432  012746  000001                   MOV     #1,-(SP)
012436  010600                           MOV     SP,R0
012440  104415                           TRAP    C$PNTX
012442  062706  000004                   ADD     #4,SP
012446  000766                           BR      10$             ;DO ANOTHER MSG LINE
012450  012700  000012          20$:     MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
012454  004737  014632                   JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012460                                   ENDMSG
012460                          L10013:
012460  104423                           TRAP    C$MSG

012462  012500  012542  012633  STATCOD:        .WORD   1$,2$,3$,4$,5$,6$,0
012500    045     116     045   1$:.ASCIZ  '<LF><LF>A Tape Bus Signals in Word #0:'
012542    045     116     045   2$:.ASCIZ  '<LF><LF>A       PARERR<15>  IEOT  <12>  IFMK  <9>  IRDY<6>  IRWD<2>'
012633    045     116     045   3$:.ASCIZ  '<LF><LF>A       IRESV2<14>  IIDENT<11>  IHER  <8>  IONL<5>  IFBY<1>'
012724    045     116     045   4$:.ASCIZ  '<LF><LF>A       IRESV1<13>  ICER  <10>  ISPEED<7>  ILDP<4>  IFPT<0>'
013015    045     116     045   5$:.ASCIZ  '<LF><LF>A Tape Bus Signals in Word #9:'
013057    045     116     045   6$:.ASCIZ  '<LF><LF>A       DATMIS<7>  ILW<6>  OUTRDY<5>  INRDY<4>'
                                        .EVEN
```

MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS

```
                                        .SBTTL  MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
                                ;*
                                ;
                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
013134                          BGNMSG  MSGLOOP
013134                          MSGLOOP::
013134  012701  013176          MOV     #LOOPCOD,R1     ;ASCII ADDRESS TABLE
013140  012100                  10$:    MOV     (R1)+,R0        ;DONE ALL MSG LINES?
013142  001410                  BEQ     20$             ;BR IF YES
013144                          PRINTX  R0              ;PRINT STATUS BIT NAMES
013144  010046                  MOV     R0,-(SP)
013146  012746  000001          MOV     #1,-(SP)
013152  010600                  MOV     SP,R0
013154  104415                  TRAP    C$PNTX
013156  062706  000004          ADD     #4,SP
013162  000766                  BR      10$             ;DO ANOTHER MSG LINE
013164  012700  000012          20$:    MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
013170  004737  014632          JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
013174                          ENDMSG
013174                          L10014:
013174  104423                  TRAP    C$MSG

013176  013216  013271  013370  LOOPCOD:        .WORD   1$,2$,3$,4$,5$,6$,7$,0
013216    045     116     045   1$:.ASCIZ  '%N%A Tape Bus Loopback Signals in Word #8:'
013271    045     116     045   2$:.ASCIZ  '%N%A       PARERR<15>          IRESV2<14>          IRESV1<13>'
013370    045     116     045   3$:.ASCIZ  '%N%A IHISP=>IEOT<12>    IWRT=>IIDENT<11>    IREV  =>ICER  <10>'
013467    045     116     045   4$:.ASCIZ  '%N%A IWFM =>IFMK<09>    IEDIT=>IHER <08>    IFAD  =>ISPEED<07>'
013566    045     116     045   5$:.ASCIZ  '%N%A ITAD0=>IRDY<06>    ITAD1=>IONL <05>    IERASE=>ILDP  <04>'
013665    045     116     045   6$:.ASCIZ  '%N%A IREW =>IDBY<03>    IRWU =>IRWD <02>    IFEN  =>IFBY  <01>'
013764    045     116     045   7$:.ASCIZ  '%N%A IGO  =>IFPT<00>'
                                        .EVEN
```

C6

MSGSUB PRINT WRITE SUBSYSTEM MESSAGE BUFFER

```
                                        .SBTTL  MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
                                ;*
                                ;
                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
                                ;
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
                                        BGNMSG  MSGSUB
014012                          MSGSUB::
014012  012700  000012                  MOV     #10.,R0         ;SIZE OF WRITE SUBSYSTEM BUFFER
014016  004737  014632                  JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
014022                                  ENDMSG
014022                          L10015:
014022  104423                          TRAP    C$MSG

                                        .SBTTL  MEMADD - PRINT MEMORY ADDRESS DATA ERROR
                                ;*
                                ;
                                ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       ERRHI   - MEMORY ERROR HIGH ORDER ADDRESS
                                ;       ERRLO   - MEMORY ERROR LOW ORDER ADDRESS
                                ;       EXP     - EXPECTED DATA
                                ;       RECV    - RECEIVED DATA
                                ;-
                                        BGNMSG  MEMADD
014024                          MEMADD::
014024  004737  010160                  JSR     PC,PRIADD       ;PRINT MEMORY ADDRESS IN ERROR
014030  013701  002232                  MOV     EXPD,R1         ;GET EXPD DATA
014034  013702  002234                  MOV     RECV,R2         ;GET RECEIVED DATA
014040  004737  007742                  JSR     PC,PRIXOR       ;PRINT EXPD/RECV
014044                                  ENDMSG
014044                          L10016:
014044  104423                          TRAP    C$MSG
```

D6

PRAMPKT - PRINT RAM AND PACKET DATA

```
                                        .SBTTL  PRAMPKT - PRINT RAM AND PACKET DATA
                                ;+
                                ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
                                ;WHEN THE RAM DATA DOES NOT MATCH.
                                ;
                                ;INPUTS:
                                ;
                                ;       R4      POINTER TO COMMAND PACKET
                                ;IMPLICIT INPUTS:
                                ;       RAMDATA         DATA AS READ FROM THE RAM
                                ;       RAMSIZ          NUMBER OF BYTES IN PACKET
                                ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
                                ;
                                ;IMPLICIT OUTPUTS:
                                ;       RAMSIZ  SET TO 0
                                ;-
                                PRAMPKT:
014046                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
014046  012701  002242                  MOV     #RAMDATA,R1             ;DATA FROM THE RAM
014052  005002                          CLR     R2                      ;INIT BYTE NUMBER
014056  122124                  5$:     CMPB    (R1)+,(R4)+             ;COMPARE EXPECTED, RECEIVED
014060  001005                          BNE     7$                      ;BR IF NO MATCH
014062                                  FORCERROR       7$,NOTSSR
014074  000436                          BR      10$                     ;add
014076  116105  177777          7$:     MOVB    -1(R1),R5               ;GET RECV RAM DATA
014102  116403  177777                  MOVB    -1(R4),R3               ;GET EXPD PACKET DATA
014106                                  XOR     R5,R3                   ;XOR EXPD/RECV
014116  042703  177400                  BIC     #177400,R3              ;LOW BYTE ONLY
014122  116137  177777  002234          MOVB    -1(R1),RECV             ;GET RECEIVED RAM DATA
014130  116437  177777  002232          MOVB    -1(R4),EXPD             ;GET EXPECTED RAM DATA
014136                                  PRINTB  #RAMASC,R2,RECV,EXPD,R3
014136  010346                          MOV     R3,-(SP)
014140  013746  002232                  MOV     EXPD,-(SP)
014144  013746  002234                  MOV     RECV,-(SP)
014150  010246                          MOV     R2,-(SP)
014152  012746  014226                  MOV     #RAMASC,-(SP)
014156  012746  000005                  MOV     #5,-(SP)
014162  010600                          MOV     SP,R0
014164  104414                          TRAP    C$PNTB
014166  062706  000014                  ADD     #14,SP
014172  005202          10$:    INC     R2                      ;UPDATE BYTE COUNT
014174  005737  002302                  TST     RAMSIZ                  ;DEFAULT TO 8.?
014200  001404                          BEQ     15$                     ;BR IF YES
014202  020237  002302                  CMP     R2,RAMSIZ               ;DONE ALL BYTES?
014206  003724                          BLE     5$                      ;BR IF NO
014210  000403                          BR      25$                     ;
014212  020227  000010          15$:    CMP     R2,#8.                  ;DONE DEFAULT NUMBER OF BYTES?
014216  002720                  20$:    BLT     5$                      ;BR IF NO
014220  005037  002302          25$:    CLR     RAMSIZ                  ;SET DEFAULT RAMSIZ
014224  000207                          RTS     PC                      ;RETURN

014226      045     116     045 RAMASC: .ASCIZ  '%N%A  BYTE: %D2%A RAM: %O3%A Packet: %O3%A XOR:%O3'
                                        .EVEN
```

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```
                                        .SBTTL  PRMESS  - PRINT CONTENTS OF MESSAGE BUFFER
                                ;+
                                ;THIS ROUTINE PRINTS THE CONTENTS OF
                                ;THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE TSV-05.
                                ;
                                ;INPUT:
                                ;       R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
                                ;       R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
                                ;       NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
                                ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
                                ;-
014312                          PRMESS: SAVREG                  ;SAVE THE REGISTERS
014316  010005                          MOV     R0,R5           ;SAVE LOW ORDER ADDRESS
014320  005737  003134                  TST     KTENABLE        ;ADDRESS ABOVE 28K?
014324  001001                          BNE     10$             ;BR IF YES
014326  005001                          CLR     R1              ;SET HIGH ORDER ADDRESS TO 0
014330  010103                  10$:    MOV     R1,R3           ;SAVE HIGH ORDER ADDRESS
014332  006100                          ROL     R0              ;SHIFT BIT15 TO C BIT
014334  006101                          ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
014336                          1693    PRINTX  #PROASC,R1,R5   ;PRINT MESSAGE BUFFER ADDRESS
014336  010546                          MOV     R5,-(SP)
014340  010146                          MOV     R1,-(SP)
014342  012746  014500                  MOV     #PROASC,-(SP)
014346  012746  000003                  MOV     #3,-(SP)
014352  010600                          MOV     SP,R0
014354  104415                          TRAP    C$PNTX
014356  062706  000010                  ADD     #10,SP
014362                          1694    PRINTX  #PR1ASC         ;PRINT HEADER FOR CONTENTS
014362  012746  014545                  MOV     #PR1ASC,-(SP)
014366  012746  000001                  MOV     #1,-(SP)
014372  010600                          MOV     SP,R0
014374  104415                          TRAP    C$PNTX
014376  062706  000004                  ADD     #4,SP
014402  005004                  1695    CLR     R4              ;NUMBER OF THE NEXT WORD
014404  010501                  1696    MOV     R5,R1           ;COPY LOW ORDER ADDRESS
014406  010300                  1697    MOV     R3,R0           ;COPY HIGH ORDER ADDRESS
014410  001403                  1698    BEQ     20$             ;BR IF NOT ABOVE 28K
014412  004737  017376          1699    JSR     PC,SETMAP       ;SETUP PAR ADDRESS IN R0
014416  010005                  1700    MOV     R0,R5           ;GET PAR FORMAT ADDRESS ABOVE 28K
014420                          1701    PRINTX  #PRASC,R4,(R5)+ ;PRINT THE CONTENTS OF MEMORY BUFFER
014420  012546                          MOV     (R5)+,-(SP)
014422  010446                          MOV     R4,-(SP)
014424  012746  014603                  MOV     #PRASC,-(SP)
014430  012746  000003                  MOV     #3,-(SP)
014434  010600                          MOV     SP,R0
014436  104415                          TRAP    C$PNTX
014440  062706  000010                  ADD     #10,SP
014444  005204                  1702    INC     R4              ;NUMBER OF THE NEXT
014446  020427  000007          1703    CMP     R4,#7           ;DONE ALL YET ?
014452  003011                  1704    BGT     50$             ;BRANCH IF ALL DONE
014454  002761                  1705    BLT     20$             ;PRINT FIRST 7 WORDS
014456  022737  000001  002226  1706    CMP     #1,REV          ;IS IT A NEW MICROCODE
014464  001355                  1707    BNE     20$             ;NO BR
014466  032763  000200  000012  1708    BIT     #X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
014474  001351                  1709    BNE     20$             ;PRINT EXTENDED STATUS WORD
014476  000207                  1710    RTS     PC              ;RETURN
014500      045     116     045 1711    PROASC: .ASCIZ  '%N%A Message Buffer Address = %O1%O5'
```

F6

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: %O'
```

```
                                        .EVEN
                                        .SBTTL  PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
                                ;+
                                ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
                                ;       R0      - NUMBER OF WORDS IN BUFFER
                                ;IMPLICIT INPUTS:
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
014632                          PRMSGEXP::
014632                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
014636  010005                          MOV     R0,R5                   ;SAVE NUMBER OF WORDS
014640  013700  002306                  MOV     RCVLOADD,R0             ;GET RECV LOW ADDRESS
014644  010004                          MOV     R0,R4                   ;COPY LOW ADDRESS
014646  013701  002304                  MOV     RCVHIADD,R1             ;GET RECV HIGH ADDRESS
014652  006100                          ROL     R0                      ;SHIFT BIT15 TO C BIT
014654  006101                          ROL     R1                      ;SHIFT TO HIGH ORDER FOR PRINTOUT
014656                                  PRINTX  #PRMSG0,R1,R4           ;PRINT MESSAGE BUFFER ADDRESS
014656  010446                          MOV     R4,-(SP)
014660  010146                          MOV     R1,-(SP)
014662  012746  015012                  MOV     #PRMSG0,-(SP)
014666  012746  000003                  MOV     #3,-(SP)
014672  010600                          MOV     SP,R0
014674  104415                          TRAP    C$PNTX
014676  062706  000010                  ADD     #10,SP
014702                                  PRINTX  #PRMSG1                 ;PRINT HEADER FOR CONTENTS
014702  012746  015057                  MOV     #PRMSG1,-(SP)
014706  012746  000001                  MOV     #1,-(SP)
014712  010600                          MOV     SP,R0
014714  104415                          TRAP    C$PNTX
014716  062706  000004                  ADD     #4,SP
014722  005004                          CLR     R4                      ;NUMBER OF THE CURRENT WORD
014724  012701  002322                  MOV     #EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
014730  012702  002466                  MOV     #RECMSG,R2              ;GET RECV BUFFER ADDRESS
014734  011100                  20$:    MOV     (R1),R0                 ;GET EXPD
014736  011203                          MOV     (R2),R3                 ;GET RECV
014740                                  XOR     R0,R3                   ;XOR EXPD/RECV
014750                                  PRINTX  #PRMSG2,R4,(R1)+,(R2)+,R3
014750  010346                          MOV     R3,-(SP)
014752  012246                          MOV     (R2)+,-(SP)
014754  012146                          MOV     (R1)+,-(SP)
014756  010446                          MOV     R4,-(SP)
014760  012746  015115                  MOV     #PRMSG2,-(SP)
014764  012746  000005                  MOV     #5,-(SP)
014770  010600                          MOV     SP,R0
014772  104415                          TRAP    C$PNTX
014774  062706  000014                  ADD     #14,SP
015000  005204                          INC     R4                      ;NUMBER OF THE NEXT
015002  020405                          CMP     R4,R5                   ;DONE ALL YET?
015004  002001                          BGE     50$                     ;BR IF YES
015006  000752                          BR      20$                     ;DO ANOTHER
015010  000207                  50$:    RTS     PC                      ;RETURN
015012     045     116     045  PRMSG0: .ASCIZ  '%N%A Message Buffer Address = %01%05'
015057     045     116     045  PRMSG1: .ASCIZ  '%N%A Message Buffer Contents:'
015115     045     116     045  PRMSG2: .ASCIZ  '%N%A    WORD #%D2%A  EXPD: %06%A  RECV: %06%A  XOR: %06'
```

H6

PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS

```
                                        .EVEN
                                        .SBTTL  PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
                                ;+
                                ;
                                ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
                                ;       ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
                                ;
                                ;       R0      - NUMBER OF BYTES IN BUFFER
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;-
015202                          PRBYTEXP::
015202                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
015206  010005                          MOV     R0,R5                   ;SAVE NUMBER OF BYTES
015210  005037  002320                  CLR     PRMNO                   ;INIT ERROR COUNT
015214  005004                          CLR     R4                      ;NUMBER OF THE CURRENT BYTE
015216  012701  002322                  MOV     @EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
015222  012702  002466                  MOV     @RECMSG,R2              ;GET RECV BUFFER ADDRESS
015226  111100                  20$:    MOVB    (R1),R0                 ;GET EXPD BYTE
015230  042700  177400                  BIC     #^C<377>,R0             ;CLEAR UPPER BYTE
015234  110037  015550                  MOVB    R0,PRBEXP               ;SAVE FOR ERROR REPORT
015240  111203                          MOVB    (R2),R3                 ;GET RECV BYTE
015242  042703  177400                  BIC     #^C<377>,R3             ;CLEAR UPPER BYTE
015246  110337  015552                  MOVB    R3,PRBREC               ;FOR ERROR REPORT
015252                                  XOR     R0,R3                   ;XOR EXPD/RECV
015262  122122                          CMPB    (R1)+,(R2)+             ;EXPD = RECV?
015264  001431                          BEQ     30$                     ;BR IF YES
015266  005237  002320                  INC     PRMNO                   ;UPDATE ERROR COUNT
015272  023727  002320  000010          CMP     PRMNO,#8.               ;PRINTED 8?
015300  101023                          BHI     30$                     ;BR IF YES
015302                          27$:    PRINTX  #PRBMSG,R4,PRBEXP,PRBREC,R3
015302  010346                          MOV     R3,-(SP)
015304  013746  015552                  MOV     PRBREC,-(SP)
015310  013746  015550                  MOV     PRBEXP,-(SP)
015314  010446                          MOV     R4,-(SP)
015316  012746  015416                  MOV     #PRBMSG,-(SP)
015322  012746  000005                  MOV     #5,-(SP)
015326  010600                          MOV     SP,R0
015330  104415                          TRAP    C$PNTX
015332  062706  000014                  ADD     #14,SP
015336                                  FORCEXIT        50$             ;@@D
015346  000404                          BR      35$                     ;@@D
015350                          30$:
015350                                  FORCERROR  27$,NOTSSR           ;@@D
015360                          35$:                                    ;@@D
015360  005204                          INC     R4                      ;NUMBER OF THE NEXT
015362  020405                          CMP     R4,R5                   ;DONE ALL YET?
015364  002001                          BGE     50$                     ;BR IF YES
015366  000717                          BR      20$                     ;DO ANOTHER
015370                          50$:    PRINTX  #PRBTOT,PRMNO           ;PRINT TOTAL ERROR COUNT
015370  013746  002320                  MOV     PRMNO,-(SP)
015374  012746  015503                  MOV     #PRBTOT,-(SP)
015400  012746  000002                  MOV     #2,-(SP)
015404  010600                          MOV     SP,R0
```

```
PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER

        015406  104415                          TRAP    C$PNTX
        015410  062706  000006                  ADD     #6,SP
  1796  015414  000207                          RTS     PC              ;RETURN
  1797
  1798  015416     045     116     045  PRBMSG: .ASCIZ  '#N#A   BYTE #D2#A  EXPD: #03#A  RECV: #03#A  XOR: #03'
  1799  015503     045     116     045  PRBTOT: .ASCIZ  '#N#A NUMBER OF BYTES IN ERROR = #D2'
  1800                                          .EVEN
  1801  015550  000000                  PRBEXP: .WORD   0               ;EXPD
  1802  015552  000000                  PRBREC: .WORD   0               ;RECV
  1803                                          .SBTTL  EXPREC  - PRINT EXPD/RECV WORD DATA
  1804                                  ;+
  1805                                  ;
  1806                                  ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
  1807                                  ;
  1808                                  ;INPUTS:
  1809                                  ;
  1810                                  ;       R1      RECEIVED DATA
  1811                                  ;       R2      EXPECTED DATA
  1812                                  ;
  1813                                  ;-
  1814
  1815  015554                                  BGNMSG  EXPREC
        015554                          EXPREC::
  1816  015554  004737  007742                  JSR     PC,PRIXOR               ;PRINT THE DATA
  1817  015560                                  ENDMSG
        015560                          L10017:
        015560  104423                          TRAP    C$MSG
  1818                                          .SBTTL  EXPBREC - PRINT EXPD/RECV BYTE DATA
  1819                                  ;+
  1820                                  ;
  1821                                  ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
  1822                                  ;
  1823                                  ;
  1824                                  ;INPUTS:
  1825                                  ;
  1826                                  ;       R1      RECEIVED DATA BYTE
  1827                                  ;       R2      EXPECTED DATA BYTE
  1828                                  ;
  1829                                  ;-
  1830
  1831  015562                                  BGNMSG  EXPBREC
        015562                          EXPBREC::
  1832  015562  004737  007612                  JSR     PC,PRIBXOR              ;PRINT THE DATA
  1833  015566                                  ENDMSG
        015566                          L10020:
        015566  104423                          TRAP    C$MSG
  1834
  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

```
                                        ;IMPLICIT INPUTS:
                                        ;
                                        ;       RAMDATA         DATA AS READ FROM THE RAM
                                        ;       RAMSIZ          NUMBER OF BYTES IN PACKET
                                        ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
                                        ;
                                        ;IMPLICIT OUTPUTS:
                                        ;
                                        ;       RAMSIZ  SET TO 0
                                        ;-
                                        
                                                BGNMSG  RAMERR
                                        RAMERR::
015570  004737  014046                          JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
                                                ENDMSG
                                        L10021:
015574  104423                                  TRAP    C$MSG
                                        
                                                .SBTTL  RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
                                        ;*
                                        ;
                                        ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R4      POINTER TO COMMAND PACKET
                                        ;
                                        ;IMPLICIT INPUTS:
                                        ;
                                        ;       RAMDATA         DATA AS READ FROM THE RAM
                                        ;       RAMSIZ          NUMBER OF BYTES IN PACKET
                                        ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
                                        ;       ERRHI           HIGH ORDER TEST ADDRESS
                                        ;       ERRLO           LOW ORDER TEST ADDRESS
                                        ;
                                        ;IMPLICIT OUTPUTS:
                                        ;
                                        ;       RAMSIZ  SET TO 0
                                        ;-
                                        
                                                BGNMSG  RAMTADD
                                        RAMTADD::
015576  004737  010274                          JSR     PC,PRITADD      ;PRINT TEST ADDRESS
015602  004737  014046                          JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
                                                ENDMSG
                                        L10022:
015606  104423                                  TRAP    C$MSG
                                        
                                                .SBTTL  RAMEXP  - PRINT RAM EXPD/RECV DATA
                                        ;*
                                        ;
                                        ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      RECEIVED DATA
                                        ;       R2      EXPECTED DATA
```

K6

RAMEXP - PRINT RAM EXPD/RECV DATA

```
                                        ;       R4      CONTROLLER RAM ADDRESS
                                        ;-

                                                BGNMSG  RAMEXP
                                        RAMEXP::
                                                BIC     #^C<377>,R1             ;SAVE EXPD RAM DATA BYTE
                                                BIC     #^C<377>,R2             ;SAVE EXPD RAM DATA BYTE
                                                JSR     PC,PRIRAM               ;PRINT THE RAM ADDRESS
                                                JSR     PC,PRIXOR               ;PRINT THE DATA
                                                ENDMSG
                                        L10023:
                                                TRAP    C$MSG

                                                .SBTTL  TIMEXP  - PRINT TIMER A,B AND EXP/REC
                                        ;*
                                        ;
                                        ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                                        ;AND TIMER A,B HEADER MESSAGE
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      RECEIVED DATA
                                        ;       R2      EXPECTED DATA
                                        ;-

                                                BGNMSG  TIMEXP
                                        TIMEXP::
                                                PRINTX  #TIMSG0                 ;PRINT HEADER
015632  012746  015660                          MOV     #TIMSG0,-(SP)
015636  012746  000001                          MOV     #1,-(SP)
015642  010600                                  MOV     SP,R0
015644  104415                                  TRAP    C$PNTX
015646  062706  000004                          ADD     #4,SP
                                                JSR     PC,PRIXOR               ;PRINT THE DATA
                                                ENDMSG
                                        L10024:
                                                TRAP    C$MSG

015660      045     116     045 TIMSG0: .ASCIZ  '<NL>A TIMER A STATUS IS IN BIT 3<NL>A TIMER B STATUS IS IN BIT 2'
                                        .EVEN
                                                .SBTTL  BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

                                        ;*
                                        ;
                                        ;PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      CONTENTS OF TSSR
                                        ;       R2      DATA WRITTEN (8 BITS)
                                        ;
                                        ;-

                                                BGNMSG  BADSSR
                                        BADSSR::
                                                MOV     R2,-(SP)                ;SAVE DATA TRANSFERRED
                                                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

```
                                .SBTTL  GLOBAL SUBROUTINES SECTION

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

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

016054                          SOFINIT::
016054                                  SAVREG                          ; SAVE THE REGISTERS
016060  012765  000000  000002          MOV     #0,TSSR(R5)             ; DO THE INIT.
016066  004737  016330                  JSR     PC,WAITF                ; WAIT FOR SSR
016072  016500  000002                  MOV     TSSR(R5),R0             ;GET THE TSSR REGISTER
016076  010004                          MOV     R0,R4                   ;TSSR CONTENTS
016100  042704  176277                  BIC     #^C<HIADDR!OFL>,R4
016104  052704  002200                  BIS     #SSR!NBA,R4             ;R4 HAS EXPECTED CONTENTS
016110  020400                          CMP     R4,R0                   ;ONLY EXPECTED BITS SET ?
016112  001402                          BEQ     5$                      ;BRANCH IF OKAY
016114  000241                          CLC                             ;CLEAR THE CARRY FOR ERROR
016116  000401                          BR      10$                     ;GO TO EXIT
016120  000261                  5$:     SEC                             ;SET THE CARRY BIT
016122  000207                  10$:    RTS     PC                      ;RETURN TO CALLER
```

N6

CHKAMB - CHECK TSSR FOR AMBIGUITY

```
                                .SBTTL  CHKAMB  - CHECK TSSR FOR AMBIGUITY

                                ;+
                                ;
                                ;THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
                                ;FOR AMBIGUITY
                                ;
                                ;INPUT:
                                ;
                                ;       R0      CONTENTS OF TSSR
                                ;
                                ;OUTPUT:
                                ;
                                ;       R0      CONTENTS OF TSSR
                                ;
                                ;       CARRY   SET - NO AMBIGUITY
                                ;               CLR - AMBIGUOUS CONTENTS
                                ;
                                ;-

016124                          CHKAMB:
016124                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
016130  010004                          MOV     R0,R4                   ;CONTENTS OF TSSR
016132  032700  100000                  BIT     #SC,R0                  ;IS BIT 15 SET ?
016136  001004                          BNE     5$                      ;BRANCH IF YES
016140  032700  174077                  BIT     #^C<NBA!OFL!SSR!HIADDR>,R0      ;ANY OTHER BITS SET ?
016144  001023                          BNE     40$                     ;MUST BE AN ERROR
016146  000424                          BR      45$                     ;RETURN WITH SUCCESS
016150  032700  000200          5$:     BIT     #SSR,R0                 ;IS READY BIT SET ?
016154  001011                          BNE     10$                     ;BRANCH IF READY BIT IS SET.
016156  032700  000040                  BIT     #BIT5,R0                ;IS FATAL ERROR BIT SET ?
016162  001414                          BEQ     40$                     ;ERROR IF NOT
016164  042704  177761                  BIC     #^CTERCLS,R4            ;CLEAR ALL BUT TERMINATION CODE
016170  020427  000016                  CMP     R4,#16                  ;ALL THREE BITS MUST BE SET
016174  001007                          BNE     40$                     ;ERROR IF NOT SET
016176  000410                          BR      45$                     ;OK IF ALL ARE SET
016200  032700  000040          10$:    BIT     #BIT5,R0                ;IS FATAL ERROR BIT SET ?
016204  001405                          BEQ     45$                     ;ERROR IF BIT IS SET WITH SSR
016206  032700  000006                  BIT     #BIT2!BIT1,R0           ;IS THIS A FUNCTION REJECT
016212  001002                          BNE     45$                     ;BR, IF TSSR IS OK
016214  000241          40$:    CLC                                     ;AMBIGUOUS CONTENTS
016216  000401                  BR      50$
016220  000261          45$:    SEC                                     ;SHOW SUCCESS - NO AMBIGUITY
016222  000207          50$:    RTS     PC                              ;RETURN TO CALLER
```

```
                                .SBTTL  ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS
                                ;
                                ; DEFAULT DISPLAY INTERRUPT HANDLERS.
                                ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
                                ; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
                                ;

                                ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
                                ;
 000200                                 IOKCKIN=BIT7    ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
 000001                                 IOKSTP=BIT0     ; EXPECT "STOP" INTERRUPT.

                                ;INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
016224   000                   INTMASK:        .BYTE   0
                                ;INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
016225   000                   INTFLAG:        .BYTE   0

                                ;SAVED INTERRUPT VECTOR:
016226  000000                  INTVEC: .WORD   0
                                ;SAVE CPU PC
016230  000000                  INTCPC: .WORD   0

                                ;SUBROUTINE TO ENABLE INTERRUPTS:
016232  010046                  ENAINT: MOV     R0,-(SP)        ;SAVE R0
016234  013700  002206                  MOV     IVEC,R0         ;GET POINTER TO VECTORS
016240  012720  016276                  MOV     #INTR,(R0)+     ;SET UP INTERRUPT VECTOR
016244  012720  000340                  MOV     #PRI07,(R0)+
016250  012600                          MOV     (SP)+,R0        ;RESTORE R0
016252  011646                          MOV     (SP),-(SP)
016254  012766  000000  000002          MOV     #0,2(SP)        ;SET CPU TO LEVEL 0
016262  000002                          RTI

                                ;SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
016264  011646                  DSBINT: MOV     (SP),-(SP)
016266  012766  000340  000002          MOV     #PRI07,2(SP)
016274  000002                          RTI
                                        .SBTTL  INTR    - INTERRUPT HANDLERS

                                        BGNSRV  INTR            ;DEFINE INTERRUPT ENTRY
016276
016276                          INTR::
016276  012737  000001  002222          MOV     #1,INTRECV      ;SET FLAG TO SHOW INTERRUPT RECEIVED
016304  105037  016225                  CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
016310  132737  000001  016224          BITB    #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
016316  001003                          BNE     1$              ;BR IF YES
016320  152737  000001  016225          BISB    #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.

                                ;SAVE REGISTERS, MSG BUFFER, ETC.
016326                          1$:
016326                                  ENDSRV
016326                          L10026:
016326  000002                          RTI
```

WAITF - WAIT FOR SUBSYSTEM READY

```
                                        .SBTTL  WAITF   - WAIT FOR SUBSYSTEM READY
                                ;
                                ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
                                ;
                                ;INPUTS:
                                ;
                                ;       R5      ADDRESS OF FIRST DEVICE REGISTER
                                ;
                                ;OUTPUTS:
                                ;
                                ;       R0      CONTENTS OF LAST TSSR READ
                                ;       CARRY   SET - READY BIT SET
                                ;               CLR - TIMEOUT WAITING FOR READY
                                ;
016330  000401                  WAITF:: BR      1$                      ;NOP WHEN SUPER FIXED
016332                                  BREAK                           ; DO A SUPVSR BREAK FIRST.
016332  104422                          TRAP    C$BRK
016334  012746  011000          1$:     MOV     #11000,-(SP)            ;25-APRIL-83 REV B  - 1100 MSEC TIMER
016340  016500  000002          2$:     MOV     TSSR(R5),R0             ;READ THE TSSR REGISTER
016344  105700                          TSTB    R0                      ;TEST FOR READY BIT SET
                                ;
016346  100420                          BMI     3$                      ; EXIT ON STOP FLAG.
016350                                  DELAY   1                       ; WAIT 100 USEC
016350  012727  000001                  MOV     #1,(PC)+
016354  000000                          .WORD   0
016356  013727  002116                  MOV     L$DLY,(PC)+
016362  000000                          .WORD   0
016364  005367  177772                  DEC     -6(PC)
016370  001375                          BNE     .-4
016372  005367  177756                  DEC     -22(PC)
016376  001367                          BNE     .-20
016400  005316                          DEC     (SP)                    ;REDUCE DELAY COUNT
016402  001356                          BNE     2$                      ;RETRY UNTIL TIMER EXPIRES
016404  000241                          CLC                             ; C = 0, CONTROLLER STILL RUNNING...
016406  000401                          BR      4$                      ;...OR HUNG-UP AFTER 300 MSEC.
016410  000261                  3$:     SEC                             ; C = 1, CONTROLLER IS STOPPED.
016412  005326                  4$:     DEC     (SP)+                   ;RESTORE STACK WITHOUT CHANGING CARRY BIT
016414  000207                          RTS     PC
```

CHKTSSR - CHECK TSSR FOR READY

```
                                .SBTTL  CHKTSSR - CHECK TSSR FOR READY
                                ;+
                                ;THIS ROUTINE WAITS FOR READY IN THE TSSR
                                ;AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
                                ;
                                ;INPUT:
                                ;       R5      ADDRESS OF CSR REGISTERS
                                ;
                                ;OUTPUT:
                                ;       R0      CONTENTS OF TSSR
                                ;       CARRY   SET - OKAY
                                ;               CLR - NOT READY AMBIGUOUS, OR SC SET
                                ;-
016416                          CHKTSSR:
016416  004737  016330                  JSR     PC,WAITF                ;WAIT FOR READY
016422  103014                          BCC     20$                     ;BRANCH IF TIME OUT
016424  004737  016124                  JSR     PC,CHKAMB               ;TSSR AMBIGOUS?
016430  103006                          BCC     10$                     ;BR IF YES
016432  032700  100000                  BIT     #SC,R0                  ;SPECIAL CONDITION SET?
016436  001405                          BEQ     15$                     ;BR IF NO
016440  032700  074000                  BIT     #<SCL!BIE!RMR!NXM>,R0   ;ANY ERROR BITS SET?
016444  001402                          BEQ     15$                     ;BR IF NO
016446  000241                  10$:    CLC                             ;SET FAILURE
016450  000401                          BR      20$                     ;
016452  000261                  15$:    SEC                             ;SET SUCCESS
016454  000207                  20$:    RTS     PC                      ;RETURN TO CALLER
                                        .SBTTL  XNXM    - CHECK FOR NONEXISTENT MEMORY
                                ;+
                                ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
                                ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
                                ;               "C" = 0, ALL ADDRESSES OK.
                                ;
                                ;CALL:  MOV ADR1,R1
                                ;       MOV ADR2,R2
                                ;       JSR PC,NXM
                                ;       RETURN                  ;TEST "C" AND PROCEED.
016456  012737  016510  000004  XNXM:   MOV     #2$,@#4                 ; SET BUSERR VECTOR.
016464  012737  000200  000006          MOV     #PRI04,@#6
016472  005003                          CLR     R3                      ;FLAG.
016474  005711                  1$:     TST     (R1)                    ;TEST THE ADDRESS(ES).
                                                                        ;IF ANY TRAP, CONTINUE AT 2$.
016476  020102                          CMP     R1,R2                   ;OTHERWISE, CONTINUE HERE.
016500  001407                          BEQ     3$                      ;BR IF FINISHED (NO NEXM'S).
016502  062701  000002                  ADD     #2,R1                   ;SET NEXT ADDRESS...
016506  000772                          BR      1$                      ;...AND CONTINUE.
016510  005103                  2$:     COM     R3                      ;GOT ONE, SET FLAG...
016512  012716  016520                  MOV     #3$,(SP)
016516  000002                          RTI                             ;...AND DISMISS INTERRUPT...
016520                          3$:     CLRVEC  #4                      ;...AND GIVE BACK THE VECTOR.
016520  012700  000004                  MOV     #4,R0
016524  104436                          TRAP    C$CVEC
016526  005703                          TST     R3                      ;DID WE CATCH ONE ??
016530  001401                          BEQ     .+4                     ;NO, "C" = 0, SKIP NEXT.
016532  000261                          SEC                             ;YES, "C" = 1, (R1) = NEXM ADDR.
016534  000207                          RTS     PC
```

```
                                        .SBTTL  TSTLOOP - CHECK ITERATION COUNT
                                ;*
                                ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
                                ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
                                ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
                                ;
                                ; CALL: LOOPTO  ARG
                                ;
016536                          TSTLOOP::
016536  005737  002166                  TST     NOITS           ; ITERATIONS INHIBITED?
016542  001006                          BNE     1$              ; YES.
016544  005737  002202                  TST     QVP             ; NO.
016550  100403                          BMI     1$              ;LOOPS DISALLOWED IN QUICK PASS.
016552  005337  002214                  DEC     LOOPCNT         ; BUMP LOOP COUNTER.
016556  001002                          BNE     2$
016560  000241                  1$:     CLC                     ;LOOP DISALLOWED, OR DONE.
016562  000401                          BR      3$
016564  000261                  2$:     SEC                     ;LOOP ENABLED.
016566  000207                  3$:     RTS     PC

                                        .SBTTL  TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
                                ;*
                                ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
                                ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
                                ; IN THE CURRENT RUN SEQUENCE.
                                ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
                                ;
                                ;INPUT:
                                ;
                                ;       R0      POINTER TO TEST ID ASCIZ STRING
                                ;
                                ;OUTPUT:
                                ;
                                ;       R5      ADDRESS OF FIRST DEVICE REGISTER
                                ;
                                ;IMPLICIT OUTPUTS:
                                ;
                                ;       TSTCNT  UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;       INTERRUPT LEVEL IS RAISED TO LEVEL OF
                                ;       THE DEVICE UNDER TEST
                                ;
                                ;-
                                TSTSETUP::
016570  010046                          MOV     R0,-(SP)        ;SAVE THE TEST ID MESSAGE
016572  005037  003154                  CLR     SIFLAG          ; CLEAR "SOFT INIT" FLAG
016576  005037  017036                  CLR     ERRK            ; CLEAR LOCAL ERROR COUNTER.
016602  005037  005772                  CLR     EXTA            ; CLEAR ERROR EXTENSION FLAG.
016606  105037  016224                  CLRB    INTMASK         ; CLEAR INTERRUPT MASK (CHECK ERROR)
016612  013700  002200                  MOV     UNITN,R0        ; GET THE UNIT NUMBER,
016616  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
016620  005737  003114                  TST     NODEV           ; DID STARTUP FIND THE DEVICE?
016624  001430                          BEQ     4$              ; BR IF YES
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  104455                          TRAP    C$ERDF
     016640  000001                          .WORD   1
     016642  003740                          .WORD   NXR
     016644  005736                          .WORD   NXRERR
2233 016646  000407                          BR      2$
2234 016650  052760  160001  003176  3$:     BIS     #160001,ERTABL(R0)      ; FLAG ERROR IN THE ERROR TABLE
2235 016656                                  ERRDF   2,NOINIT        ; DEVICE NOT IDLE
     016656  104455                          TRAP    C$ERDF
     016660  000002                          .WORD   2
     016662  004335                          .WORD   NOINIT
     016664  000000                          .WORD   0
2236 016666  012737  177777  003112  2$:     MOV     #-1,DUFLG       ; DROP THE UNIT
2237 016674                                  DODU    UNITN
     016674  013700  002200                  MOV     UNITN,R0
     016700  104451                          TRAP    C$DODU
2238 016702                                  DOCLN                   ; ABORT THE PASS
     016702  104444                          TRAP    C$DCLN
2239 016704  000423                          BR      5$
2240
2241 016706                          4$:     RFLAGS  R0              ; GET THE OPERATOR FLAGS.
     016706  104421                          TRAP    C$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  010046                          MOV     R0,-(SP)
     016722  012746  016764                  MOV     #TNAM,-(SP)
     016726  012746  000002                  MOV     #2,-(SP)
     016732  010600                          MOV     SP,R0
     016734  104417                          TRAP    C$PNTF
     016736  062706  000006                  ADD     #6,SP
2246 016742  005237  002212          1$:     INC     TSTCNT          ; BUMP TEST COUNTER.
2247 016746                                  SETPRI  IPRI            ;PRIORITY THAT OF DEVICE
     016746  013700  002210                  MOV     IPRI,R0
     016752  104441                          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                                          .EVEN
2253                                          .SBTTL  TSTEND  - PRINT ERRORS RECEIVED
2254                                  ;
2255                                  ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
2256                                  ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
2257                                  ;
2258 017000                          TSTEND: RFLAGS  R0
     017000  104421                          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  013746  017036                  MOV     ERRK,-(SP)
     017014  012746  017040                  MOV     #ESUM,-(SP)
     017020  012746  000002                  MOV     #2,-(SP)
     017024  010600                          MOV     SP,R0
     017026  104417                          TRAP    C$PNTF
```

TSTEND - PRINT ERRORS RECEIVED

```
        017030  062706  000006                  ADD     #6,SP
   2262 017034  000207                  1$:     RTS     PC
   2263
   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
```

```
                                        .SBTTL  INCERK  - INCREMENT LOCAL ERROR COUNT
                                ;*
                                ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
                                ;-
017124  005237  017036          INCERK: INC     ERRK            ; INCREMENT LOCAL ERROR COUNT
017130  010046                          MOV     R0,-(SP)        ; SAVE R0
017132  013700  002200                  MOV     UNITN,R0        ; GET UNIT NUMBER.
017136  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
017140  062700  003176                  ADD     #ERTABL,R0      ; R0 GETS ADDRESS OF ERROR TABLE ENTRY.
017144  005210                          INC     (R0)            ; INCREMENT THE DEVICE ERROR COUNT
017146  032710  007777                  BIT     #7777,(R0)      ; DID WE OVERFLOW THE FIELD?
017152  001001                          BNE     1$              ; BR IF NO.
017154  005310                          DEC     (R0)            ; YES -- BACK IT UP TO 7777.
017156  012600                  1$:     MOV     (SP)+,R0        ; RESTORE R0
017160  000207                          RTS     PC              ; RETURN TO CALLER

017162  010046                  CKEMAX: MOV     R0,-(SP)        ; SAVE R0
017164  013700  002200                  MOV     UNITN,R0        ; GET UNIT NUMBER
017170  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET
017172  016000  003176                  MOV     ERTABL(R0),R0   ; GET ERROR TABLE ENTRY
017176  042700  170000                  BIC     #170000,R0      ; EXTRACT ERROR COUNT FIELD
017202  020037  002172                  CMP     R0,GERRMAX      ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
017206  103004                          BHIS    1$              ; BR IF YES
017210  023737  017036  002170          CMP     ERRK,LERRMAX    ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
017216  103417                          BLO     2$              ; BR IF NO
017220                          1$:     RFLAGS  R0              ; GET OPERATOR FLAGS
017220  104421                          TRAP    C$RFLA
017222  032700  000040                  BIT     #IDU,R0         ; IS DROPPING INHIBITED?
017226  001013                          BNE     2$              ; BR IF YES.
017230  012737  177777  003112          MOV     #-1,DUFLG       ; NO -- DROP THE UNIT
017236                                  ERRDF   4,EMAXDU
017236  104455                          TRAP    C$ERDF
017240  000004                          .WORD   4
017242  017057                          .WORD   EMAXDU
017244  000000                          .WORD   0
017246                                  DODU    UNITN
017246  013700  002200                  MOV     UNITN,R0
017252  104451                          TRAP    C$DODU
017254                                  DOCLN
017254  104444                          TRAP    C$DCLN
017256  012600                  2$:     MOV     (SP)+,R0        ; RESTORE R0
017260  000207                          RTS     PC              ; RETURN TO CALLER
```

```
                                        .SBTTL  CKDROP  - CHECK IF UNIT SHOULD BE DROPPED
                                ;*
                                ; CHECK IF UNIT SHOULD BE DROPPED
                                ;-
017262  010046                  CKDROP: MOV     R0,-(SP)
017264                                  FORCERROR       1$,NOTSSR
017274                                  RFLAGS  R0
017274  104421                          TRAP    C$RFLA
017276  032700  000040                  BIT     #IDU,R0
017302  001010                          BNE     1$
017304  011600                          MOV     (SP),R0
017306  012737  177777  003112          MOV     #-1,DUFLG
017314                                  DODU    UNITN
017314  013700  002200                  MOV     UNITN,R0
017320  104451                          TRAP    C$DODU
017322                                  DOCLN                   ;ABORT THE PASS
017322  104444                          TRAP    C$DCLN
017324  012600                  1$:     MOV     (SP)+,R0
017326  000207                          RTS     PC

                                        .SBTTL  CONFIG  - DETERMINE CONFIGURATION OF SYSTEM
                                ;
                                ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
                                ;
017330                          CONFIG:
017330  004737  016054                  JSR     PC,SOFINIT
017334  000207                          RTS     PC
                                        .SBTTL  KTON,KTOFF      - ENABLE/DISABLE MEMORY MANAGEMENT
                                ;
                                ; SUBROUTINE - ENABLE MEM MGT.
                                ;
017336  005737  003132          KTON:   TST     KTFLG           ; GOT KT?
017342  001403                          BEQ     1$              ; NO.
017344  012737  000001  177572          MOV     #1,SR0          ; YES. ENABLE KT11.
017352  000207                  1$:     RTS     PC

                                ;
                                ; SUBROUTINE - DISABLE MEM MGT.
                                ;
017354  005737  003132          KTOFF:  TST     KTFLG           ; GOT KT11?
017360  001405                          BEQ     1$              ; NO.
017362  000240                          NOP
017364  000240                          NOP
017366  012737  000000  177572          MOV     #0,SR0          ; DISABLE KT.
017374  000207                  1$:     RTS     PC
```

SETMAP SETUP PAR6 MAPPING

```
                                        .SBTTL  SETMAP  - SETUP PAR6 MAPPING

                                ;+
                                ;
                                ;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
                                ;-
017376                          SETMAP:
017376                                  SAVREG                          ;SAVE R1-R4 UNTIL NEXT RETURN
017402  005737  003132                  TST     KTFLG                   ;SYSTEM HAVE ABOVE 28K?
017406  001433                          BEQ     10$                     ;BR IF NO
017410  010102                          MOV     R1,R2                   ;SAVE LOW ORDER BITS
        000006                          .REPT   6
                                        ASR     R0                      ;CONVERT WORD ADDRESS TO 32W BLOCKS
                                        ROR     R1                      ;MAKE IT DOUBLE PRECISION
                                        .ENDR
017442  042701  000177                  BIC     #177,R1                 ;ALINE FOR LOWER 4K BOUNDARY
017446  020137  003132                  CMP     R1,KTFLG                ;HIGHER THAN EXISTING MEMORY?
017452  103011                          BHIS    10$                     ;BR IF YES
017454  010137  172354                  MOV     R1,@#KIPAR6             ;SETUP MAPPING REGISTER PAR6
017460  042702  160000                  BIC     #160000,R2              ;SETUP DISPLACEMENT IN PAGE
017464  062702  140000                  ADD     #140000,R2              ;ADD IN PAR6 BIAS
017470  010200                          MOV     R2,R0                   ;RETURN IN R0
017472  000261                          SEC                             ;SET SUCCESS
017474  000401                          BR      15$                     ;
017476  000241                  10$:    CLC                             ;SET FAILURE
017500  000207                  15$:    RTS     PC                      ;RETURN
                                        .SBTTL  FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN
                                ;+
                                ;  FILL MEMORY WITH A BACKGROUND PATTERN
                                ;
                                ; INPUTS:
                                ;
                                ;       R0 = BACKGROUND PATTERN
                                ;       FREE    = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
                                ;       KTFLG   = SET TO HIGHEST MEMORY LOCATION IF > 28K.
                                ;
                                ; OUTPUTS:
                                ;
                                ;       NONE
                                ;-
                                ;
017502                          FILLMEM:
017502                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
017506  004737  017354                  JSR     PC,KTOFF                ;DISABLE KT.
```

K7

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                 10$:     MOV     R3,(R1)+        ;STORE A BACKGROUND WORD
2408 017526  005302                          DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
2409 017530  003375                          BGT     10$             ;BR IF NO
2410 017532  005737  003132                  TST     KTFLG           ; GOT KT?
2411 017536  001477                          BEQ     55$             ; 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                 30$:     MOV     R3,(R0)+        ;STORE TEST PATTERN IN >28K ADDRESS
2422 017624  020027  160000                  CMP     R0,#160000      ;END OF PAR6 MAPPING AREA?
2423 017630  103774                          BLO     30$             ;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     50$             ;BR IF YES
2428 017654  005737  003144                  TST     T23A            ;11/23A?
2429 017660  001407                          BEQ     35$             ;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     50$             ;EXIT IF THERE
2434 017700  005737  003146          35$:    TST     T23B            ;11/23B?
2435 017704  001410                          BEQ     45$             ;NO KEEP GOING
2436 017706  023727  172354  007600          CMP     @#KIPAR6,#7600  ;REACHED 18 BITS?
2437 017714  103001                          BHIS    40$             ;YES
2438 017716  000403                          BR      45$             ;NO KEEP GOING
2439 017720  012737  000020  172516  40$:    MOV     #20,SR3         ;SET 22 BIT RELOCATION
2440 017726  000137  017622          45$:    JMP     30$             ;KEEP GOING ON ETC.
2441 017732  004737  017354          50$:    JSR     PC,KTOFF        ; DISABLE KT.
2442 017736  000207                  55$:    RTS     PC
```

```
                                        .SBTTL  CMPMEM  - COMPARE MEMORY TO BACKGROUND PATTERN
                                ;+
                                ; COMPARE MEMORY WITH A BACKGROUND PATTERN
                                ;
                                ; INPUTS:
                                ;
                                ;       R0 = BACKGROUND PATTERN
                                ;       FREE    = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
                                ;       KTFLG   = SET TO HIGHEST MEMORY LOCATION IF > 28K.
                                ;
                                ; OUTPUTS:
                                ;
                                ;       CARRY   - SET IF NO ERROR
                                ;       CARRY   - CLR IF ERROR
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       ERRHI   - ERROR HIGH ADDRESS
                                ;       ERRLO   - ERROR LOW ADDRESS
                                ;       EXPD    - EXPECTED DATA
                                ;       RECV    - RECEIVED DATA
                                ;-
017740                          CMPMEM:
017740                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
017744  010003                          MOV     R0,R3           ;COPY TEST PATTERN
017746  004737  017354                  JSR     PC,KTOFF        ;DISABLE KT.
017752  013701  003124                  MOV     FREE,R1         ;GET FIRST FREE LOCATION
017756  013702  003126                  MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
017762  020311                  10$:    CMP     R3,(R1)         ;FREE SPACE LOCATION EQUAL TO EXPD?
017764  001411                          BEQ     15$             ;BR IF YES
017766  010137  002240                  MOV     R1,ERRLO        ;SAVE ADDRESS IN ERROR
017772  005037  002236                  CLR     ERRHI           ;NO HIGH ADDRESS
017776  010337  002232                  MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
020002  011137  002234                  MOV     (R1),RECV       ;SAVE RECV FOR ERROR REPORT
020006  000474                          BR      50$             ;
020010  005721                  15$:    TST     (R1)+           ;POINT TO NEXT ADDRESS
020012  005302                          DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
020014  003362                          BGT     10$             ;BR IF NO
020016  005737  003132                  TST     KTFLG           ; GOT KT?
020022  001472                          BEQ     55$             ; NO. GET OUT.
020024  004737  017336                  JSR     PC,KTON         ; YES. ENABLE KT.
020030  005000                          CLR     R0              ;HIGH ORDER ADDRESS START
020032  013701  003152                  MOV     PST32W,R1       ;GET >28K START ADDRESS (IN 32W BLOCKS)
        000006                          .REPT   6
                                        ROL     R1              ;CONVERT BLOCKS TO WORDS
                                        ROL     R0              ;MAKE IT DOUBLE PRECISION
                                        .ENDR
020066  042701  000177                  BIC     #177,R1         ;ALINE 4K BOUNDARY
020072  010046                          MOV     R0,-(SP)        ;SAVE HIGH ORDER
020074  010146                          MOV     R1,-(SP)        ;SAVE LOW ORDER
020076  004737  017376                  JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
020102  010004                          MOV     R0,R4           ;COPY ADDRESS BIASED TO PAR6
020104  012601                          MOV     (SP)+,R1        ;RESTORE LOW ORDER IN NON PAR6 FORMAT
020106  012600                          MOV     (SP)+,R0        ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
020110  020314                  30$:    CMP     R3,(R4)         ;ABOVE 28K LOCATION EQUAL EXPD?
020112  001411                          BEQ     32$             ;BR IF YES
020114  010037  002236                  MOV     R0,ERRHI        ;SAVE HIGH ORDER IN ERROR
```

M7

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 020220                          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
```

```
                                        .SBTTL  GETPAT  - GET 8 BIT PATTERN FROM OPERATOR
                                ;+
                                ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
                                ;
                                ;INPUTS:        NONE.
                                ;
                                ;OUTPUTS:
                                ;       R0      OCTAL NUMBER FROM THE OPERATOR
                                ;
                                ;CALLING SEQUENCE:
                                ;       JSR     PC,GETPAT
                                ;-
020254                          GETPAT::
020254                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
020260                          1$:     GMANID  DATASC,PATDAT,0,377,0,377,NO
020260  104443                          TRAP    C$GMAN
020262  000406                          BR      10000$
020264  020310                          .WORD   PATDAT
020266  000022                          .WORD   T$CODE
020270  020312                          .WORD   DATASC
020272  000377                          .WORD   377
020274  000000                          .WORD   T$LOLIM
020276  000377                          .WORD   T$HILIM
020300                          10000$:
020300                                  BNCOMPLETE      1$      ;RETRY IF ERROR
020300  103367                          BCC     1$
020302  013700  020310                  MOV     PATDAT,R0       ;DATA PATTERN FROM OPERATOR
020306  000207                          RTS     PC              ;RETURN TO CALLER

                                ;+
                                ;LOCAL DATA AREA
                                ;-

020310  000000                  PATDAT: .WORD   0               ;TEMPORARY STORAGE FOR DATA
020312      105     116     124 DATASC: .ASCIZ  'ENTER DATA PATTERN'
                                        .EVEN
```

```
                                .SBTTL  GETSEL  - ISSUE MENU AND GET OPERATOR RESPONSE
                                ;+
                                ;ROUTINE TO ISSUE A MENU AND GET THE OPERATOR'S RESPONSE.
                                ;
                                ;INPUTS:
                                ;       R0      ADDRESS OF ASCIZ STRING OF MENU
                                ;       R1      MAXIMUM ALLOWABLE OPERATOR RESPONSE
                                ;
                                ;OUTPUTS:
                                ;       R0      NUMBER OF THE OPERATOR'S SELECTION
                                ;-
020336                          GETSEL::
020336                                  SAVREG                          ;SAVE GENERAL REGISTERS
020342  010002                          MOV     R0,R2                   ;SAVE THE MENU ADDRESS
020344  010203                  1$:     MOV     R2,R3                   ;START OF MENU STRING
020346  005713                  2$:     TST     (R3)                    ;END OF ASCII ?
020350  001412                          BEQ     3$                      ;BRANCH IF ALL LINES DISPLAYED
020352                                  PRINTF  #SELASC,(R3)+           ;DISPLAY THE MENU
020352  012346                          MOV     (R3)+,-(SP)
020354  012746  020522                  MOV     #SELASC,-(SP)
020360  012746  000002                  MOV     #2,-(SP)
020364  010600                          MOV     SP,R0
020366  104417                          TRAP    C$PNTF
020370  062706  000006                  ADD     #6,SP
020374  000764                          BR      2$
020376                          3$:     GMANID  MENASC,MENRES,D,-1,0,-1,NO
020376  104443                          TRAP    C$GMAN
020400  000406                          BR      10001$
020402  020556                          .WORD   MENRES
020404  000042                          .WORD   T$CODE
020406  020527                          .WORD   MENASC
020410  177777                          .WORD   -1
020412  000000                          .WORD   T$LOLIM
020414  177777                          .WORD   T$HILIM
020416                          10001$:
020416                                  BNCOMPLETE      1$              ;RETRY IF ERROR
020416  103352                          BCC     1$
020420  013700  020556                  MOV     MENRES,R0               ;GET THE OPERATOR'S REPLY
020424  020001                          CMP     R0,R1                   ;COMPARE TO MAXIMUM ALLOWED
020426  101411                          BLOS    5$                      ;BRANCH IF OK
020430                                  PRINTF  #MENERR                 ;DISPLAY ERROR MESSAGE
020430  012746  020454                  MOV     #MENERR,-(SP)
020434  012746  000001                  MOV     #1,-(SP)
020440  010600                          MOV     SP,R0
020442  104417                          TRAP    C$PNTF
020444  062706  000004                  ADD     #4,SP
020450  000735                          BR      1$                      ;RETRY
020452  000207                  5$:     RTS     PC                      ;RETURN TO CALLER
020454     045     116     045  MENERR: .ASCIZ   '%N%A *** Menu Selection Too Large ***'
020522     045     116     045  SELASC: .ASCIZ   '%N%T'
020527     105     156     164  MENASC: .ASCIZ   'Enter Menu Selection: '
                                        .EVEN
020556  000000                  MENRES:         .WORD   0
```

```
                                        .SBTTL  CHKMAN  - CHECK MANUAL INTERVENTION LEGALITY
                                ;+
                                ;
                                ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
                                ;
                                ;INPUT:
                                ;
                                ;       NONE.
                                ;
                                ;OUTPUT:
                                ;
                                ;       CARRY   0       MANUAL INTERVENTION NOT ALLOWED
                                ;               1       MANUAL INTERVENTION IS OK
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;       A MESSAGE IS DISPLAYED WARNING THAT TEST IS
                                ;       NOT EXECUTED IF MANUAL INTERVENTION IS NOT
                                ;       ALLOWED.
                                ;
                                ;-

020560                          CHKMAN::
020560                                  SAVREG                  ;SAVE THE REGISTERS
020564                                  MANUAL                  ;SEE IF MANUAL INTERVENTION OK
020564  104450                          TRAP    C$MANI
020566                                  BCOMPLETE 1$            ;BRANCH IF ALLOWED
020566  103411                          BCS     1$
020570                                  PRINTF  #NOMAN          ;PRINT THE WARNING MESSAGE
020570  012746  020614                  MOV     #NOMAN,-(SP)
020574  012746  000001                  MOV     #1,-(SP)
020600  010600                          MOV     SP,R0
020602  104417                          TRAP    C$PNTF
020604  062706  000004                  ADD     #4,SP
020610  000241                          CLC                     ;CLEAR CARRY FOR ERROR
020612  000207                  1$:     RTS     PC              ;RETURN

020614      045     116     045 NOMAN:  .ASCIZ  '<N>A *** Manual Intervention not Allowed - Test Aborted ***'
                                        .even
```

```
                                        .SBTTL  ENVIRN  - SETUP FREE DIAGNOSTIC SPACE
                                ;
                                ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
                                ;
020710                          ENVIRN: MEMORY  R0
020710  104431                          TRAP    C$MEM
020712  010037  003124                  MOV     R0,FREE         ; GET 1ST FREE ADDRESS...
020716  062737  000002  003124          ADD     #2,FREE
020724  011037  003126                  MOV     (R0),FRESIZ     ;...AND WORD COUNT.
020730  162737  000004  003126          SUB     #4,FRESIZ
020736  013702  002012                  MOV     L$UNIT,R2       ; GET NUMBER OF UNITS
020742  162737  000007  003126  10$:     SUB     #7,FRESIZ       ; TAKE AWAY 7 WORDS PER UNIT
020750  005302                          DEC     R2
020752  001373                          BNE     10$
020754  013700  003124                  MOV     FREE,R0         ;GET FIRST FREE ADDRESS
020760  063700  003126                  ADD     FRESIZ,R0       ;POINT TO LAST FREE ADDRESS
020764  162700  000002                  SUB     #2,R0           ;BACKUP 1 WORD
020770  010037  003130                  MOV     R0,FREEHI       ;STORE LAST FREE ADDRESS
020774  000240                          NOP                     ;*****************************************
020776  012701  177520                  MOV     #BDVPCR,R1      ;GET BDV11 PCR ADDRESS
021002  010102                          MOV     R1,R2           ;COPY TO R2
021004  062702  000002                  ADD     #2,R2           ;SET THE RANGE
021010  004737  016456                  JSR     PC,XNXM         ;SEE IF WE HAVE ONE
021014  103001                          BCC     15$             ;OK TO SET FLAGS
021016  000423                          BR      40$             ;RETURN WITH FLAGS CLEAR
021020  013701  177520          15$:    MOV     BDVPCR,R1       ;SAVE PCR CONTENTS
021024  062701  000001                  ADD     #1,R1           ;ADD ONE TO IT
021030  012702  177520                  MOV     #BDVPCR,R2      ;GET BDV11 PCR ADDRESS
021034  005212                          INC     (R2)            ;TRY TO WRITE TO IT
021036  013703  177520                  MOV     BDVPCR,R3       ;GET RESULTS
021042  020103                          CMP     R1,R3           ;DID IT CHANGE?
021044  001006                          BNE     20$             ;NO, MUST BE 11/23B
021046  005237  003144                  INC     T23A            ;SET THE FLAG
021052  042737  170000  002120          BIC     #170000,L$HIME  ;SUPERVISOR COULD BE WRONG
                                        NOP                     ;BR 40$ FOR RELEASE
                                ;       PRINTF  #M8186          ;TELL THE SYSTEM TYPE
021060  000402                          BR      40$             ;RETURN
021062  005237  003146          20$:    INC     T23B            ;SET THE FLAG
                                ;       NOP                     ;BR 40$ FOR RELEASE
                                ;       PRINTF  #M8189          ;TELL THE SYSTEM TYPE
021066                          40$:
021066  000207                          RTS     PC              ;RETURN
```

```
                                        .SBTTL  KTINIT  - SETUP KT11 MEMORY MANAGEMENT REGISTERS
                                ;+
                                ;
                                ;ROUTINE TO INIT KT-11
                                ;
                                ;-
021070                          KTINIT:
021070  005037  003132                  CLR     KTFLG           ; INIT >28K MEMORY FLAG
021074  005037  003134                  CLR     KTENABLE        ; INIT TEST >28K FLAG
021100  023727  002120  001577          CMP     L$HIME,#1577    ; GOT ENOUGH MEMORY (>28K)?
021106  101444                          BLOS    9$              ; NO.
021110  013700  000004                  MOV     @#ERRVEC,R0     ; SAVE OLD ERR VEC PTR.
021114  012737  021206  000004          MOV     #2$,@#ERRVEC    ; SET ERR VEC PTR.
021122  005737  177572                  TST     @#SR0           ; GOT KT11?
021126  000240                          NOP                     ;  (TRAP IF NO).
021130  013737  002120  003132          MOV     L$HIME,KTFLG    ; YES. SET KT FLAG.
021136  042737  000177  003132          BIC     #177,KTFLG      ;
021144  010037  000004                  MOV     R0,@#ERRVEC     ; RESTORE OLD FRR VEC PTR.
021150  005000                          CLR     R0              ; R0 = AR DATA.
021152  012701  172340                  MOV     #KIPAR0,R1      ; R1 = KI REGS PTR.
021156  012761  077406  177740  1$:     MOV     #77406,-40(R1)  ; SET DESCRIPTOR REG.
021164  010021                          MOV     R0,(R1)+        ; SET KIPAR REG.
021166  062700  000200                  ADD     #200,R0         ; BUMP AR DATA BY "4K".
021172  020027  002000                  CMP     R0,#2000        ; AT "I/O"?
021176  001367                          BNE     1$              ; NO.
021200  012741  177600                  MOV     #177600,-(R1)   ; YES. SET KTPAR7 FOR I/O.
021204  000405                          BR      9$
                                ;
021206  012716  021214          2$:     MOV     #6$,(SP)        ; SET UP RETURN
021212  000002                          RTI                     ; RTI TO NEXT LOCATION
                                ;
021214  010037  000004          6$:     MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
                                ;
021220  000207                  9$:     RTS     PC
```

```
                                        ;*
                                        ;       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.
                                        ;
                                        ;-

021222                                  INVERT::

021222  005737  002224                          TST     EXTFEA                  ; IS SWITCH SET?
021226  001020                                  BNE     1$                      ; YES,EXIT STAGE RIGHT!(or the next one outa town!)
021230  012737  100206  021274                  MOV     #100206,CMDPKT          ; WRT SUB-SYS MEM CMD
021236  012737  021304  021276                  MOV     #WSMBK,CMDPKT+2         ; MSG BUF ADDR
021244  012737  000006  021302                  MOV     #6,CMDPKT+6             ; BYTE COUNT
021252  012737  100010  021304                  MOV     #100010,WSMBK           ; INVERT THE SWITCH
021260  012704  021274                          MOV     #CMDPKT,R4              ; SET CMDPKT INTO R4
021264  004737  010662                          JSR     PC,WRTCHR               ; DO IT
021270  000207                          1$:     RTS     PC                      ; RETURN

                                        ;       COMMAND PACKET.

021274                                  .       =       <.+3>&177774    ;MUST BE ON MOD 4 BOUNDRY.

021274  000000                          CMDPKT:: 0                      ;1ST WORD IS TSO5 COMMAND.
021276  000000                                    0                     ;2ND WORD IS THE BUFFER LOW ADDRESS.
021300  000000                                    0                     ;3RD WORD IS THE BUFFER HIGH ADDRESS.
021302  000000                                    0                     ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.

                                        ;       WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.

021304  000000                          WSMBK:: 0                       ;1ST WORD:: SEL 0
021306  000000                                  0                       ;2ND WORD:: SEL 2
021310  000000                                  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.
                                        ;-

021312                                  MEMCK::

021312                                          SAVREG                          ;SAVE THE REGISTERS
021316  005037  003136                          CLR     NXMFLG                  ;CLEAR THE FLAG
021322  005037  003140                          CLR     NXMLO                   ;CLEAR THE TEST ADDRESS LO
021326  005037  003142                          CLR     NXMHI                   ;CLEAR THE TEST ADDRESS HI
021332  005737  003146                          TST     T23B                    ;IS IT A 11/23B?
```

```
 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          1$:     TST     T23A                    ;IS IT A 11/23A?
 2789 021362  001413                          BEQ     4$                      ;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          13$:    INC     NXMFLG                  ;SET THE FLAG
 2801 021440  000411                          BR      15$                     ;EXIT
 2802 021442  000410                  14$:    BR      15$                     ;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                                         .TITLE  TSV4   MISCELLANEOUS SECTIONS
      8
      9 021562                                  BGNMOD  TSV4
        021562                          TSV4::
     10
     16
     17
     18
     19                                         .SBTTL  PROTECTION TABLE
     20 021562                                  BGNPROT
        021562                          L$PROT::
     21 021562  177777  177777  177777         .WORD   -1, -1, -1, -1          ;NO DEVICE PROTECTION REQUIRED.
     22 021572                                  ENDPROT
```

INITIALIZE SECTION

```
                                        .SBTTL  INITIALIZE SECTION

                                ;++
                                ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ;AT THE BEGINNING OF EACH PASS.
                                ;
                                ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
                                ;IF "CONTINUE", NOTHING IS REQUIRED.
                                ;
                                ;--
                                ;+
                                ;INSERT TEMPORARY JUMP TO ODT
                                ;-
021572                                  BGNINIT
021572                          L$INIT::
021572  005037  002224          40$:    CLR     EXTFEA
021576  005037  003136                  CLR     NXMFLG
021602  012737  006360  002176          MOV     #EPRT1,EPRTSW           ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
021610  005037  003154                  CLR     SIFLAG                  ;CLEAR "SOFT INIT" FLAG
021614  005037  003134                  CLR     KTENABLE                ;CLEAR TEST ABOVE 28K FLAG
021620  005037  002302                  CLR     RAMSIZ                  ;CLEAR RAM SIZE FOR RAMERR ROUTINE
021624                                  READEF  #EF.CONTINUE
021624  012700  000036                  MOV     #EF.CONTINUE,R0
021630  104447                          TRAP    C$REFG
021632                                  BNCOMPLETE 1$
021632  103023                          BCC     1$
021634  023737  002200  002012          CMP     UNITN,L$UNIT            ;UNIT IN RANGE?
021642  103070                          BHIS    4$                      ;BR IF NO.
021644  005737  003112                  TST     DUFLG                   ;DROPPED UNIT?
021650  100472                          BMI     NXTU                    ;BR IF YES
021652  013701  002200                  MOV     UNITN,R1
021656  006301                          ASL     R1
021660  005761  003176                  TST     ERTABL(R1)
021664  001516                          BEQ     SETU
021666  032761  040000  003176          BIT     #BIT14,ERTABL(R1)       ;DROPPED?
021674  001060                          BNE     NXTU
021676                                  EXIT    INIT                    ;DO NOTHING IF "CONTINUE".
021676  104432                          TRAP    C$EXIT
021700  000416                          .WORD   L10030-.
021702                          1$:     READEF  #EF.NEW
021702  012700  000035                  MOV     #EF.NEW,R0
021706  104447                          TRAP    C$REFG
021710                                  BNCOMPLETE NXTU                 ;TAKE NEXT UNIT IF NOT NEW PASS.
021710  103052                          BCC     NXTU
021712                                  READEF  #EF.START
021712  012700  000040                  MOV     #EF.START,R0
021716  104447                          TRAP    C$REFG
021720                                  BCOMPLETE 2$
021720  103404                          BCS     2$
021722                                  READEF  #EF.RESTART
021722  012700  000037                  MOV     #EF.RESTART,R0
021726  104447                          TRAP    C$REFG
021730                                  BNCOMPLETE 31$
021730  103031                          BCC     31$
021732                          2$:                                     ;1ST PASS, BUS-INIT...
021732                                  BRESET                          ;BUS RESET.
021732  104433                          TRAP    C$RESET
```

J8

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$,-(SP)              ;RETURN TO DEBUGGER
    71                                      ;    JMP     @.ODT           ;;@@ENTER THE DEBUGGER
    72 021754  005037  003400                    CLR     SKIPT           ;CLEAR THE SUBTEST "SKIPPER"
    73 021760                           20$:
    74 021760  012737  177777  002202            MOV     #-1,@VP                 ;...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,R0
    78 022002  005020                      30$:  CLR     (R0)+                   ;CLEAR THE ERROR TABLE
    79 022004  020027  003376                    CMP     R0,#ERTABE
    80 022010  103774                            BLO     30$
    81 022012  000404                            BR      4$
    82 022014  005037  002202                31$:  CLR     @VP
    83 022020  000137  022070                    JMP     PASRPT                  ;GO REPORT THE STATUS
    84
    85 022024                           4$:
    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
       022036  104422                            TRAP    C$BRK
    89 022040  005237  002200                    INC     UNITN                   ;...AND SET NEXT UNIT NUMBER.
    90 022044  023737  002200  002012            CMP     UNITN,L$UNIT
    91 022052  103423                            BLO     SETU
    92 022054  012737  177777  003112            MOV     #-1,DUFLG
    93 022062  000401                            BR      11$
    94 022064                                   DOCLN                           ;ABORT, NO MORE UNITS.
       022064  104444                            TRAP    C$DCLN
    95 022066  000240                      11$:  NOP
    96 022070                           PASRPT:
    97 022070  023727  002012  000001            CMP     L$UNIT,#1               ;HOW MANY UNITS SELECTED?
    98 022076  101752                            BLOS    NEWPAS                  ;BR IF ONLY 1
    99 022100  005737  002216                    TST     DEVCNT                  ;ARE ANY STILL RUNNING?
   100 022104  001747                            BEQ     NEWPAS                  ;BR IF NO
   101 022106                                   RFLAGS  R0
       022106  104421                            TRAP    C$RFLA
   102 022110  032700  000100                    BIT     #ISR,R0                 ;SHOULD WE PRINT STATISTICS
   103 022114  001343                            BNE     NEWPAS                  ;BR IF NO
   104
   105 022116                                   DORPT
       022116  104424                            TRAP    C$DRPT
   106 022120  000741                            BR      NEWPAS
   107 022122                           10$:
   108
   109 022122                           SETU:   GPHARD  UNITN,R0                ;GET UNIT N P-TABLE POINTER.
       022122  013700  002200                    MOV     UNITN,R0
       022126  104442                            TRAP    C$GPHRD
   110 022130                                   BNCOMPLETE NXTU                 ;BR IF UNIT NOT AVAILABLE.
       022130  103342                            BCC     NXTU
   111 022132  005037  003112                    CLR     DUFLG                   ;CLEAR "DROPPED" FLAG.
   112 022136  005237  002216                    INC     DEVCNT
   113 022142  012001                            MOV     (R0)+,R1                ;GET 1ST REGISTER ADDRESS.
   114 022144  010137  002204                    MOV     R1,CSRADDR              ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
```

```
INITIALIZE SECTION

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

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


                                        ;
                                        ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
                                        ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
                                        ;
                                                MOV     UNITN,R1
                                                ASL     R1
                                                BIS     #BIT15,ERTABL(R1)       ;SAY DEVICE RUNNING
                                                CLR     EXTA                    ;CLEAR ERROR EXTENSION FLAG.
                                                CMP     L$UNIT,#1               ;ARE WE TESTING MULTIPLE UNITS?
                                                BLOS    10$                     ;BR IF NO.
                                                RFLAGS  R0                      ;YES -- GET OPERATOR FLAGS.
                                                TRAP    C$RFLA
                                                BIT     #PNT,R0                 ;SHOULD WE PRINT UNIT #?
                                                BEQ     10$                     ;BR IF NOT.
                                                PRINTF  #PUNIT,UNITN            ;PRINT THE UNIT #
                                                MOV     UNITN,-(SP)
                                                MOV     #PUNIT,-(SP)
                                                MOV     #2, (SP)
                                                MOV     SP,R0
                                                TRAP    C$PNTF
                                                ADD     #6,SP
                                        10$:
                                                CLR     NODEV
                                                MOV     CSRADDR,R1              ;ADDRESS OF FIRST REGISTER
                                                MOV     R1,R2                   ;START OF REGISTERS
                                                ADD     #TSSR,R2                ;ADDRESS OF TSSR REGISTER
                                                JSR     PC,XNXM                 ;TEST BOTH CONTROLLER REGISTERS...
                                                BCC     2$                      ;...AND BR IF ALL OK.
                                                MOV     R1,NODEV                ;FLAG DEVICE AS NON-EXISTENT
                                                MOV     #-1,DUFLG               ;DROP THIS UNIT.
                                        2$:
                                        ;
                                        ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
                                        ;
                                        5$:     SETPRI  #PRI00          ;ENABLE INTERRUPTS.
                                                MOV     #PRI00,R0
                                                TRAP    C$SPRI
                                                ENDINIT
                                        L10030:
                                                TRAP    C$INIT

                                        PUNIT:  .ASCIZ  /%N%N%A***** TESTING UNIT %D2%A *****/
                                                .EVEN
```

ADD AND DROP UNITS SECTIONS

```
                                        .SBTTL  ADD AND DROP UNITS SECTIONS

                                ;**
                                ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
                                ;    OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
                                ;--
022366                                  BGNAU
022366                          L$AU::
022366  010001                          MOV     R0,R1                   ; GET UNIT TO BE ADDED (R0)
022370  006301                          ASL     R1                      ; MAKE IT A WORD INDEX
022372  052761  100000  003176          BIS     #100000,ERTABL(R1)      ; SET THE "ACTIVE" BIT
022400  042761  040000  003176          BIC     #40000,ERTABL(R1)       ; CLEAR THE "DROPPED" BIT
022406                                  PRINTF  #1$,R0
022406  010046                          MOV     R0,-(SP)
022410  012746  022434                  MOV     #1$,-(SP)
022414  012746  000002                  MOV     #2,-(SP)
022420  010600                          MOV     SP,R0
022422  104417                          TRAP    C$PNTF
022424  062706  000006                  ADD     #6,SP
022430                                  EXIT    AU
022430  000167                          .WORD   J$JMP
022432  000026                          .WORD   L10031-2-.
022434    045     116     045   1$:     .ASCIZ  /%N%A  UNIT %D%A ADDED/
                                        .EVEN

022462                                  ENDAU           ; UNUSED.
022462                          L10031:
022462  104452                          TRAP    C$AU
                                ;**
                                ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ; TO BE REMOVED FROM THE TEST LIST.
                                ;
                                ; SUPVSR DOES THE "DROPPING".  THIS IS JUST TO TELL THE MAN.
                                ; "DROPPED" UNITS ARE RE-SELECTED ON OPERATOR "STA" OR "ADD"
                                ; COMMAND, OTHERWISE REMAIN INACTIVE.  THE "DISPLAY" COMMAND
                                ; WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
                                ; WHICH ARE STILL ACTIVE.
                                ; UPON ENTRY, R0 CONTAINS THE UNIT TO BE DROPPED.

022464                                  BGNDU
022464                          L$DU::
022464  012737  177777  003112          MOV     #-1,DUFLG
022472  010001                          MOV     R0,R1
022474  006301                          ASL     R1
022476  052761  140000  003176          BIS     #140000,ERTABL(R1)      ; SAY DROPPED
022504  000240  000240  000240          240,240,240                     ; ?????????
022512                                  PRINTF  #1$,R0
022512  010046                          MOV     R0,-(SP)
022514  012746  022540                  MOV     #1$,-(SP)
022520  012746  000002                  MOV     #2,-(SP)
022524  010600                          MOV     SP,R0
022526  104417                          TRAP    C$PNTF
022530  062706  000006                  ADD     #6,SP
022534                                  EXIT    DU
022534  000167                          .WORD   J$JMP
022536  000030                          .WORD   L10032-2-.
```

M8

ADD AND DROP UNITS SECTIONS

```
    197 022540      045     116     045 1$:     .ASCIZ  /%N%A  UNIT %D%A DROPPED/
    198                                 .EVEN
    199 022570                          ENDDU
        022570                      L10032:
        022570  104453                 TRAP    C$DU
    200                             ;++
    201                             ; AUTO-DROP CODE SECTION.
    202                             ;--
    203 022572                          BGNAUTO
        022572                      L$AUTO::
    204 022572  013705  002204          MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
    205 022576  012703  000550          MOV     #360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
    206 022602  004737  016330      10$:    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
    207 022606  103420                  BCS     20$                     ;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     L$DLY,(PC)+
        022622  000000                  .WORD   0
        022624  005367  177772          DEC     -6(PC)
        022630  001375                  BNE     .-4
        022632  005367  177756          DEC     -22(PC)
        022636  001367                  BNE     .-20
    209 022640  005303                  DEC     R3                      ;BUMP COUNTER DOWN
    210 022642  001357                  BNE     10$                     ;KEEP GOING
    211 022644  004737  017262          JSR     PC,CKDROP               ;TRY AND DROP UNIT
    212 022650                      20$:
    213 022650                          ENDAUTO                     ; UNUSED.
        022650                      L10033:
        022650  104461                 TRAP    C$AUTO
```

CLEAN-UP AND REPORT CODING SECTIONS

```
                                        .SBTTL  CLEAN-UP AND REPORT CODING SECTIONS


                                ;**
                                ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
                                ; EXECUTED AT THE END OF EACH PASS (OR SUB-PASS).
                                ; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
                                ;--
022652                                  BGNCLN
022652                          L$CLEAN::
022652  013705  002204                  MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
022656  005737  003112                  TST     DUFLG           ;"DROPPED" FLAG IS SET ON...
022662  100405                          BMI     1$                      ;...AND GROSS CONTROLLER FAULT...
                                                                        ;...DON'T TRY TO XCT CLEANUP CODE.

022664  012765  000000  000002          MOV     #0,TSSR(R5)     ;DO SOFT INIT
022672  004737  016330                  JSR     PC,WAITF
022676                          1$:
022676                          2$:     ENDCLN
022676                          L10034:
022676  104412                          TRAP    C$CLEAN

                                ;**
                                ; THE REPORT CODING SECTION CONTAINS THE
                                ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;--
022700                                  BGNRPT
022700                          L$RPT::
022700                                  PRINTS  #DEVSUM
022700  012746  023142                  MOV     #DEVSUM,-(SP)
022704  012746  000001                  MOV     #1,-(SP)
022710  010600                          MOV     SP,R0
022712  104416                          TRAP    C$PNTS
022714  062706  000004                  ADD     #4,SP
022720  010246                          MOV     R2,-(SP)
022722  010346                          MOV     R3,-(SP)
022724  010446                          MOV     R4,-(SP)
022726  012704  003176                  MOV     #ERTABL,R4      ; GET START OF ERROR TABLE.
022732  005003                          CLR     R3              ; CLEAR UNIT NUMBER
022734  011402                  1$:     MOV     (R4),R2         ; GET ERROR TABLE ENTRY & TEST IT.
022736  001467                          BEQ     4$              ; ZERO IF UNIT NOT RUN
022740  100066                          BPL     4$
022742  032702  040000                  BIT     #BIT14,R2       ; WAS UNIT DROPPED?
022746  001015                          BNE     2$              ; BR IF YES
022750  042702  170000                  BIC     #^C7777,R2      ; GET ERROR COUNT FIELD
022754                                  PRINTS  #DEVONL,R3,R2   ; PRINT
022754  010246                          MOV     R2,-(SP)
022756  010346                          MOV     R3,-(SP)
022760  012746  023177                  MOV     #DEVONL,-(SP)
022764  012746  000003                  MOV     #3,-(SP)
022770  010600                          MOV     SP,R0
022772  104416                          TRAP    C$PNTS
022774  062706  000010                  ADD     #10,SP
023000  000446                          BR      4$
023002  020227  160000          2$:     CMP     R2,#160000      ; WAS UNIT NON-EXISTENT?
023006  001012                          BNE     3$              ; BR IF NO
023010                                  PRINTS  #DEVNXR,R3
023010  010346                          MOV     R3,-(SP)
023012  012746  023247                  MOV     #DEVNXR,-(SP)
```

```
                023016  012746  000002                  MOV     #2,-(SP)
                023022  010600                          MOV     SP,R0
                023024  104416                          TRAP    C$PNTS
                023026  062706  000006                  ADD     #6,SP
            254 023032  000431                          BR      4$
            255 023034  020227  160001          3$:     CMP     R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
            256 023040  001012                          BNE     30$             ; 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$PNTS
                023060  062706  000006                  ADD     #6,SP
            258 023064  000414                          BR      4$
            259 023066  042702  170000          30$:    BIC     #^C7777,R2
            260 023072                                  PRINTS  #DEVDRO,R3,R2
                023072  010246                          MOV     R2,-(SP)
                023074  010346                          MOV     R3,-(SP)
                023076  012746  023412                  MOV     #DEVDRO,-(SP)
                023102  012746  000003                  MOV     #3,-(SP)
                023106  010600                          MOV     SP,R0
                023110  104416                          TRAP    C$PNTS
                023112  062706  000010                  ADD     #10,SP
            261 023116  062704  000002          4$:     ADD     #2,R4
            262 023122  005203                          INC     R3
            263 023124  020427  003376                  CMP     R4,#ERTABE
            264 023130  103701                          BLO     1$
            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 DEVDRO: .ASCIZ  /%A  UNIT %D3%A    DROPPED, ERRORS = %D%N/
            275                                         .EVEN
            276
            277 023462                                  ENDMOD
            278
```

CLEAN-UP AND REPORT CODING SECTIONS

```
                                        .TITLE  TSV5A - HARDWARE TESTS
                                        BGNMOD  TSV5
023462                          TSV5::
                                        .SBTTL  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.
                                        BGNTST
                                                                                T1::
023462  012700  023660                  MOV     #TST1ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
023466  004737  016570                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
023472  012737  000024  002214          MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
023500                          T1LOOP:
023500  005003                          CLR     R3                      ;USE R3 AS FATAL ERROR FLAG

023502                                  BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
                                                                                T1.1:
023502  104402                                                                  TRAP    C$BSUB
```

TEST 1: BUS RESET TEST

```
     76
     77 023504                                  BRESET                          ;ISSUE A BUS RESET
        023504  104433                                                                  TRAP    C$RESET
     78 023506  004737  016330                  JSR     PC,WAITF                ;WAIT FOR READY
     79 023512  016501  000002                  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
     80 023516  010102                          MOV     R1,R2                   ;CONTENTS OF TSSR
     81 023520  042702  176277                  BIC     #^C<HIADDR!OFL>,R2      ;THESE BITS MAY BE SET
     82 023524  052702  002200                  BIS     #SSR!NBA,R2             ;READY AND NEW DATA SHOULD BE SET
     83 023530  020102                          CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
     84 023532  001405                          BEQ     10$                     ;BRANCH IF COMPARE
     88 023534                                  ERRDF   ERRNO,SFHERR,SFFMSG     ;REPORT A FATAL ERROR
        023534  104455                                                                  TRAP    C$ERDF
        023536  000145                                                                  .WORD   101
        023540  003705                                                                  .WORD   SFHERR
        023542  012152                                                                  .WORD   SFFMSG
     89 023544  005203                          INC     R3                      ;SET THE FATAL ERROR FLAG
     90 023546                          10$:
     91 023546                                  ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
        023546                                                                      L10037:
        023546  104403                                                                  TRAP    C$ESUB
     92
     93 023550  005703                          TST     R3                      ;DID WE HAVE FATAL ERROR ?
     94 023552  001402                          BEQ     20$                     ;BRANCH IF NOT
     95 023554  004737  017262                  JSR     PC,CKDROP               ;GO DROP THIS UNIT, IF ALLOWED
     96 023560  005003                  20$:    CLR     R3                      ;RESET FATAL ERROR FLAG
     97
     98
     99 023562                                  BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
        023562                                                                      T1.2:
        023562  104402                                                                  TRAP    C$BSUB
    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     #^C<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                          CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
    108 023614  001405                          BEQ     10$                     ;BRANCH IF COMPARE
    112 023616                                  ERRDF   ERRNO,SFIERR,SFFMSG     ;REPORT A FATAL ERROR
        023616  104455                                                                  TRAP    C$ERDF
        023620  000146                                                                  .WORD   102
        023622  003652                                                                  .WORD   SFIERR
        023624  012152                                                                  .WORD   SFFMSG
    113 023626  005203                          INC     R3                      ;SET THE ERROR FLAG
    114 023630                          10$:
    115 023630                                  ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
        023630                                                                      L10040:
        023630  104403                                                                  TRAP    C$ESUB
    116
    117
    118 023632  005703                          TST     R3                      ;FATAL ERROR DETECTED ?
    119 023634  001402                          BEQ     20$                     ;BRANCH IF NOT
    120 023636  004737  017262                  JSR     PC,CKDROP               ;SEE IF TIME TO DROP UNIT
    121 023642  004737  016536          20$:    JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
    122 023646  103002                          BCC     40$                     ;BRANCH IF NOT
    123 023650  000137  023500                  JMP     T1LOOP                  ;LOOP UNTIL COUNT EXPIRED
```

E9

TEST 1: BUS RESET TEST

```
    124 023654                          40$:    EXIT    TST                     ;ALL DONE THIS TEST
        023654  104432                                                                  TRAP    C$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                                                          L10036:
        023700  104401                                                         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 DC005 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 023702                              BGNTST
        023702                                                                  T2::
    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              5$:     TST     R3                      ;DO WE NEED SOFT INIT
```

F9

TEST 2: WRAP DATA - HIGH BYTE

```
180 023730  001412                        BEQ     10$                     ;BRANCH IF NOT
181 023732  005003                        CLR     R3                      ;DON'T NEED INIT NEXT TIME THRU
182 023734  004737  016054                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
183 023740  103406                        BCS     10$                     ;BR IF SOFT INIT = OK
187 023742  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
188 023744                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
    023744  104455                                                                  TRAP    C$ERDF
    023746  000311                                                                  .WORD   201
    023750  003652                                                                  .WORD   SFIERR
    023752  012104                                                                  .WORD   SFIMSG
189 023754  005203                        INC     R3                      ;FORCE SOFT INIT ON NEXT PASS
190 023756  005037  002220          10$:  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
191
192 023762                                BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
    023762  104404                                                                  TRAP    C$BSEG
193
194 023764  110465  000001                MOVB    R4,TSDBH(R5)            ;SET MAINT MODE + WRITE DATA
195 023770  004737  016330                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
196 023774  103411                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
197 023776  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
198 024000  010402                        MOV     R4,R2                   ;DATA THAT WAS WRITTEN
202 024002                                ERRDF   ERRNO,T2SSR,EXPREC      ;DEVICE FATAL SSR FAILED TO SET
    024002  104455                                                                  TRAP    C$ERDF
    024004  000312                                                                  .WORD   202
    024006  024276                                                                  .WORD   T2SSR
    024010  015554                                                                  .WORD   EXPREC
203 024012  005203                        INC     R3                      ;FORCE SOFT INIT ON NEXT PASS
204 024014  005237  002220                INC     FATFLG                  ;SET FATAL ERROR FLAG
205 024020                          15$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
    024020  104406                                                                  TRAP    C$CLP1
206 024022  005737  002220                TST     FATFLG                  ;WAS FATAL ERROR RECEIVED ?
207 024026  001402                        BEQ     20$                     ;BRANCH IF NOT
208 024030  004737  017262                JSR     PC,CKDROP               ;SEE IF TIME TO DROP UNIT
209 024034  010402                  20$:  MOV     R4,R2                   ;DATA PATTERN WRITTEN
210 024036  042702  177774                BIC     #^C<BIT0!BIT1>,R2       ;CLEAR ALL BUT LOW 2 BITS
211 024042  000302                        SWAB    R2                      ;BITS 8 AND 9 HAVE LOW DATA BITS
212 024044  052702  002200                BIS     #SSR!NBA,R2             ;THESE BITS MUST BE SET ALSO
213 024050  016501  000002                MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
214 024054  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
215 024060  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
216 024062  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
217 024066  020201                  25$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
218 024070  001405                        BEQ     30$                     ;OKAY IF MATCH
222 024072                                ERRHRD  ERRNO,T2TSSR,EXPREC     ;TSSR WASN'T CORRECT
    024072  104456                                                                  TRAP    C$ERHRD
    024074  000313                                                                  .WORD   203
    024076  024231                                                                  .WORD   T2TSSR
    024100  015554                                                                  .WORD   EXPREC
223 024102  005203                        INC     R3                      ;FORCE SOFT INIT ON NEXT PASS
224 024104                          30$:  CKLOOP                          ;LOOP ON ERROR ?
    024104  104406                                                                  TRAP    C$CLP1
225 024106  016501  000000                MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
226 024112  005002                        CLR     R2                      ;
227 024114  150402                        BISB    R4,R2                   ;DATA PATTERN WRITTEN
228 024116  000302                        SWAB    R2                      ;MOVE INTO TOP BYTE
229 024120  150402                        BISB    R4,R2                   ;BOTH HALVES SHOULD BE SAME
230 024122  020102                        CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
```

```
TEST  2: WRAP DATA - HIGH BYTE

    231 024124  001405                          BEQ     35$                     ;ERROR IF NOT EQUAL
    235 024126                                  ERRHRD  ERRNO,T2TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
        024126  104456                                                              TRAP    C$ERHRD
        024130  000314                                                              .WORD   204
        024132  024164                                                              .WORD   T2TSBA
        024134  015554                                                              .WORD   EXPREC
    236 024136  005203                          INC     R3                      ;FORCE SOFT INIT ON NEXT PASS
    237
    238 024140                          35$:    ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        024140                                                                      10000$:
        024140  104405                                                              TRAP    C$ESEG
    239
    240 024142  105204                          INCB    R4                      ;NEXT DATA PATTERN TO TEST
    241 024144  001270                          BNE     5$                      ;BRANCH TILL BACK TO ZERO
    242 024146  004737  016536                  JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
    243 024152  103002                          BCC     40$                     ;BRANCH IF NOT
    244 024154  000137  023720                  JMP     T2LOOP                  ;LOOP UNTIL COUNT EXPIRED
    245 024160                          40$:    EXIT    TST                     ;ALL DONE THIS TEST
        024160  104432                                                              TRAP    C$EXIT
        024162  000214                                                              .WORD   L10041-.
    246
    247                                 ;+
    248                                 ;LOCAL TEXT MESSAGES FOR TEST
    249                                 ;-
    250
    251 024164      124     123     102 T2TSBA: .ASCIZ  'TSBA Incorrect After TSDB High Write'
    252 024231      124     123     123 T2TSSR: .ASCIZ  'TSSR Incorrect After TSDB High Write'
    253 024276      116     157     040 T2SSR:  .ASCIZ  'No Sub-System Ready After TSDB High Write'
    254 024350      127     162     141 TST2ID: .ASCIZ  'Wrap Data - High Byte'
    255                                         .EVEN
    256 024376                                  ENDTST
        024376                                                                  L10041:
        024376  104401                                                              TRAP    C$ETST
    257
    258                                 .SBTTL  TEST  3: WRAP DATA - LOW BYTE
    259
    260                             ;       THIS TEST FURTHER VERIFIES OPERATION OF MANY OF THE SAME
    261                             ;       ELEMENTS TESTED IN TEST 2, AND ADDITIONALLY VERIFIES:
    262                             ;
    263                             ;
    264                             ;            1.  LOW BYTE OF THE TSDB INPUT FILE REGISTER,
    265                             ;
    266                             ;            2.  LOW BYTE OF INTERNAL DAL BUS DRIVERS ON THE DC005
    267                             ;                TRANSCEIVER CIRCUITS,
    268                             ;
    269                             ;            3.  BASIC FUNCTIONING OF PARTS OF THE RAM.
    270                             ;
    271                             ;
    272                             ;
    273                             ;       THE PROGRAM WRITES A TEST DATA BYTE INTO THE LOW BYTE OF TSDB,
    274                             ;       WAITS FOR THE SSR BIT IN TSSR TO SET, THEN CHECKS THE CONTENTS
    275                             ;       OF BOTH TSBA AND TSSR.  THE MODULE IS FUNCTIONING CORRECTLY IF
    276                             ;       DATA WRITTEN APPEARS IN BOTH BYTES OF TSBA AND THE FINAL CONTENT
    277                             ;       OF TSSR IS CORRECT (SAME AS AFTER INITIALIZATION EXCEPT FOR BITS
    278                             ;       8 AND 9, WHICH SHOULD CONTAIN BITS 8 AND 9 OF THE DATA PATTERN
    279                             ;       WRITTEN.  AN ERROR IS REPORTED AND A DESCRIPTIVE ANALYSIS GIVEN
    280                             ;       IF A DISCREPANCY IN TSBA OR TSSR IS DETECTED.  THE ANALYSIS
```

TEST 3: WRAP DATA - LOW BYTE

```
                                        ;       LISTS LIKELY FAULTY CANDIDATES FROM THE LOGIC ELEMENTS LISTED
                                        ;       ABOVE.  THE TEST IS REPEATED FOR ALL COMBINATIONS OF TEST DATA
                                        ;       BYTES (0-377 OCTAL).
                                        ;
024400                                  BGNTST
024400                                                                          T3::
024400  012700  025043                  MOV     #TST3ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
024404  004737  016570                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
024410  012737  000024  002214          MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
024416  005004                  T3LOOP: CLR     R4                      ;STARTING DATA PATTERN
024420  012703  000001                  MOV     #1,R3                   ;FORCE SOFT INIT ON FIRST PASS
024424  005703                  5$:     TST     R3                      ;SET IF INIT IS REQUIRED
024426  001412                          BEQ     10$                     ;BRANCH IF NO INIT NEEDED
024430  005003                          CLR     R3                      ;ASSUME NO INIT NEEDED ON NEXT LOOP
024432  004737  016054                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
024436  103406                          BCS     10$                     ;BR IF SOFT INIT = OK
024440  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
024442                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
024442  104455                                                                  TRAP    C$ERDF
024444  000455                                                                  .WORD   301
024446  003652                                                                  .WORD   SFIERR
024450  012104                                                                  .WORD   SFIMSG
024452  005203                          INC     R3                      ;FORCE INIT ON NEXT PASS
024454  005037  002220          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
                                        
024460                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
024460  104404                                                                  TRAP    C$BSEG
                                        
024462  110465  000000                  MOVB    R4,TSDB(R5)             ;SET MAINT MODE + WRITE DATA
024466  004737  016330                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
024472  103411                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
024474  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
024476  010402                          MOV     R4,R2                   ;DATA THAT WAS WRITTEN
024500                                  ERRDF   ERRNO,T3SSR,EXPREC      ;DEVICE FATAL SSR FAILED TO SET
024500  104455                                                                  TRAP    C$ERDF
024502  000456                                                                  .WORD   302
024504  024772                                                                  .WORD   T3SSR
024506  015554                                                                  .WORD   EXPREC
024510  005203                          INC     R3                      ;FORCE INIT ON NEXT PASS
024512  005237  002220                  INC     FATFLG                  ;SET THE FATAL ERROR FLAG
024516                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
024516  104406                                                                  TRAP    C$CLP1
024520  005737  002220                  TST     FATFLG                  ;DID WE GET FATAL ERROR ?
024524  001402                          BEQ     20$                     ;BRANCH IF NOT
024526  004737  017262                  JSR     PC,CKDROP               ;DROP UNIT, IF DROP ALLOWED
024532  010402                  20$:    MOV     R4,R2                   ;DATA PATTERN WRITTEN
024534  042702  177774                  BIC     #^C<BIT0!BIT1>,R2       ;CLEAR ALL BUT LOW 2 BITS
024540  000302                          SWAB    R2                      ;BITS 8 AND 9 HAVE LOW DATA BITS
024542  052702  002200                  BIS     #SSR!NBA,R2             ;THESE BITS MUST BE SET ALSO
024546  016501  000002                  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
024552  032701  000100                  BIT     #OFL,R1                 ;IS DRIVE OFF-LINE
024556  001402                          BEQ     25$                     ;BRANCH IF NOT OFFLINE
024560  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED
024564  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
024566  001405                          BEQ     30$                     ;OKAY IF MATCH
024570                                  ERRHRD  ERRNO,T3TSSR,EXPREC     ;TSSR WASN'T CORRECT
024570  104456                                                                  TRAP    C$ERHRD
```

TEST 3: WRAP DATA - LOW BYTE

```
                024572  000457                                                          .WORD   303
                024574  024726                                                          .WORD   T3TSSR
                024576  015554                                                          .WORD   EXPREC
    339         024600  005203                  INC     R3              ;FORCE INIT ON NEXT PASS
    340         024602                  30$:    CKLOOP                  ;LOOP ON ERROR ?
                024602  104406                                                          TRAP    C$CLP1
    341         024604  016501  000000          MOV     TSBA(R5),R1     ;GET TSBA REGISTER CONTENTS
    342         024610  005002                  CLR     R2              ;
    343         024612  150402                  BISB    R4,R2           ;DATA PATTERN WRITTEN
    344         024614  000302                  SWAB    R2              ;MOVE INTO TOP BYTE
    345         024616  150402                  BISB    R4,R2           ;BOTH HALVES SHOULD BE SAME
    346         024620  020102                  CMP     R1,R2           ;COMPARE EXPECTED TO RECEIVED
    347         024622  001405                  BEQ     35$             ;BR IF ERROR NOT EQUAL
    351         024624                          ERRHRD  ERRNO,T3TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
                024624  104456                                                          TRAP    C$ERHRD
                024626  000460                                                          .WORD   304
                024630  024662                                                          .WORD   T3TSBA
                024632  015554                                                          .WORD   EXPREC
    352         024634  005203                  INC     R3              ;FORCE INIT ON NEXT PASS
    353
    354         024636                  35$:    ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                024636                                                                  10000$:
                024636  104405                                                          TRAP    C$ESEG
    355
    356         024640  105204                  INCB    R4              ;NEXT DATA PATTERN TO TEST
    357         024642  001270                  BNE     5$              ;ALL DONE WHEN BACK TO ZERO
    358         024644  004737  016536          JSR     PC,TSTLOOP      ;SHOULD WE DO ITERATIONS ?
    359         024650  103002                  BCC     40$             ;BRANCH IF NOT
    360         024652  000137  024416          JMP     T3LOOP          ;LOOP UNTIL COUNT EXPIRED
    361         024656                  40$:    EXIT    TST             ;ALL DONE THIS TEST
                024656  104432                                                          TRAP    C$EXIT
                024660  000210                                                          .WORD   L10042-.
    362
    363                                 ;*
    364                                 ;LOCAL TEXT MESSAGES FOR TEST
    365                                 ;-
    366
    367 024662      124     123     102 T3TSBA: .ASCIZ  'TSBA Incorrect After TSDB Low Write'
    368 024726      124     123     123 T3TSSR: .ASCIZ  'TSSR Incorrect After TSDB Low Write'
    369 024772      116     157     040 T3SSR:  .ASCIZ  'No Sub-System Ready After TSDB Low Write'
    370 025043      127     162     141 TST3ID: .ASCIZ  'Wrap Data - Low Byte'
    371                                         .EVEN
    372 025070                          ENDTST
                025070                                                  L10042:
                025070  104401                                                          TRAP    C$ETST
    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 CAPABLITY OF THE 2901 TO CORRECTLY PERFORM AN
    383                                 ;       ADD.
    384                                 ;
```

```
                        ;
                        ;
                        ;       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.
                        ;
                        ;
                        ;
421 025072                      BGNTST
    025072                                                          T4::
422
423 025072                      BGNSUB                  ;////////////// BEGIN SUBTEST ////////////
    025072                                                          T4.1:
    025072  104402                                                         TRAP    C$BSUB
424
429 025074  012700  026402      MOV     #TST4ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
430 025100  004737  016570      JSR     PC,TSTSETUP     ;DO INITIAL TEST SETUP
431 025104  012737  000005  002214  MOV #5,LOOPCNT      ;PERFORM 5 ITERATIONS
432 025112                  T4LOOP:
433 025112  004737  016054      JSR     PC,SOFINIT      ;DO INITIALIZE ON CONTROLLER
434 025116  103405              BCS     20$             ;BR IF INIT WAS OK
438 025120  010001              MOV     R0,R1           ;CONTENTS OF TSSR REGISTER
439 025122                      ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
    025122  104455                                                  TRAP    C$ERDF
    025124  000621                                                 .WORD   401
    025126  003652                                                 .WORD   SFIERR
    025130  012104                                                 .WORD   SFIMSG
440 025132  005004          20$:    CLR     R4              ;SET RAM ADDRESS AT ZERO
441 025134  004737  016416      JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
```

TEST 4: RAM TEST

```
442 025140  105065  000000          25$:  CLRB    TSDB(R5)                ;SET INTO MAINTENANCE MODE
443 025144
444 025144                                BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
    025144  104404                                                               TRAP    C$BSEG
445
446 025146  110402                        MOVB    R4,R2                   ;EXPECTED DATA FROM WRAP-AROUND
447 025150  004737  016416                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
448 025154  010465  000000                MOV     R4,TSDB(R5)             ;LOAD ADDRESS INTO TSDB
449 025160  004737  016416                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
450 025164  110265  000000                MOVB    R2,TSDB(R5)             ;LOADS DATA INTO RAM LOCATION
451 025170  004737  016416                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
452 025174  016501  000000                MOV     TSBA(R5),R1             ;READS WRAP DATA
453 025200  120102                        CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ
454 025202  001404                        BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
458 025204                                ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
    025204  104456                                                               TRAP    C$ERHRD
    025206  000622                                                               .WORD   402
    025210  026240                                                               .WORD   TSBAM2
    025212  015554                                                               .WORD   EXPREC
459 025214                          30$:
460 025214                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
    025214                                                                      10000$:
    025214  104405                                                               TRAP    C$ESEG
461
462 025216  005204                        INC     R4                      ;NEXT ADDRESS
463 025220  020427  010000                CMP     R4,#10000               ;END OF RAM MEMORY CHECK
464 025224  001347                        BNE     25$                     ;LOOP TILL ALL RAM WRITTEN
465 025226                                ESCAPE  SUB                     ;DON'T CONTINUE IF ERROR ON WRITE
    025226  104410                                                               TRAP    C$ESCAPE
    025230  000102                                                               .WORD   L10044-.
466
467 025232  005002                        CLR     R2                      ;CLEAR OUT R2 HIGH BITS
468 025234  005304                        DEC     R4                      ;SET BACK TO 7777
469 025236  110402                  40$:  MOVB    R4,R2                   ;GET DATA PATTERN BACK IN SHAPE
470 025240  004737  016416                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
471 025244  010465  000000                MOV     R4,TSDB(R5)             ;LOAD UP RAM ADDRESS POINTER
472 025250  004737  016416                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
473 025254  016501  000000                MOV     TSBA(R5),R1             ;READ RAM CONTENTS BACK
474 025260  120102                        CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
475 025262  001404                        BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
479 025264                                ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
    025264  104456                                                               TRAP    C$ERHRD
    025266  000623                                                               .WORD   403
    025270  026240                                                               .WORD   TSBAM2
    025272  015554                                                               .WORD   EXPREC
480 025274                          45$:  CKLOOP                          ;SCOPE LOOP
    025274  104406                                                               TRAP    C$CLP1
481 025276  116501  000001                MOVB    TSBAH(R5),R1            ;HIGH BYTE READ OF TSBA
482 025302  010402                        MOV     R4,R2                   ;DATA PATTERN WRITTEN
483 025304  000302                        SWAB    R2                      ;HIGH TO LOW
484 025306  060402                        ADD     R4,R2                   ;TOTAL OF BYTES IN LOW BYTE
485 025310  120102                        CMPB    R1,R2                   ;SUM OF BYTES WRITTEN TO TSDB = TSBAH
486 025312  001404                        BEQ     50$                     ;BR IF OK, THEY SHOULD BE
490 025314                                ERRHRD  ERRNO,M2901,EXPREC      ;2901 PROBLEM ADDER
    025314  104456                                                               TRAP    C$ERHRD
    025316  000624                                                               .WORD   404
    025320  026150                                                               .WORD   M2901
```

```
TEST  4: RAM TEST

     025322  015554                                                          .WORD   EXPREC
491  025324                  50$:    CKLOOP                          ;SCOPE LOOP
     025324  104406                                                          TRAP    C$CLP1
492  025326  005304                  DEC     R4                      ;DROP DATA COUNTER (PATTERN)
493  025330  002342                  BGE     40$                     ;NOT AT LOC. ZERO YET
494
495  025332                          ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     025332                                                                  L10044:
     025332  104403                                                          TRAP    C$ESUB
496
497
498  025334                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     025334                                                                  T4.2:
     025334  104402                                                          TRAP    C$BSUB
499                          ;          TEST 4, SUBTEST 2
500
501                          ;
502                          ;          THIS SUBTEST WRITES RAM WITH ALL ZEROS
503                          ;          THEN WALKS AN ALL ONES WORD DOWN THROUGH MEMORY
504                          ;
505  025336  004737  016054            JSR     PC,$OFINIT              ;DO INITIALIZE ON CONTROLLER
506  025342  103405                  BCS     20$                     ;BR IF INIT WAS OK
510  025344  010001                  MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
511  025346                          ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
     025346  104455                                                          TRAP    C$ERDF
     025350  000625                                                          .WORD   405
     025352  003652                                                          .WORD   SFIERR
     025354  012104                                                          .WORD   SFIMSG
512  025356  005002                  20$:    CLR     R2                      ;TEST DATA = 0
513  025360  005004                  CLR     R4                      ;STARTING RAM ADDRESS = 0
514  025362  004737  016416            JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
515  025366  105065  000000            CLRB    TSDB(R5)                ;SET INTO MAINTENANCE MODE
516  025372                  25$:
517  025372                          BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     025372  104404                                                          TRAP    C$BSEG
518
519  025374  004737  016416            JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
520  025400  010465  000000            MOV     R4,TSDB(R5)             ;LOAD ADDRESS INTO TSDB
521  025404  004737  016416            JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
522  025410  110265  000000            MOVB    R2,TSDB(R5)             ;LOADS DATA INTO RAM LOCATION
523  025414  004737  016416            JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
524  025420  016501  000000            MOV     TSBA(R5),R1             ;READS WRAP DATA
525  025424  120102                  CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ ?
526  025426  001404                  BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
530  025430                          ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
     025430  104456                                                          TRAP    C$ERHRD
     025432  000626                                                          .WORD   406
     025434  026240                                                          .WORD   TSBAM2
     025436  015554                                                          .WORD   EXPREC
531  025440                  30$:
532  025440                          ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<<
     025440                                                                  10000$:
     025440  104405                                                          TRAP    C$ESEG
533
534  025442  005204                  INC     R4                      ;NEXT ADDRESS
535  025444  020427  010000            CMP     R4,#10000               ;END OF RAM MEMORY CHECK
536  025450  001350                  BNE     25$                     ;BR, MORE RAM TO GO
```

TEST 4: RAM TEST

```
537
538 025452  005304                  35$:    DEC     R4                      ;SET BACK TO 7777
539 025454  005002                          CLR     R2                      ;SET TO ALL ZEROS
540 025456  004737  016416          40$:    JSR     PC,CHKTSSR              ;WAIT FOR READY (SSR) TO SET
541 025462  010465  000000                  MOV     R4,TSDB(R5)             ;LOAD UP THE ADDRESS FOR RAM
542 025466  004737  016416                  JSR     PC,CHKTSSR              ;WAIT FOR READY (SSR) TO SET
543 025472  016501  000000                  MOV     TSBA(R5),R1             ;READ THE RAM CONTENTS BACK
544 025476  005002                          CLR     R2                      ;LOOKING FOR 000000 (EXPECTED)
545 025500  120102                          CMPB    R1,R2                   ;BOTH SHOULD BE 00000000 BINARY
546 025502  001404                          BEQ     43$                     ;BR, IF DATA IS GOOD
550 025504                                  ERRHRD  ERRNO,TSBAM3,EXPREC     ;CHARACTERISTICS DATA NOT CORRECT
    025504  104456                                                              TRAP    C$ERHRD
    025506  000627                                                              .WORD   407
    025510  026322                                                              .WORD   TSBAM3
    025512  015554                                                              .WORD   EXPREC
551 025514  012702  000377          43$:    MOV     #000377,R2              ;SET ALL ONES WORD
552 025520  010465  000000                  MOV     R4,TSDB(R5)             ;LOAD UP RAM ADDRESS POINTER
553 025524  004737  016416                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
554 025530  110265  000000                  MOVB    R2,TSDB(R5)             ;WRITE DATA INTO RAM
555 025534  004737  016416                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
556 025540  016501  000000                  MOV     TSBA(R5),R1             ;READ RAM CONTENTS BACK
557 025544  120102                          CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
558 025546  001404                          BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
562 025550                                  ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
    025550  104456                                                              TRAP    C$ERHRD
    025552  000630                                                              .WORD   408
    025554  026240                                                              .WORD   TSBAM2
    025556  015554                                                              .WORD   EXPREC
563 025560                          45$:    CKLOOP                          ;SCOPE LOOP
    025560  104406                                                              TRAP    C$CLP1
564 025562  004737  016416                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
565 025566  010465  000000                  MOV     R4,TSDB(R5)             ;WORD WRITE TO SET UP ADDRESS
566 025572  004737  016416                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
567 025576  116501  000001                  MOVB    TSBAH(R5),R1            ;HI ! BYTE READ OF TSBA
568 025602  010403                          MOV     R4,R3                   ;DATA PATTERN WRITTEN
569 025604  000303                          SWAB    R3                      ;HIGH TO LOW
570 025606  060403                          ADD     R4,R3                   ;TOTAL OF BYTES IN LOW BYTE
571 025610  120103                          CMPB    R1,R3                   ;SUM OF BYTES WRITTEN TO TSDB = TSBAH
572 025612  001404                          BEQ     50$                     ;BR IF OK, THEY SHOULD BE
576 025614                                  ERRHRD  ERRNO,M2901,EXPREC      ;2901 PROBLEM ADDER
    025614  104456                                                              TRAP    C$ERHRD
    025616  000631                                                              .WORD   409
    025620  026150                                                              .WORD   M2901
    025622  015554                                                              .WORD   EXPREC
577 025624                          50$:    CKLOOP                          ;SCOPE LOOP
    025624  104406                                                              TRAP    C$CLP1
578 025626  005304                          DEC     R4                      ;DROP RAM ADDRESS POINTER
579 025630  002312                          BGE     40$                     ;NOT AT LOC. ZERO YET
580
581 025632                                  ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
    025632                                                                  L10045:
    025632  104403                                                              TRAP    C$ESUB
582
583
584 025634                                  BGNSUB                          ;/////////////// BEGIN SUBTEST ///////////////
    025634                                                                  T4.3:
    025634  104402                                                              TRAP    C$BSUB
```

TEST 4: RAM TEST

```
                                ;       TEST 4, SUBTEST 3
                                ;
                                ;       THIS SUBTEST WRITES RAM WITH ALL ONES
                                ;       THEN WALKS A ZERO WORD DOWN THROUGH MEMORY
                                ;
025636  004737  016054          20$:    JSR     PC,$OFINIT              ;DO INITIALIZE ON CONTROLLER
025642  103405                          BCS     20$                     ;BR IF INIT WAS OK
025644  010001                          MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
025646                                  ERRDF   ERRNO.SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
025646  104455                                                                  TRAP    C$ERDF
025650  000632                                                                  .WORD   410
025652  003652                                                                  .WORD   SFIERR
025654  012104                                                                  .WORD   SFIMSG
025656  012702  177777          25$:    MOV     #177777,R2              ;SET DATA AT ALL ONES
025662  005004                          CLR     R4                      ;SET RAM ADDRESS AT ZERO
025664  004737  016416                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025670  105065  000000                  CLRB    TSDB(R5)                ;SET INTO MAINTENANCE MODE
025674                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
025674  104404                                                                  TRAP    C$BSEG

025676  004737  016416                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025702  010465  000000                  MOV     R4,TSDB(R5)             ;LOAD ADDRESS INTO TSDB
025706  004737  016416                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025712  110265  000000                  MOVB    R2,TSDB(R5)             ;LOADS DATA INTO RAM LOCATION
025716  004737  016416                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025722  016501  000000                  MOV     TSBA(R5),R1             ;READS WRAP DATA
025726  120102                          CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ ?
025730  001404                          BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
025732                                  ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
025732  104456                                                                  TRAP    C$ERHRD
025734  000633                                                                  .WORD   411
025736  026240                                                                  .WORD   TSBAM2
025740  015554                                                                  .WORD   EXPREC
025742                          30$:
025742                                  ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
025742                                                                  10000$:
025742  104405                                                                  TRAP    C$ESEG

025744  005204                          INC     R4                      ;NEXT ADDRESS
025746  020427  010000                  CMP     R4,#10000               ;END OF RAM MEMORY CHECK
025752  001350                          BNE     25$                     ;BR, MORE RAM TO GO

025754                                  ESCAPE  SUB                     ;NO CHECK IF WRITTEN INCORRECTLY
025754  104410                                                                  TRAP    C$ESCAPE
025756  000152                                                                  .WORD   L10046-.
025760  005304                  35$:    DEC     R4                      ;SET BACK TO 7777
025762  004737  016416          40$:    JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025766  012702  000377                  MOV     #000377,R2              ;SET UP EXPECTED DATA REGISTER
025772  005001                          CLR     R1                      ;CLEAN OUT REGISTER
025774  010465  000000                  MOV     R4,TSDB(R5)             ;SELECT ADDRESS IN RAM
026000  004737  016416                  JSR     PC,CHKTSSR              ;WAIT FOR READY (SSR)
026004  016501  000000                  MOV     TSBA(R5),R1             ;PICK UP RAM CONTENTS
026010  120102                          CMPB    R1,R2                   ;IS MEMORY STILL ALL ONES
026012  001404                          BEQ     43$                     ;BR, IF OK (ALL ONES)
026014                                  ERRHRD  ERRNO,TSBAM3,EXPREC     ;MEMORY CHANGED AFTER ALL ONES WRITE
```

TEST 4: RAM TEST

```
                026014  104456                                                          TRAP    C$ERHRD
                026016  000634                                                          .WORD   412
                026020  026322                                                          .WORD   TSBAM3
                026022  015554                                                          .WORD   EXPREC
    638         026024  005002          43$:    CLR     R2              ;SET UP NEW EXPECTED
    639         026026  010465  000000          MOV     R4,TSDB(R5)     ;LOAD UP RAM ADDRESS POINTER
    640         026032  004737  016416          JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
    641         026036  110265  000000          MOVB    R2,TSDB(R5)     ;WRITE DATA INTO RAM
    642         026042  004737  016416          JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
    643         026046  016501  000000          MOV     TSBA(R5),R1     ;READ RAM CONTENTS BACK
    644         026052  120102                  CMPB    R1,R2           ;CHECK WITH DATA WRITTEN
    645         026054  001404                  BEQ     45$             ;BR IF OK, DATA IN = DATA OUT
    649         026056                          ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
                026056  104456                                                          TRAP    C$ERHRD
                026060  000635                                                          .WORD   413
                026062  026240                                                          .WORD   TSBAM2
                026064  015554                                                          .WORD   EXPREC
    650         026066          45$:    CKLOOP                          ;SCOPE LOOP
                026066  104406                                                          TRAP    C$CLP1
    651         026070  004737  016416          JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
    652         026074  116501  000001          MOVB    TSBAH(R5),R1    ;HIGH BYTE READ OF TSBA
    653         026100  010203                  MOV     R2,R3           ;DATA PATTERN WRITTEN
    654         026102  000303                  SWAB    R3              ;HIGH TO LOW
    655         026104  060203                  ADD     R2,R3           ;TOTAL OF BYTES IN LOW BYTE
    656         026106  120103                  CMPB    R1,R3           ;SUM OF BYTES WRITTEN TO TSDB = TSBAH
    657         026110  001404                  BEQ     50$             ;BR IF OK, THEY SHOULD BE
    661         026112                          ERRHRD  ERRNO,M2901,EXPREC      ;2901 PROBLEM ADDER
                026112  104456                                                          TRAP    C$ERHRD
                026114  000636                                                          .WORD   414
                026116  026150                                                          .WORD   M2901
                026120  015554                                                          .WORD   EXPREC
    662         026122          50$:    CKLOOP                          ;SCOPE LOOP
                026122  104406                                                          TRAP    C$CLP1
    663         026124  005304                  DEC     R4              ;DROP RAM ADDRESS POINTER
    664         026126  001315                  BNE     40$             ;NOT AT LOC. ZERO YET
    665
    666         026130                          ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                026130                                                                  L10046:
                026130  104403                                                          TRAP    C$ESUB
    667
    668         026132  004737  016536          JSR     PC,TSTLOOP      ;DO WE NEED TO ITERATE TEST ?
    669         026136  103002                  BCC     63$             ;BRANCH IF NOT
    670         026140  000137  025112          JMP     T4LOOP          ;EXECUTE AGAIN
    671         026144          63$:    EXIT    TST                     ;ALL DONE THIS TEST
                026144  104432                                                          TRAP    C$EXIT
                026146  000256                                                          .WORD   L10043-.
    672                         ;+
    673                         ;LOCAL TEXT MESSAGES FOR TEST
    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                         .EVEN
    681 026424                  ENDTST
                026424                                                                  L10043:
```

TEST 4: RAM TEST

```
                026424  104401                                                  TRAP    C$ETST
682
683                                         .SBTTL  TEST  5: SECOND INITIALIZATION TEST
684
685                                         ;THIS TEST VERIFIES THE SAME ELEMENTS AS DID INITIALIZATION TEST
686                                         ;#1 AND ALSO CHECKS THAT CERTAIN PARTS OF RAM IS CLEARED TO ZERO
687                                         ;AND THAT 2901 REGISTERS 10 AND 11 ARE ALSO CLEARED TO ZERO.
688                                         ;THIS IS A CONFIDENCE CHECK OF A PART OF THE SELF-TEST SEQUENCE
689                                         ;(I.E., THAT IT IS REALLY BEING EXECUTED).  FOR EACH OF TWO
690                                         ;SUBTESTS (ONE FOR INITIALIZING VIA A BUS INIT, THE OTHER FOR
691                                         ;INITIALIZING BY WRITING INTO THE TSSR), THE FOLLOWING SEQUENCE
692                                         ;IS PERFORMED:
693                                         ;
694                                         ;
695                                         ;     1.  EACH RAM LOCATION AND 2901 REGISTERS 10 AND 11 ARE SET
696                                         ;         TO ALL 1'S BY USING WRITES INTO THE TSDB REGISTER (LOW
697                                         ;         BYTE AND MAINTENANCE MODE WORD WRITES).
698                                         ;
699                                         ;     2.  THE CONTROLLER IS INITIALIZED AND THE VARIOUS CHECKS ON
700                                         ;         THE TSSR DESCRIBED IN INITIALIZATION TEST #1 ARE
701                                         ;         PERFORMED.
702                                         ;
703                                         ;     3.  #1'S (377 OCTAL) ARE WRITTEN INTO THE LOW BYTE OF TSDB,
704                                         ;         WHICH SHOULD CAUSE RAM LOCATION 0 TO BE WRITTEN TO ALL
705                                         ;         1'S SINCE 2901 REGISTERS 10 AND 11, SPECIFYING THE RAM
706                                         ;         ADDRESS, SHOULD BE 0.  RAM LOCATION 0 IS VERIFIED BY
707                                         ;         WRITING A WORD OF ZEROS INTO THE TSDB.  THE RESULTING
708                                         ;         LOW BYTE OF TSBA SHOULD CONTAIN ALL 1'S.
709                                         ;
710                                         ;     4.  THE ENTIRE RAM IS SCANNED.  LOCATION 0 SHOULD CONTAIN
711                                         ;         ALL 1'S AND THE REMAINING LOCATIONS, EXCEPT FOR THE
712                                         ;         MESSAGE BUFFER IMAGE AREA, SHOULD CONTAIN 0.
713                                         ;         DISCREPANCIES ARE REPORTED.  AN ERROR AT THIS POINT IS
714                                         ;         MOST LIKELY DUE TO A ROM, PIPELINE OR SEQUENCER PROBLEM
715                                         ;         OR A TIMING PROBLEM.
716                                         ;
717                                         ;
718                                         ;
719                                         ;
720 026426                                  BGNTST
    026426                                                                      T5::
725 026426  012700  027400                  MOV     #TSTSID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
726 026432  004737  016570                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
727 026436  012737  000024  002214          MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
728 026444                          T5LOOP:
729 026444  005037  002220                  CLR     FATFLG                  ;CLEAR THE FATAL ERROR FLAG
730
731 026450                                  BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
    026450                                                                      T5.1:
    026450  104402                                                                      TRAP    C$BSUB
732
733 026452  004737  016054                  JSR     PC,SOFINIT              ;DO A SOFT TO START
734 026456  103404                          BCS     10$                     ;BRANCH IF O.K.
738 026460                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;REPORT ERROR AND DROP DRIVE
    026460  104455                                                                      TRAP    C$ERDF
    026462  000765                                                                     .WORD   501
    026464  003652                                                                     .WORD   SFIERR
```

TEST 5: SECOND INITIALIZATION TEST

```
                026466  012104                                                          .WORD   SFIMSG
    739         026470  012702  177777          10$:    MOV     #-1,R2                  ;ALL ONE DATA PATTERN
    740         026474  005004                  CLR     R4                              ;STARTING RAM ADDRESS
    741         026476  004737  016416          JSR     PC,CHKTSSR                      ;WAIT FOR READY, NON-AMBIGUOUS
    742         026502  105065  000000          15$:    CLRB    TSDB(R5)                ;SET MAINTENANCE MODE
    743         026506  004737  016416          JSR     PC,CHKTSSR                      ;WAIT FOR READY, NON-AMBIGUOUS
    744         026512  010465  000000          MOV     R4,TSDB(R5)                     ;SET THE NEXT RAM ADDRESS
    745         026516  004737  016416          JSR     PC,CHKTSSR                      ;WAIT FOR READY, NON-AMBIGUOUS
    746         026522  110265  000000          MOVB    R2,TSDB(R5)                     ;LOAD TEST DATA
    747         026526  005204                  INC     R4                              ;NEXT ADDRESS TO TEST
    748         026530  020427  007777          CMP     R4,#7777                        ;COMPARE TO LAST ADDRESS
    749         026534  003762                  BLE     15$                             ;BRANCH TILL ALL DATA WRITTEN
    750         026536                          BRESET                                  ;ISSUE A BUS RESET
                026536  104433                                                          TRAP    C$RESET
    751         026540  004737  016416          JSR     PC,CHKTSSR                      ;WAIT FOR READY, NON-AMBIGUOUS
    752         026544  016501  000002          MOV     TSSR(R5),R1                     ;GET THE CONTENTS OF TSSR
    753         026550  010102                  MOV     R1,R2                           ;CONTENTS OF TSSR
    754         026552  042702  176277          BIC     #^C<HIADDR!OFL>,R2              ;THESE BITS MAY BE SET
    755         026556  052702  002200          BIS     #SSR!NBA,R2                     ;READY AND NEW DATA SHOULD BE SET
    756         026562  020102                  CMP     R1,R2                           ;COMPARE EXPECTED TO RECEIVED
    757         026564  001406                  BEQ     20$                             ;BRANCH IF COMPARE
    761         026566                          ERRDF   ERRNO,SFHERR,SFFMSG             ;REPORT A FATAL ERROR
                026566  104455                                                          TRAP    C$ERDF
                026570  000766                                                          .WORD   502
                026572  003705                                                          .WORD   SFHERR
                026574  012152                                                          .WORD   SFFMSG
    762         026576  005237  002220          INC     FATFLG                          ;SET FATAL ERROR FLAG
    763         026602                  20$:    CKLOOP                                  ;LOOP ON ERROR IF FLAG SET
                026602  104406                                                          TRAP    C$CLP1
    764         026604                          ESCAPE  SUB                             ;EXIT IF FATAL ERROR DETECTED
                026604  104410                                                          TRAP    C$ESCAPE
                026606  000170                                                          .WORD   L10050-.
    765         026610  004737  016416          JSR     PC,CHKTSSR                      ;WAIT FOR SSR TO SET
    766         026614  105065  000000          CLRB    TSDB(R5)                        ;PUT BACK INTO MAINTENANCE MODE
    767         026620  004737  016416          JSR     PC,CHKTSSR                      ;WAIT FOR READY, NON-AMBIGUOUS
    768         026624  005065  000000          CLR     TSDB(R5)                        ;SET ADDRESS BACK TO 0000
    769         026630  012702  000377          MOV     #377,R2                         ;
    770         026634  004737  016416          JSR     PC,CHKTSSR                      ;WAIT FOR READY, NON-AMBIGUOUS
    771         026640  110265  000000          MOVB    R2,TSDB(R5)                     ;SHOULD POINT TO RAM 0
    772         026644  004737  016416          JSR     PC,CHKTSSR                      ;WAIT FOR READY, NON-AMBIGUOUS
    773         026650  005065  000000          CLR     TSDB(R5)                        ;SELECT LOCATION 0
    774         026654  004737  016416          JSR     PC,CHKTSSR                      ;WAIT FOR READY, NON-AMBIGUOUS
    775         026660  116501  000000          MOVB    TSBA(R5),R1                     ;READ RAM LOCATION SPECIFIED
    776         026664  120102                  CMPB    R1,R2                           ;LOCATION SHOULD BE 377 OCTAL
    777         026666  001406                  BEQ     25$                             ;BR IF OK
    778         026670                          ERRDF   ERRNO,TSADDR,EXPREC             ;WASN'T POINTING TO CORRECT LOC.
                026670  104455                                                          TRAP    C$ERDF
                026672  000766                                                          .WORD   502
                026674  027466                                                          .WORD   TSADDR
                026676  015554                                                          .WORD   EXPREC
    779         026700  005237  002220          INC     FATFLG                          ;SET THE FATAL ERROR FLAG
    780         026704                  25$:    CKLOOP                                  ;SCOPE LOOP
                026704  104406                                                          TRAP    C$CLP1
    781         026706                          ESCAPE  SUB                             ;NO MORE CHECKS IF FATAL ERROR
                026706  104410                                                          TRAP    C$ESCAPE
                026710  000066                                                          .WORD   L10050-.
    782         026712  012704  000310          MOV     #310,R4                         ;START WITH LOC 310
```

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          30$:  MOV     R4,TSDB(R5)             ;SELECT LOCATION SPECIFIED
    786 026730  004737  016416                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    787 026734  116501  000000                MOVB    TSBA(R5),R1             ;READ LOC CONTENTS
    788 026740  120102                        CMPB    R1,R2                   ;CHECK MEMORY FOR 000000
    789 026742  001406                        BEQ     40$                     ;BRANCH IF DATA OKAY
    790 026744                                ERRDF   ERRNO,TSMEM,SFFMSG      ;MEMORY NOT ZERO AFTER INIT.
        026744  104455                                                            TRAP    C$ERDF
        026746  000766                                                            .WORD   502
        026750  027430                                                            .WORD   TSMEM
        026752  012152                                                            .WORD   SFFMSG
    791 026754  005237  002220                INC     FATFLG                  ;SET THE FATAL ERROR FLAG
    792 026760                          40$:  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$ESUB
    799
    800 027000  005737  002220                TST     FATFLG                  ;IS FATAL ERROR FLAG SET ?
    801 027004  001404                        BEQ     50$                     ;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                          50$:
    805
    806 027016                                BGNSUB                          ;////// ///// BEGIN SUBTEST ////////////
        027016                                                                T5.2:
        027016  104402                                                            TRAP    C$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$ERDF
        027030  000767                                                            .WORD   503
        027032  003652                                                            .WORD   SFIERR
        027034  012104                                                            .WORD   SFIMSG
    814 027036  012702  177777          10$:  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          15$:  CLRB    TSDB(R5)                ;SET MAINTENANCE MODE
    818 027054  004737  016416                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    819 027060  010465  000000                MOV     R4,TSDB(R5)             ;SET THE NEXT RAM ADDRESS
    820 027064  004737  016416                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    821 027070  110265  000000                MOVB    R2,TSDB(R5)             ;LOAD TEST DATA
    822 027074  005204                        INC     R4                      ;NEXT ADDRESS TO TEST
    823 027076  020427  007777                CMP     R4,#7777                ;COMPARE TO LAST ADDRESS
    824 027102  003762                        BLE     15$                     ;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
```

F10

TEST 5: SECOND INITIALIZATION TEST

```
828 027120  010102                      MOV     R1,R2                   ;CONTENTS OF TSSR
829 027122  042702  176277              BIC     #^C<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     20$                     ;BRANCH IF COMPARE
836 027136                              ERRDF   ERRNO,SFHERR,SFFMSG     ;REPORT A FATAL ERROR
    027136  104455                                                              TRAP    C$ERDF
    027140  000770                                                              .WORD   504
    027142  003705                                                              .WORD   SFHERR
    027144  012152                                                              .WORD   SFFMSG
837 027146  005237  002220              INC     FATFLG                  ;SET FATAL ERROR FLAG
838 027152                      20$:    CKLOOP                          ;LOOP ON ERROR IF FLAG SET
    027152  104406                                                              TRAP    C$CLP1
839 027154                              ESCAPE  SUB                     ;EXIT IF FATAL ERROR DETECTED
    027154  104410                                                              TRAP    C$ESCAPE
    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     25$                     ;BR IF OK
853 027240                              ERRDF   ERRNO,T5ADDR,EXPREC     ;WASN'T POINTING TO CORRECT LOC.
    027240  104455                                                              TRAP    C$ERDF
    027242  000770                                                              .WORD   504
    027244  027466                                                              .WORD   T5ADDR
    027246  015554                                                              .WORD   EXPREC
854 027250  005237  002220              INC     FATFLG                  ;SET THE FATAL ERROR FLAG
855 027254                      25$:    CKLOOP                          ;SCOPE LOOP
    027254  104406                                                              TRAP    C$CLP1
856 027256                              ESCAPE  SUB                     ;NO MORE CHECKS IF FATAL ERROR
    027256  104410                                                              TRAP    C$ESCAPE
    027260  000066                                                              .WORD   L10051-.
857 027262  012704  000310              MOV     #310,R4                 ;START WITH _OC 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      30$:    MOV     R4,TSDB(R5)             ;SELECT LOCATION SPECIFIED
861 027300  004737  016416              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
862 027304  116501  000000              MOVB    TSBA(R5),R1             ;READ LOC CONTENTS
863 027310  120102                      CMPB    R1,R2                   ;CHECK MEMORY FOR 000000
864 027312  001406                      BEQ     40$                     ;BRANCH IF DATA OKAY
865 027314                              ERRDF   ERRNO,TSMEM,SFFMSG      ;MEMORY NOT ZERO AFTER INIT.
    027314  104455                                                              TRAP    C$ERDF
    027316  000770                                                              .WORD   504
    027320  027430                                                              .WORD   TSMEM
    027322  012152                                                              .WORD   SFFMSG
866 027324  005237  002220              INC     FATFLG                  ;SET THE FATAL ERROR FLAG
867 027330                      40$:    CKLOOP
    027330  104406                                                              TRAP    C$CLP1
868 027332                              ESCAPE  SUB                     ;EXIT ON FATAL ERROR
```

```
TEST  5: SECOND INITIALIZATION TEST

        027332  104410                                                          TRAP    C$ESCAPE
        027334  000012                                                          .WORD   L10051-.
    869 027336  005204                  INC     R4                  ;LOOK AT NEXT RAM LOC.
    870 027340  020427  000400          CMP     R4,#400             ;AT TOP OF RAM ADDRESS SPACE
    871 027344  001353                  BNE     30$                 ;BRANCH TILL ALL MEMORY TESTED
    872
    873 027346                          ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        027346                                                                  L10051:
        027346  104403                                                          TRAP    C$ESUB
    874
    875 027350  005737  002220          TST     FATFLG              ;IS FATAL ERROR FLAG SET ?
    876 027354  001402                  BEQ     50$                 ;BRANCH IF NOT
    877 027356  004737  017262          JSR     PC,CKDROP           ;NO LOOP, TRY TO DROP DEVICE
    878 027362  004737  016536  50$:    JSR     PC,TSTLOOP          ;SHOULD WE DO ITERATIONS ?
    879 027366  103002                  BCC     60$                 ;BRANCH IF NOT
    880 027370  000137  026444          JMP     T5LOOP              ;LOOP UNTIL COUNT EXPIRED
    881 027374                  60$:    EXIT    TST                 ;ALL DONE THIS TEST
        027374  104432                                                          TRAP    C$EXIT
        027376  000132                                                          .WORD   L10047-.
    882
    883                                 ;+
    884                                 ;LOCAL TEXT MESSAGES FOR TEST
    885                                 ;-
    886
    887 027400      105     170     164 T5TSID: .ASCIZ  'Extended Initialization'
    888 027430      111     156     143 T5MEM:  .ASCIZ  'Incorrect RAM Data After Init'
    889 027466      111     156     143 T5ADDR: .ASCIZ  'Incorrect RAM Address After Init'
    890                                 .EVEN
    891 027530                          ENDTST
        027530                                                                  L10047:
        027530  104401                                                          TRAP    C$ETST
    892
    893
    894                         .SBTTL  TEST  6: COMMAND REJECT
    895
    896
    897                         ;       THIS TEST VERIFIES THAT ALL COMMANDS OTHER THAN WRITE
    898                         ;       CHARACTERISTICS ARE REJECTED DUE TO THE NEED BUFFER ADDRESS
    899                         ;       (NBA) BIT BEING SET IN TSSR, AND THAT THE TSBA AND TSSR
    900                         ;       REGISTERS ARE LEFT IN THE PROPER STATE AFTER EACH COMMAND IS
    901                         ;       REJECTED.  THIS TEST CHECKS MICROPROCESSOR SEQUENCING, BASIC
    902                         ;       COMMAND DECODING AND DATI DMA HANDLING.  THIS TEST CONTAINS TWO
    903                         ;       SUBTESTS:  SUBTEST 1 SEQUENCES THROUGH ALL COMMAND WORDS (OTHER
    904                         ;       THAN WRITE CHARACTERISTICS) WITH THE INTERRUPT ENABLE (IE) BIT
    905                         ;       CLEAR AND VERIFIES THAT AN INTERRUPT IS NOT GENERATED BY THE
    906                         ;       REJECTED COMMAND;  SUBTEST 2 PERFORMS SIMILARLY TO SUBTEST 1 BUT
    907                         ;       SETS THE IE BIT IN EACH COMMAND WORD AND VERIFIES THAT AN
    908                         ;       INTERRUPT IS GENERATED WHEN THE COMMAND IS REJECTED.  SUBTEST 1
    909                         ;       SETS UP THE INTERRUPT SERVICE ROUTINE TO FLAG UNEXPECTED
    910                         ;       INTERRUPTS.  THE COMMAND WORD IN THE COMMAND BUFFER IS
    911                         ;       INITIALIZED TO 100000 (OCTAL) AND THE REMAINING THREE WORDS IN
    912                         ;       THE COMMAND BUFFER ARE SET TO KNOWN UNIQUE PATTERNS.  THEN THE
    913                         ;       FOLLOWING SEQUENCE IS PERFORMED:
    914                         ;
    915                         ;
    916                         ;            1.  INITIALIZE THE CONTROLLER BY WRITING INTO THE TSSR;
    917                         ;                PROPER INITIAL CONDITIONS ARE VERIFIED.
```

TEST 6: COMMAND REJECT

```
                        ;
                        ;           2.  TSDB IS WRITTEN WITH ADDRESS OF THE COMMAND BUFFER TO
                        ;               START PROCESSING.
                        ;
                        ;           3.  THE PROGRAM WAITS FOR SSR TO SET;  IF SSR DOES NOT SET,
                        ;               AN ERROR REPORT IS ISSUED AND THE TEST IS ABORTED.
                        ;
                        ;           4.  THE CONTENTS OF TSSR ARE CHECKED.  TSSR IS CORRECT IF
                        ;               IT CONTAINS EITHER OCTAL 102206 OR 102306 (BIT 6
                        ;               DEPENDS UPON THE STATE OF THE TAPE TRANSPORT).
                        ;
                        ;           5.  THE CONTENTS OF TSBA ARE CHECKED.  TSBA SHOULD CONTAIN
                        ;               THE INITIAL COMMAND BUFFER ADDRESS (LOADED IN STEP 2)
                        ;               PLUS 10 (OCTAL);  I.E., TSBA SHOULD POINT TO THE WORD
                        ;               JUST AFTER THE COMMAND PACKET (NOTE THAT 4 COMMAND
                        ;               PACKET WORDS ARE ALWAYS FETCHED).
                        ;
                        ;           6.  USING THE MAINTENANCE MODE WRAPAROUND FUNCTIONS, THE
                        ;               COMMAND IMAGE BLOCK IN THE M7196'S RAM (LOCATIONS
                        ;               201-210 (OCTAL)) ARE CHECKED;  THE IMAGE SHOULD CONTAIN
                        ;               A COPY OF THE FOUR COMMAND PACKET WORDS AS SET UP IN
                        ;               CPU MEMORY.
                        ;
                        ;           7.  THE COMMAND WORD IN THE COMMAND BUFFER IS INCREMENTED
                        ;               TO THE NEXT PATTERN NOT CONTAINING WRITE
                        ;               CHARACTERISTICS OR IE.  THE REMAINING THREE WORD OF THE
                        ;               COMMAND BUFFER ARE SEQUENCED WITH PSEUDO-RANDOM DATA.
                        ;               IF THE COMMAND WORD HAS NOT REACHED ITS MAXIMUM VALUE
                        ;               (177777+1), THE TEST SEQUENCE IS REPEATED.
                        ;
                        ;
                        ;       SUBTEST 2 IS IDENTICAL TO SUBTEST 1, EXCEPT THAT THE PROGRAM
                        ;       CAUSES THE IE BIT TO BE SET IN EACH COMMAND WORD AND THEN
                        ;       VERIFIES THAT AN INTERRUPT OCCURS.
                        ;
027532                          BGNTST
027532                                                                  T6::
027532  012700  030763                MOV     #TST6ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
027536  004737  016570                JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
027542  012737  000024  002214        MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
027550                          T6LOOP:
027550                          BGNSUB                                  ;////////////// BEGIN SUBTEST ////////////
027550                                                                          T6.1:
027550  104402                                                                  TRAP    C$BSUB

027552                          SETPRI  #PRI00                          ;LOWER PRIORITY TO ALLOW INTERRUPTS
027552  012700  000000                                                          MOV     #PRI00,R0
027556  104441                                                                  TRAP    C$SPRI
027560  012704  030440                MOV     #T6PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
027564  012703  002752                MOV     #TSTBLK,R3              ;BLOCK OF TEST DATA
027570  012314          5$:           MOV     (R3)+,(R4)              ;INSERT THE NEXT TEST DATA WORD
027572                                BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
027572  104404                                                                  TRAP    C$BSEG
027574  004737  016054                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
027600  103405                        BCS     10$                     ;BR IF SOFT INIT = OK
027602  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

TEST 6: COMMAND REJECT

```
    976 027604                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
        027604  104455                                                              TRAP    C$ERDF
        027606  001131                                                              .WORD   601
        027610  003652                                                              .WORD   SFIERR
        027612  012104                                                              .WORD   SFIMSG
    977 027614  005037  002220          10$:    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     15$                     ;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$ERDF
        027652  001132                                                              .WORD   602
        027654  030475                                                              .WORD   T6SSR
        027656  012116                                                              .WORD   PKTSSR
    989 027660  005237  002220                  INC     FATFLG                  ;SET FATAL ERROR FLAG
    990 027664                          15$:    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     22$                     ;BRANCH IF NOT
    997 027700                                  ERRHRD  ERRNO,T6INT,PKTSSR
        027700  104456                                                              TRAP    C$ERHRD
        027702  001133                                                              .WORD   603
        027704  030553                                                              .WORD   T6INT
        027706  012116                                                              .WORD   PKTSSR
    998 027710  012702  102206          22$:    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     25$                     ;BRANCH IF NOT OFF-LINE
   1003 027732  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
   1004 027736  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
   1005 027740  001404                          BEQ     30$                     ;OKAY IF MATCH
   1009 027742                                  ERRHRD  ERRNO,T6NBA,PKTSSR      ;NBA NOT SET TO REJECT
        027742  104456                                                              TRAP    C$ERHRD
        027744  001134                                                              .WORD   604
        027746  030450                                                              .WORD   T6NBA
        027750  012116                                                              .WORD   PKTSSR
   1010 027752                          30$:    CKLOOP                          ;LOOP ON ERROR ?
        027752  104406                                                              TRAP    C$CLP1
   1011 027754  004737  016416                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
   1012 027760  016501  000000                  MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
   1013 027764  010402                          MOV     R4,R2                   ;START OF THE PACKET
   1014 027766  062702  000010                  ADD     #10,R2                  ;EXPECT TSDA TO PACKET + 10
   1015 027772  020102                          CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
   1016 027774  001404                          BEQ     35$                     ;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
```

TEST 6: COMMAND REJECT

```
1021
1022
1023 030006  004737  011154          35$:    JSR     PC,CKRAM                ;SEE IF DATA IN RAM IS CORRECT
1024 030012  103404                          BCS     40$                     ;BRANCH IF PACKET IN RAM IS CORRECT
1028 030014                                  ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
     030014  104456                                                                  TRAP    C$ERHRD
     030016  001136                                                                  .WORD   606
     030020  004745                                                                  .WORD   PKTRAM
     030022  015570                                                                  .WORD   RAMERR
1029 030024                          40$:    ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     030024                                                                       10000$:
     030024  104405                                                                  TRAP    C$ESEG
1030 030026  011300                          MOV     (R3),R0                 ;PACKET COMMAND WORD
1031 030030  042700  177740                BIC     #177740,R0              ;GET BITS 0-4
1032 030034  020027  000004                CMP     R0,#4                   ;DON'T TEST WRITE CHARACTERISTICS
1033 030040  001002                          BNE     45$                     ;BRANCH IF OK
1034 030042  062703  000002                ADD     #2,R3                   ;GET NEXT WORD FROM DATA TABLE
1035 030046  020327  003062          45$:    CMP     R3,#TBLEND              ;REACHED END OF TABLE ?
1036 030052  103002                          BHIS    50$                     ;BRANCH IF END OF TABLE
1037 030054  000137  027570                JMP     5$                      ;CONTINUE TEST WITH NEW DATA
1038
1039 030060                          50$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     030060                                                                       L10053:
     030060  104403                                                                  TRAP    C$ESUB
1040
1041 030062  005737  002220                TST     FATFLG                  ;ANY FATAL ERRORS ?
1042 030066  001402                          BEQ     60$                     ;BRANCH IF NOT
1043 030070  004737  017262                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
1044
1045 030074                          60$:    BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     030074                                                                       T6.2:
     030074  104402                                                                  TRAP    C$BSUB
1046
1047 030076                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     030076  012700  000000                                                          MOV     #PRI00,R0
     030102  104441                                                                  TRAP    C$SPRI
1048 030104  012704  030440                MOV     #T6PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1049 030110  012703  002752                MOV     #TSTBLK,R3              ;START OF TEST DATA
1050 030114  012314          5$:     MOV     (R3)+,(R4)              ;PLACE NEXT DATA WORD IN PACKET
1051 030116                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     030116  104404                                                                  TRAP    C$BSEG
1052 030120  004737  016054                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1053 030124  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
1057 030126  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1058 030130                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     030130  104455                                                                  TRAP    C$ERDF
     030132  001137                                                                  .WORD   607
     030134  003652                                                                  .WORD   SFIERR
     030136  012104                                                                  .WORD   SFIMSG
1059 030140  005037  002220          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1060 030144  005037  002222                CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1061 030150  004737  016416                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1062 030154  052714  000200                BIS     #BIT7,(R4)              ;ENABLE INTERRUPTS
1063 030160  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
1064 030164  004737  016330                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
1065 030170  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1066 030172  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

K10

TEST 6: COMMAND REJECT

```
1070 030174                                ERRDF   ERRNO,T6SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     030174  104455                                                            TRAP    C$ERDF
     030176  001140                                                            .WORD   608
     030200  030475                                                            .WORD   T6SSR
     030202  012116                                                            .WORD   PKTSSR
1071 030204  005237  002220                INC     FATFLG                  ;SET FATAL ERROR FLAG
1072 030210                          15$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     030210  104406                                                            TRAP    C$CLP1
1073 030212                                ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
     030212  104410                                                            TRAP    C$ESCAPE
     030214  000170                                                            .WORD   L10054-.
1074 030216  005737  002222                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1075 030222  001004                        BNE     22$                     ;BRANCH IF YES
1079 030224                                ERRHRD  ERRNO,T6NINT,PKTSSR     ;REPORT ERROR IF NO INTERRUPT
     030224  104456                                                            TRAP    C$ERHRD
     030226  001141                                                            .WORD   609
     030230  030631                                                            .WORD   T6NINT
     030232  012116                                                            .WORD   PKTSSR
1080 030234  012702  102206          22$:   MOV     #SC!NBA!SSR!TSREJ,R2    ;EXPECTED CONTENTS OF TSSR
1081 030240  004737  016416                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1082 030244  016501  000002                MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1083 030250  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1084 030254  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1085 030256  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1086 030262  020201          25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1087 030264  001404                        BEQ     30$                     ;OKAY IF MATCH
1091 030266                                ERRHRD  ERRNO,T6NBA,PKTSSR      ;NBA NOT SET TO REJECT
     030266  104456                                                            TRAP    C$ERHRD
     030270  001142                                                            .WORD   610
     030272  030450                                                            .WORD   T6NBA
     030274  012116                                                            .WORD   PKTSSR
1092 030276                          30$:   CKLOOP                          ;LOOP ON ERROR ?
     030276  104406                                                            TRAP    C$CLP1
1093 030300  004737  016416                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1094 030304  016501  000000                MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
1095 030310  010402                        MOV     R4,R2                   ;START OF THE PACKET
1096 030312  062702  000010                ADD     #10,R2                  ;EXPECT TSDA TO PACKET + 10
1097 030316  020102                        CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1098 030320  001404                        BEQ     35$                     ;ERROR IF NOT EQUAL
1102 030322                                ERRHRD  ERRNO,T6TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     030322  104456                                                            TRAP    C$ERHRD
     030324  001143                                                            .WORD   611
     030326  030711                                                            .WORD   T6TSBA
     030330  015554                                                            .WORD   EXPREC
1103
1104
1105 030332  004737  011154          35$:   JSR     PC,CKRAM                ;SEE IF DATA IN RAM IS CORRECT
1106 030336  103404                        BCS     40$                     ;BRANCH IF PACKET IN RAM IS CORRECT
1110 030340                                ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
     030340  104456                                                            TRAP    C$ERHRD
     030342  001144                                                            .WORD   612
     030344  004745                                                            .WORD   PKTRAM
     030346  015570                                                            .WORD   RAMERR
1111 030350                          40$:   ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     030350                                                                 10000$:
     030350  104405                                                            TRAP    C!ESEG
1112 030352  011300                        MOV     (R3),R0                 ;NEXT PACKET COMMAND WORD
```

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 CHARACTERISTCS
1117 030372  020327  003062    45$:      CMP     R3,#TBLEND          ;HAVE WE COMPLETED DATA TABLE ?
1118 030376  103002                      BHIS    50$                 ;BRANCH IF ALL TESTED
1119 030400  000137  030114              JMP     5$                  ;TEST WITH NEXT DATA
1120
1121 030404                      50$:    ENDSUB                      ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     030404                                                                   L10054:
     030404  104403                                                           TRAP    C$ESUB
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$:    EXIT    TST                 ;ALL DONE THIS TEST
     030432  104432                                                           TRAP    C$EXIT
     030434  000346                                                           .WORD   L10052-.
1129
1130                             ;+
1131                             ;LOCAL STORAGE FOR THIS TEST
1132                             ;-
1133
1135 030436                              .BLKB   10-<.-TSV2&7>
1137 030440                      T6PACKET:                           ;COMMAND PACKET FOR TEST
1138 030440  000000                      .WORD   0                   ;WILL CONTAIN VARIABLE COMMANDS
1139 030442  052525                      .WORD   052525
1140 030444  125252                      .WORD   125252
1141 030446  052525                      .WORD   052525
1142
1143
1144                             ;+
1145                             ;LOCAL TEXT MESSAGES FOR TEST
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                                     .EVEN
1155 031002                             ENDTST
     031002                                                           L10052:
     031002  104401                                                           TRAP    C$ETST
1156
1157                                     .SBTTL  TEST  7: WRITE CHARACTERISTICS
1158
1159
1160                             ;       THIS TEST VERIFIES BASIC OPERATION OF THE WRITE CHARACTERISTICS
1161                             ;       COMMAND.  IT VERIFIES THAT THE COMMAND BLOCK AND CHARACTERISTICS
1162                             ;       DATA BLOCK ARE FETCHED PROPERLY FROM CPU MEMORY, THE NEED BUFFER
1163                             ;       ADDRESS (NBA) BIT IN TSSR IS HANDLED PROPERLY, AND THAT A PROPER
1164                             ;       MESSAGE PACKET IS STORED, WHERE APPROPRIATE.  THIS TEST DOES NOT
1165                             ;       CHECK THAT THE VARIOUS FUNCTIONS ENABLED BY CHARACTERISTIC MODE
```

TEST 7: WRITE CHARACTERISTICS

```
                                ;       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.
                                ;
                                ;       THIS TEST CHECKS VARIOUS MICROPROGRAM SEQUENCES, COMMAND
                                ;       DECODING, DMA LOGIC, AND BASIC PACKET PROTOCOL HANDLING.  THIS
                                ;       IS THE FIRST TEST IN WHICH DATO DMA CYCLES (FOR STORING THE
                                ;       MESSAGE PACKET) ARE PERFORMED.  ANY ERRORS IN THE BODY OF THE
                                ;       TEST (I.E, ERRORS OTHER THAN INITIALIZATION ERRORS RELATED TO
                                ;       THE TRANSPORT BUS) DEFINITELY INDICATE A BAD M7196 MODULE.
                                ;
                                ;
                                ;
031004                                  BGNTST
031004                                                                  T7::
031004  012700  034363                  MOV     #TST7ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
031010  004737  016570                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
031014  012737  000024  002214          MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
031022                          T7LOOP:
031022                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
031022                                                                          T7.1:
031022  104402                                                                  TRAP    C$BSUB
031024  004737  016054                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
031030  103405                          BCS     1$                      ;BR IF SOFT INIT = OK
031032  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
031034                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
031034  104455                                                                  TRAP    C$ERDF
031036  001275                                                                  .WORD   701
031040  003652                                                                  .WORD   SFIERR
031042  012104                                                                  .WORD   SFIMSG
031044                          1$:
031044  012704  045070                  MOV     #T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
031050  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
031054  103404                          BCS     2$                      ;BR, IF COMMAND ISSUED OK
031056                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
031056  104456                                                                  TRAP    C$ERHRD
031060  001276                                                                  .WORD   702
031062  005056                                                                  .WORD   WRTMSG
031064  012104                                                                  .WORD   SFIMSG
031066                          2$:
031066  004737  034412                  JSR     PC,T7REST               ;SET PACKET TO START-UP VALUES
                                
031072                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
031072  012700  000000                                                          MOV     #PRI00,R0
031076  104441                                                                  TRAP    C$SPRI
031100  012703  002764                  MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
031104  012704  033230                  MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
031110  012764  000010  000006          MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
031116                          5$:
031116                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
031116  104404                                                                  TRAP    C$BSEG
031120  004737  016054                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
```

N10

TEST 7: WRITE CHARACTERISTICS

```
1219 031124  103405                    BCS     10$                     ;BR IF SOFT INIT = OK
1223 031126  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1224 031130                            ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     031130  104455                                                            TRAP    C$ERDF
     031132  001277                                                            .WORD   703
     031134  003652                                                            .WORD   SFIERR
     031136  012104                                                            .WORD   SFIMSG
1225
1226 031140  005037  002220      10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1227 031144  005037  002222              CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1228 031150  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
1229 031154  004737  016416              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1230 031160  103407                    BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1231 031162  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1235 031164                            ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     031164  104455                                                            TRAP    C$ERDF
     031166  001300                                                            .WORD   704
     031170  034031                                                            .WORD   T7SSR
     031172  012116                                                            .WORD   PKTSSR
1236 031174  005237  002220              INC     FATFLG                  ;SET FATAL ERROR FLAG
1237 031200                      15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     031200  104406                                                            TRAP    C$CLP1
1238 031202                            ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     031202  104410                                                            TRAP    C$ESCAPE
     031204  000164                                                            .WORD   10000$-.
1239 031206  005737  002222              TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1240 031212  001404                    BEQ     22$                     ;BRANCH IF NOT
1244 031214                            ERRHRD  ERRNO,T7INT,PKTSSR
     031214  104456                                                            TRAP    C$ERHRD
     031216  001301                                                            .WORD   705
     031220  034211                                                            .WORD   T7INT
     031222  012116                                                            .WORD   PKTSSR
1245 031224  016501  000002      22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1246 031230  012702  000200              MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
1247 031234  032701  000100              BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1248 031240  001402                    BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1249 031242  052702  000100              BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1250 031246  020201              25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1251 031250  001404                    BEQ     30$                     ;OKAY IF MATCH
1255 031252                            ERRHRD  ERRNO,T7NBA,PKTSSR      ;NBA NOT ZERO
     031252  104456                                                            TRAP    C$ERHRD
     031254  001302                                                            .WORD   706
     031256  033370                                                            .WORD   T7NBA
     031260  012116                                                            .WORD   PKTSSR
1256 031262                      30$:    CKLOOP                          ;LOOP ON ERROR ?
     031262  104406                                                            TRAP    C$CLP1
1257 031264  004737  016416              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1258 031270  016501  000000              MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
1259 031274  012702  033256              MOV     #T7BFR,R2               ;START OF THE DATA BUFFER
1260 031300  022737  000002  002226      CMP     #2,REV                  ;IS IT A NEW MICROCODE
1261 031306  001404                    BEQ     31$                     ;YES BR
1262 031310  032762  000200  000012      BIT     #BIT7,XST2(R2)          ;IS EXTENDED FEATURES BIT SET ?
1263 031316  001405                    BEQ     32$                     ;BRANCH IF EXTENDED FEATURES OFF
1264 031320  012737  000001  002224  31$:    MOV     #1,EXTFEA               ;SET EXTENDED FEATURES FOR SUBTEST 6
1265 031326  062702  000002              ADD     #2,R2                   ;EXTRA WORD IF SPECIAL FEATURES
1266 031332  062702  000016      32$:    ADD     #14.,R2                 ;EXPECTED CONTENTS OF TSDA
1267 031336  020102                    CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
```

```
TEST  7: WRITE CHARACTERISTICS

1268 031340  001404                    BEQ     35$                ;ERROR IF NOT EQUAL
1272 031342                            ERRHRD  ERRNO,T7TSBA,EXPREC ;PRINT THE ERROR & EXPD/RECV
     031342  104456                                                       TRAP    C$ERHRD
     031344  001303                                                       .WORD   707
     031346  034300                                                       .WORD   T7TSBA
     031350  015554                                                       .WORD   EXPREC
1273
1274
1275 031352  004737  011154    35$:    JSR     PC,CKRAM           ;SEE IF DATA IN RAM IS CORRECT
1276 031356  103404                    BCS     40$                ;BRANCH IF PACKET IN RAM IS CORRECT
1280 031360                            ERRHRD  ERRNO,PKTRAM,RAMERR ;REPORT THE RAM ERROR(S)
     031360  104456                                                       TRAP    C$ERHRD
     031362  001304                                                       .WORD   708
     031364  004745                                                       .WORD   PKTRAM
     031366  015570                                                       .WORD   RAMERR
1281
1282 031370                    40$:    ENDSEG                     ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     031370                                                            10000$:
     031370  104405                                                       TRAP    C$ESEG
1283
1284 031372  012364  000006            MOV     (R3)+,PKBCNT(R4)   ;SET THE TEST WORD
1285 031376  020327  003062            CMP     R3,#TBLEND         ;HAS ALL DATA BEEN TESTED ?
1286 031402  103002                    BHIS    55$                ;BRANCH IF ALL DATA DONE
1287 031404  000137  031116            JMP     5$                 ;BRANCH TILL BACK TO ZERO
1288
1289 031410                    55$:    ENDSUB                     ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     031410                                                            L10056:
     031410  104403                                                       TRAP    C$ESUB
1290
1291 031412  005737  002220            TST     FATFLG             ;ANY FATAL ERRORS ?
1292 031416  001402                    BEQ     60$                ;BRANCH IF NOT
1293 031420  004737  017262            JSR     PC,CKDROP          ;TRY TO DROP THE UNIT
1294 031424                    60$:
1295
1296                                   ;+
1297                                   ;
1298                                   ;TEST 7, SUBTEST 2
1299                                   ;
1300                                   ;CHECK THAT UNUSED BITS BEING SET CAUSES
1301                                   ;WRITE CHARACTERISTICS COMMAND TO BE REJECTED
1302                                   ;
1303                                   ;-
1304
1305 031424                            BGNSUB                     ;////////////// BEGIN SUBTEST //////////////
     031424                                                            T7.2:
     031424  104402                                                       TRAP    C$BSUB
1306
1307 031426                            SETPRI  #PRI00             ;LOWER PRIORITY TO ALLOW INTERRUPTS
     031426  012700  000000                                                MOV     #PRI00,R0
     031432  104441                                                       TRAP    C$SPRI
1308 031434  012703  033276            MOV     #T72DATA,R3        ;START OF TEST DATA FOR SUBTEST
1309 031440  012704  033230    5$:     MOV     #T7PACKET,R4       ;GET THE ADDRESS OF COMMAND PACKET
1310 031444  004737  034412            JSR     PC,T7REST          ;RESTORE PACKET TO STARTING VALUES
1311
1312 031450                            BGNSEG                     ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     031450  104404                                                       TRAP    C$BSEG
1313
```

TEST 7: WRITE CHARACTERISTICS

```
1314 031452  004737  016054            JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1315 031456  103405                    BCS     10$                     ;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    C$ERDF
     031464  001305                                                            .WORD   709
     031466  003652                                                            .WORD   SFIERR
     031470  012104                                                            .WORD   SFIMSG
1321 031472  005037  002222      10$:  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,TSDB(R5)             ;SET THE PACKET ADDRESS
1326 031512  004737  016330            JSR     PC,WAITF                ;WAIT FOR SSR TO SET
1327 031516  103405                    BCS     15$                     ;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    C$ERDF
     031524  001306                                                            .WORD   710
     031526  034031                                                            .WORD   T7SSR
     031530  012116                                                            .WORD   PKTSSR
1333 031532            15$:  CKLOOP                                   ;LOOP ON ERROR, IF FLAG SET
     031532  104406                                                            TRAP    C$CLP1
1334 031534                            ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
     031534  104410                                                            TRAP    C$ESCAPE
     031536  000116                                                            .WORD   10000$-.
1335 031540  005737  002222            TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1336 031544  001404                    BEQ     22$                     ;BRANCH IF NOT
1340 031546                            ERRHRD  ERRNO,T7INT,PKTSSR
     031546  104456                                                            TRAP    C$ERHRD
     031550  001307                                                            .WORD   711
     031552  034211                                                            .WORD   T7INT
     031554  012116                                                            .WORD   PKTSSR
1341 031556  016501  000002      22$:  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     25$                     ;BRANCH IF NOT OFF-LINE
1345 031574  052702  000100            BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1346 031600  020201      25$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1347 031602  001414                    BEQ     30$                     ;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     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
1355 031624                            ERRHRD  ERRNO,T72REJ,PKTSSR     ;COMMAND NOT REJECTED
     031624  104456                                                            TRAP    C$ERHRD
     031626  001310                                                            .WORD   712
     031630  033443                                                            .WORD   T72REJ
     031632  012116                                                            .WORD   PKTSSR
1356 031634            30$:  CKLOOP                                   ;LOOP ON ERROR ?
     031634  104406                                                            TRAP    C$CLP1
1357 031636  032701  002000            BIT     #NBA,R1                 ;IS NBA BIT SET ?
1358 031642  001004                    BNE     35$                     ;OKAY IF NBA SET
1362 031644                            ERRHRD  ERRNO,T72NBA,PKTSSR     ;NBA NOT SET
     031644  104456                                                            TRAP    C$ERHRD
     031646  001311                                                            .WORD   713
     031650  033312                                                            .WORD   T72NBA
```

```
TEST  7: WRITE CHARACTERISTICS

      031652  012116                                                                .WORD   PKTSSR
 1363 031654                        35$:
 1364 031654                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      031654                                                                       10000$:
      031654  104405                                                                TRAP    C$ESEG
 1365
 1366 031656  062703  000004                ADD     #4,R3                   ;POINT TO NEXT DATA PAIR
 1367 031662  020327  033312                CMP     R3,#T72DONE             ;COMPARE TO END OF TEST DATA
 1368 031666  103002                        BHIS    57$                     ;BRANCH IF ALL DATA TESTED
 1369 031670  000137  031440                JMP     5$                      ;BRANCH TILL BACK TO ZERO
 1370
 1371 031674                        57$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      031674                                                                       L10057:
      031674  104403                                                                TRAP    C$ESUB
 1372
 1373                               ;*
 1374                               ;
 1375                               ;TEST 7, SUBTEST 3
 1376                               ;
 1377                               ;CHECK THE WRITE CHARACTERISTICS COMMAND IS REJECTED
 1378                               ;IF ISSUED WITH AN INVALID DATA BLOCK BYTE COUNT
 1379                               ;
 1380                               ;-
 1381
 1382 031676                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
      031676                                                                       T7.3:
      031676  104402                                                                TRAP    C$BSUB
 1383
 1384 031700                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      031700  012700  000000                                                        MOV     #PRI00,R0
      031704  104441                                                                TRAP    C$SPRI
 1385 031706  012703  000001                MOV     #1,R3                   ;STARTING BYTE COUNT
 1386 031712  012704  033230        5$:     MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
 1387 031716  004737  034412                JSR     PC,T7REST               ;RESTORE PACKET TO STARTING VALUES
 1388
 1389 031722                                BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
      031722  104404                                                                TRAP    C$BSEG
 1390
 1391 031724  004737  016054                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 1392 031730  103405                        BCS     10$                     ;BR IF SOFT INIT = OK
 1396 031732  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 1397 031734                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      031734  104455                                                                TRAP    C$ERDF
      031736  001312                                                                .WORD   714
      031740  003652                                                                .WORD   SFIERR
      031742  012104                                                                .WORD   SFIMSG
 1398 031744  005037  002222        10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 1399 031750  010364  000006                MOV     R3,PKBCNT(R4)           ;INSERT THE BYTE COUNT FOR TEST
 1400 031754  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 1401 031760  004737  016330                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 1402 031764  103405                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 1403 031766  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 1407 031770                                ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      031770  104455                                                                TRAP    C$ERDF
      031772  001313                                                                .WORD   715
      031774  034031                                                                .WORD   T7SSR
      031776  012116                                                                .WORD   PKTSSR
```

TEST 7: WRITE CHARACTERISTICS

```
1408 032000                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     032000  104406                                                                  TRAP    C$CLP1
1409 032002                                  ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     032002  104410                                                                  TRAP    C$ESCAPE
     032004  000116                                                                  .WORD   10000$-.
1410 032006  005737  002222                     TST     INTRECV             ;DID AN INTERRUPT OCCUR ?
1411 032012  001404                             BEQ     22$                 ;BRANCH IF NOT
1415 032014                                     ERRHRD  ERRNO,T7INT,PKTSSR
     032014  104456                                                                  TRAP    C$ERHRD
     032016  001314                                                                  .WORD   716
     032020  034211                                                                  .WORD   T7INT
     032022  012116                                                                  .WORD   PKTSSR
1416 032024  016501  000002          22$:       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                             BEQ     25$                 ;BRANCH IF NOT OFF-LINE
1420 032042  052702  000100                     BIS     #OFL,R2             ;SET OFF-LINE IN EXPECTED DATA
1421 032046  020201          25$:       CMP     R2,R1               ;DOES EXPECTED MATCH RECEIVED ?
1422 032050  001414                             BEQ     30$                 ;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                             BEQ     30$                 ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
1430 032072                                     ERRHRD  ERRNO,T73REJ,PKTSSR ;COMMAND NOT REJECTED
     032072  104456                                                                  TRAP    C$ERHRD
     032074  001315                                                                  .WORD   717
     032076  033542                                                                  .WORD   T73REJ
     032100  012116                                                                  .WORD   PKTSSR
1431 032102                          30$:    CKLOOP                          ;LOOP ON ERROR ?
     032102  104406                                                                  TRAP    C$CLP1
1432 032104  032701  002000                     BIT     #NBA,R1             ;IS NBA BIT SET ?
1433 032110  001004                             BNE     35$                 ;OKAY IF NBA SET
1437 032112                                     ERRHRD  ERRNO,T72NBA,PKTSSR ;NBA NOT SET
     032112  104456                                                                  TRAP    C$ERHRD
     032114  001316                                                                  .WORD   718
     032116  033312                                                                  .WORD   T72NBA
     032120  012116                                                                  .WORD   PKTSSR
1438 032122                          35$:
1439 032122                                  ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     032122                                                                  10000$:
     032122  104405                                                                  TRAP    C$ESEG
1440
1441 032124  005203                             INC     R3                  ;NEXT BYTE COUNT
1442 032126  020327  000006                     CMP     R3,#6               ;TESTED ALL INVALID ?
1443 032132  002002                             BGE     57$                 ;BRANCH IF TEST DONE
1444 032134  000137  031712                     JMP     5$                  ;BRANCH TILL BACK TO ZERO
1445
1446 032140                          57$:    ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     032140                                                                  L10060:
     032140  104403                                                                  TRAP    C$ESUB
1447
1448                                 ;*
1449                                 ;
1450                                 ;TEST 7, SUBTEST 4
1451                                 ;
1452                                 ;SUBTEST TO VERIFY THAT A WRITE CHARACTERISTICS COMMAND IS
1453                                 ;REJECTED IF AN ILLEGAL DATA BLOCK ADDRESS IS ISSUED.
```

TEST 7: WRITE CHARACTERISTICS

```
1454                                    ;
1455                                    ;-
1456
1457 032142                                     BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     032142                                                                                             T7.4:
     032142  104402                                                                                             TRAP    C$BSUB
1458
1459 032144                                     SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     032144  012700  000000                                                                                     MOV     #PRI00,R0
     032150  104441                                                                                             TRAP    C$SPRI
1460 032152  012703  033276                     MOV     #T72DATA,R3             ;START OF TEST DATA FOR SUBTEST
1461 032156  012704  033230             5$:     MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1462 032162  004737  034412                     JSR     PC,T7REST               ;RESTORE PACKET TO STARTING VALUES
1463
1464
1465 032166  004737  016054                     JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1466 032172  103405                             BCS     10$                     ;BR IF SOFT INIT = OK
1470 032174  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1471 032176                                     ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     032176  104455                                                                                             TRAP    C$ERDF
     032200  001317                                                                                             .WORD   719
     032202  003652                                                                                             .WORD   SFIERR
     032204  012104                                                                                             .WORD   SFIMSG
1472 032206  005037  002222             10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1473 032212  052737  000001  033240             BIS     #1,T7DATA               ;MAKE ADDRESS ODD
1474 032220  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
1475 032224  004737  016330                     JSR     PC,WAITF                ;WAIT FOR SSR TO SET
1476 032230  103405                             BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1477 032232  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1481 032234                                     ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     032234  104455                                                                                             TRAP    C$ERDF
     032236  001320                                                                                             .WORD   720
     032240  034031                                                                                             .WORD   T7SSR
     032242  012116                                                                                             .WORD   PKTSSR
1482 032244                             15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     032244  104406                                                                                             TRAP    C$CLP1
1483 032246                                     ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
     032246  104410                                                                                             TRAP    C$ESCAPE
     032250  000116                                                                                             .WORD   L10061-.
1484 032252  005737  002222                     TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1485 032256  001404                             BEQ     22$                     ;BRANCH IF NOT
1489 032260                                     ERRHRD  ERRNO,T7INT,PKTSSR
     032260  104456                                                                                             TRAP    C$ERHRD
     032262  001321                                                                                             .WORD   721
     032264  034211                                                                                             .WORD   T7INT
     032266  012116                                                                                             .WORD   PKTSSR
1490 032270  016501  000002             22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1491 032274  012702  102206                     MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
1492 032300  032701  000100                     BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1493 032304  001402                             BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1494 032306  052702  000100                     BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1495 032312  020201                     25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1496 032314  001414                             BEQ     30$                     ;OKAY IF MATCH
1497 032316  010100                             MOV     R1,R0                   ;DATA FROM TSSR
1498 032320                                     XOR     R2,R0                   ;FIND BITS IN ERROR
1499 032330  020027  002000                     CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
1500 032334  001404                             BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
```

```
TEST  7: WRITE CHARACTERISTICS

1504 032336                             ERRHRD  ERRNO,T74REJ,PKTSSR     ;COMMAND NOT REJECTED
     032336  104456                                                             TRAP    C$ERHRD
     032340  001322                                                             .WORD   722
     032342  033635                                                             .WORD   T74REJ
     032344  012116                                                             .WORD   PKTSSR
1505 032346                     30$:    CKLOOP                          ;LOOP ON ERROR ?
     032346  104406                                                             TRAP    C$CLP1
1506 032350  032701  002000             BIT     #NBA,R1                 ;IS NBA BIT SET ?
1507 032354  001004                     BNE     35$                     ;OKAY IF NBA SET
1511 032356                             ERRHRD  ERRNO,T72NBA,PKTSSR     ;NBA NOT SET
     032356  104456                                                             TRAP    C$ERHRD
     032360  001323                                                             .WORD   723
     032362  033312                                                             .WORD   T72NBA
     032364  012116                                                             .WORD   PKTSSR
1512
1513 032366                     35$:    ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     032366                                                                     L10061:
     032366  104403                                                             TRAP    C$ESUB
1514
1515
1516                            ;*
1517                            ;
1518                            ;TEST 7, SUBTEST 5
1519                            ;
1520                            ;CHECK THAT WRITE CHARACTERISTICS COMMAND IS REJECTED IF THE
1521                            ;MESSAGE BUFFER LENGTH SPECIFIES AN INVALID COUNT (LESS THAN 14)
1522                            ;
1523                            ;-
1524
1525 032370                             BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     032370                                                                     T7.5:
     032370  104402                                                             TRAP    C$BSUB
1526
1527 032372                             SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     032372  012700  000000                                                     MOV     #PRI00,R0
     032376  104441                                                             TRAP    C$SPRI
1528 032400  012703  000001             MOV     #1,R3                   ;STARTING BUFFER LENGTH
1529 032404  012704  033230     5$:     MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1530 032410  004737  034412             JSR     PC,T7REST               ;RESTORE PACKET TO STARTING VALUES
1531
1532 032414                             BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     032414  104404                                                             TRAP    C$BSEG
1533
1534 032416  004737  016054             JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1535 032422  103405                     BCS     10$                     ;BR IF SOFT INIT = OK
1539 032424  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1540 032426                             ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     032426  104455                                                             TRAP    C$ERDF
     032430  001324                                                             .WORD   724
     032432  003652                                                             .WORD   SFIERR
     032434  012104                                                             .WORD   SFIMSG
1541 032436  005037  002222     10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1542 032442  010337  033244             MOV     R3,T7DATA+4             ;INSERT THE BAD MESSAGE LENGTH
1543 032446  010465  000000             MOV     R4,TSD8(R5)             ;SET THE PACKET ADDRESS
1544 032452  004737  016330             JSR     PC,WAITF                ;WAIT FOR SSR TO SET
1545 032456  103405                     BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1546 032460  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

TEST 7: WRITE CHARACTERISTICS

```
1550 032462                                ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     032462  104455                                                                TRAP    C$ERDF
     032464  001325                                                                .WORD   725
     032466  034031                                                                .WORD   T7SSR
     032470  012116                                                                .WORD   PKTSSR
1551 032472                          15$:  CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
     032472  104406                                                                TRAP    C$CLP1
1552 032474                                ESCAPE  SEG                             ;BY-PASS SUBTEST IF FATAL ERROR
     032474  104410                                                                TRAP    C$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                                ERRHRD  ERRNO,T7INT,PKTSSR
     032506  104456                                                                TRAP    C$ERHRD
     032510  001326                                                                .WORD   726
     032512  034211                                                                .WORD   T7INT
     032514  012116                                                                .WORD   PKTSSR
1559 032516  016501  000002          22$:  MOV     TSSR(R5),R1                     ;GET THE CONTENTS OF TSSR
1560 032522  012702  102206                MOV     @SC!SSR!TSREJ!NBA,R2            ;EXPECTED CONTENTS OF TSSR
1561 032526  032701  000100                BIT     @OFL,R1                         ;IS OFF-LINE BIT SET ?
1562 032532  001402                        BEQ     25$                             ;BRANCH IF NOT OFF-LINE
1563 032534  052702  000100                BIS     @OFL,R2                         ;SET OFF-LINE IN EXPECTED DATA
1564 032540  020201          25$:  CMP     R2,R1                           ;DOES EXPECTED MATCH RECEIVED ?
1565 032542  001414                        BEQ     30$                             ;OKAY IF MATCH
1566 032544  010100                        MOV     R1,R0                           ;DATA FROM TSSR
1567 032546                                XOR     R2,R0                           ;FIND BITS IN ERROR
1568 032556  020027  002000                CMP     R0,@NBA                         ;IS NBA ONLY BIT IN ERROR ?
1569 032562  001404                        BEQ     30$                             ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
1573 032564                                ERRHRD  ERRNO,T75REJ,PKTSSR             ;COMMAND NOT REJECTED
     032564  104456                                                                TRAP    C$ERHRD
     032566  001327                                                                .WORD   727
     032570  033733                                                                .WORD   T75REJ
     032572  012116                                                                .WORD   PKTSSR
1574 032574                          30$:  CKLOOP                                  ;LOOP ON ERROR ?
     032574  104406                                                                TRAP    C$CLP1
1575 032576  032701  002000                BIT     @NBA,R1                         ;IS NBA BIT SET ?
1576 032602  001004                        BNE     35$                             ;OKAY IF NBA SET
1580 032604                                ERRHRD  ERRNO,T72NBA,PKTSSR             ;NBA NOT SET
     032604  104456                                                                TRAP    C$ERHRD
     032606  001330                                                                .WORD   728
     032610  033312                                                                .WORD   T72NBA
     032612  012116                                                                .WORD   PKTSSR
1581 032614                          35$:
1582 032614                                ENDSEG                                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     032614                                                                     10000$:
     032614  104405                                                                TRAP    C$ESEG
1583
1584 032616  005203                        INC     R3                              ;NEXT BUFFER LENGTH
1585 032620  020327  000016                CMP     R3,@14.                         ;HAVE ALL BAD VALUES BEEN TESTED ?
1586 032624  002002                        BGE     57$                             ;BRANCH IF ALL TESTED
1587 032626  000137  032404                JMP     5$                              ;BRANCH TILL BACK TO ZERO
1588
1589 032632                          57$:  ENDSUB                                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     032632                                                                     L10062:
     032632  104403                                                                TRAP    C$ESUB
1590
1591                                 ;*
```

TEST 7: WRITE CHARACTERISTICS

```
                                        ;
                                        ;TEST 7, SUBTEST 6
                                        ;
                                        ;THIS SUBTEST IS EXECUTED ONLY IF THE EXTENDED
                                        ;FEATURES MODE IS ENABLED (AS DETERMINED BY EXAMINING
                                        ;XST2 AFTER A PREVIOUS EXECUTION OF WRITE CHARACTERISTICS).
                                        ;IT VERIFIES THAT A FIFTH CHARACTERISTICS DATA WORD IS FETCHED
                                        ;IF THE BYTE COUNT PARAMETER IN THE COMMAND PACKET IS 10 DECIMAL
                                        ;OR GREATER.
                                        ;
                                        ;-
1603 032634                             BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     032634                                                                                  T7.6:
     032634  104402                                                                             TRAP    C$BSUB
1604 032636  005737  002224            TST     EXTFEA                  ;IS EXTENDED FEATURES SOFT. SW SET?
1605 032642  001002                    BNE     4$                      ;BR. IF SOFTWARE SWITCH IS SET (ON)
1606 032644                            EXIT    TST                     ;NO EXTENDED FEATURES EXIT THIS TEST
     032644  104432                                                                             TRAP    C$EXIT
     032646  001612                                                                             .WORD   L10055-.
1607 032650  004737  034412      4$:   JSR     PC,T7REST               ;SET PACKET TO START-UP VALUES
1608
1609 032654                            SETPRI  @PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     032654  012700  000000                                                                     MOV     @PRI00,R0
     032660  104441                                                                             TRAP    C$SPRI
1610 032662  012703  002766            MOV     @TSTBLK+12.,R3          ;START OF TEST DATA
1611 032666  012704  033230            MOV     @T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1612 032672  012737  000200  033250    MOV     @200,T7SP               ;SPECIAL BIT SET FOR EXTFEA RAM RD
1613 032700  012764  000012  000006    MOV     @10.,PKBCNT(R4)         ;START WITH EXTENDED FEATURES VALUE
1614 032706                      5$:
1615 032706                            BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     032706  104404                                                                             TRAP    C$BSEG
1616
1617 032710  004737  016054            JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1618 032714  103405                    BCS     10$                     ;BR IF SOFT INIT = OK
1622 032716  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1623 032720                            ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     032720  104455                                                                             TRAP    C$ERDF
     032722  001331                                                                             .WORD   729
     032724  003652                                                                             .WORD   SFIERR
     032726  012104                                                                             .WORD   SFIMSG
1624 032730  005037  002220      10$:  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1625 032734  005037  002222            CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1626 032740  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
1627 032744  004737  016416            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1628 032750  103407                    BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1629 032752  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1633 032754                            ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     032754  104455                                                                             TRAP    C$ERDF
     032756  001332                                                                             .WORD   730
     032760  034031                                                                             .WORD   T7SSR
     032762  012116                                                                             .WORD   PKTSSR
1634 032764  005237  002220            INC     FATFLG                  ;SET FATAL ERROR FLAG
1635 032770                      15$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     032770  104406                                                                             TRAP    C$CLP1
1636 032772                            ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     032772  104410                                                                             TRAP    C$ESCAPE
     032774  000166                                                                             .WORD   10000$-.
```

TEST 7: WRITE CHARACTERISTICS

```
1637 032776  005737  002222            TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1638 033002  001404                    BEQ     22$                     ;BRANCH IF NOT
1642 033004                            ERRHRD  ERRNO,T7INT,PKTSSR
     033004  104456                                                    TRAP    C$ERHRD
     033006  001333                                                    .WORD   731
     033010  034211                                                    .WORD   T7INT
     033012  012116                                                    .WORD   PKTSSR
1643 033014  016501  000002      22$:  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1644 033020  012702  000200            MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
1645 033024  032701  000100            BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1646 033030  001402                    BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1647 033032  052702  000100            BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1648 033036  020201      25$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1649 033040  001404                    BEQ     30$                     ;OKAY IF MATCH
1653 033042                            ERRHRD  ERRNO,T7NBA,PKTSSR      ;NBA NOT ZERO
     033042  104456                                                    TRAP    C$ERHRD
     033044  001334                                                    .WORD   732
     033046  033370                                                    .WORD   T7NBA
     033050  012116                                                    .WORD   PKTSSR
1654 033052      30$:  CKLOOP                                  ;LOOP ON ERROR ?
     033052  104406                                                    TRAP    C$CLP1
1655 033054  004737  016416            JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1656 033060  016501  000000            MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
1657 033064  012702  033256            MOV     #T7BFR,R2               ;START OF THE DATA BUFFER
1658 033070  022737  000002  002226    CMP     #2,REV                  ;IS IT A NEW MICROCODE
1659 033076  001404                    BEQ     31$                     ;YES BR
1660 033100  032762  000200  000012    BIT     #BIT7,XST2(R2)          ;IS EXTENDED FEATURES BIT SET ?
1661 033106  001402                    BEQ     32$                     ;BRANCH IF EXTENDED FEATURES OFF
1662 033110  062702  000002      31$:  ADD     #2,R2                   ;EXTRA WORD IF SPECIAL FEATURES
1663 033114  062702  000016      32$:  ADD     #14.,R2                 ;EXPECTED CONTENTS OF TSDA
1664 033120  020102                    CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1665 033122  001404                    BEQ     35$                     ;ERROR IF NOT EQUAL
1669 033124                            ERRHRD  ERRNO,T7TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     033124  104456                                                    TRAP    C$ERHRD
     033126  001335                                                    .WORD   733
     033130  034300                                                    .WORD   T7TSBA
     033132  015554                                                    .WORD   EXPREC
1670
1671
1672 033134  012704  033240      35$:  MOV     #T7DATA,R4              ;SET POINTER FOR CHECKER
1673 033140  004737  011264            JSR     PC,CKRAM2               ;SEE IF DATA IN RAM IS CORRECT
1674 033144  103404                    BCS     40$                     ;BRANCH IF PACKET IN RAM IS CORRECT
1678 033146                            ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
     033146  104456                                                    TRAP    C$ERHRD
     033150  001336                                                    .WORD   734
     033152  004745                                                    .WORD   PKTRAM
     033154  015570                                                    .WORD   RAMERR
1679
1680 033156  012704  033230      40$:  MOV     #T7PACKET,R4            ;RESET PACKET POINTER
1681 033162                            ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     033162                                                                 10000$:
     033162  104405                                                    TRAP    C$ESEG
1682
1683 033164  012364  000006            MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
1684 033170  020327  003062            CMP     R3,#TBLEND              ;HAS ALL DATA BEEN TESTED ?
1685 033174  103002                    BHIS    55$                     ;BRANCH IF ALL DATA DONE
1686 033176  000137  032706            JMP     5$                      ;BRANCH TILL BACK TO ZERO
```

TEST 7: WRITE CHARACTERISTICS

```
                                55$:    ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
                                                                                        L10063:
                                                                                                TRAP    C$ESUB

                                        TST     FATFLG                  ;ANY FATAL ERRORS ?
                                        BEQ     60$                     ;BRANCH IF NOT
                                        JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
                                60$:
                                        EXIT    TST                     ;ALL DONE THIS TEST
                                                                                                TRAP    C$EXIT
                                                                                                .WORD   L10055-.

                                ;+
                                ;LOCAL STORAGE FOR THIS TEST
                                ;-

                                        .BLKB   10-<.-TSV2&7>
                                T7PACKET:                               ;COMMAND PACKET FOR TEST
                                        .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
                                        .WORD   T7DATA                  ;ADDRESS OF CHARACTERISTICS BLOCK
                                        .WORD   0
                                        .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE

                                T7DATA:                                 ;CHARACTERISTICS DATA BLOCK
                                        .WORD   T7BFR                   ;ADDRESS OF MESSAGE BUFFER
                                        .WORD   0
                                        .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
                                        .WORD   0
                                T7SP:   .WORD   0                       ;EXTFEA EXTRA WORD

                                        .WORD   0,0                     ;SPACE
                                T7BFR:  .BLKW   8.                      ;MESSAGE BUFFER

                                ;+
                                ;
                                ;TEST DATA FOR SUBTEST TWO
                                ;
                                ;DATA HAS FORMAT:
                                ;
                                ;       1ST WORD        OFFSET TO TEST WORD IN PACKET
                                ;       2ND WORD        BITS TO SET FOR TEST
                                ;
                                ;-

                                T72DATA:
                                        .WORD   0,BIT5!BIT6!BIT9!BIT10!BIT11!BIT12!BIT13
                                        .WORD   2,BIT0
                                        .WORD   4,BIT6!BIT15
                                T72DONE=.

                                ;+
                                ;LOCAL TEXT MESSAGES FOR TEST
                                ;-

                                T72NBA: .ASCIZ  'NBA Not Set On Rejected WRITE CHARACTERISTICS'
```

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                                  ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
1758                                  ;
1759                                  ;-
1760
1761 034412                           T7REST:
1762 034412                                   SAVREG                          ;SAVE THE REGISTERS
1763 034416  012701  033230                   MOV     #T7PACKET,R1            ;START OF THE PACKET
1764 034422  012721  100004                   MOV     #100004,(R1)+           ;WRITE CHARACTERISTICS WITH ACK
1765 034426  012721  033240                   MOV     #T7DATA,(R1)+           ;ADDRESS OF CHAR DATA BLOCK
1766 034432  005021                           CLR     (R1)+                   ;EXTENDED ADDRESS
1767 034434  012721  000010                   MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
1768 034440  012721  033256                   MOV     #T7BFR,(R1)+            ;ADDRESS OF MESSAGE BUFFER
1769 034444  005021                           CLR     (R1)+
1770 034446  012721  000020                   MOV     #16.,(R1)+              ;LENGTH OF MESSAGE BUFFER
1771 034452  005021                           CLR     (R1)+
1772 034454  005011                           CLR     (R1)
1773 034456  000207                           RTS     PC                      ;RETURN
1774 034460                                   ENDTST
     034460
     034460  104401                                                   L10055:
                                                                             TRAP    C$ETST
1775
1776
1777                                  .SBTTL  TEST  8: VOLUME CHECK
1778
1779
1780                                  ;       THIS TEST VERIFIES THAT THE VOLUME CHECK (VCK) BIT, A FLAG HELD
1781                                  ;       WITHIN THE M7196 AND APPEARING IN XSTO, IS SET BY INITIALIZE AND
1782                                  ;       CLEARED BY EXECUTING A WRITE CHARACTERISTICS COMMAND WITH THE
1783                                  ;       CVC BIT SET.  IT IS ALSO VERIFIED THAT A WRITE CHARACTERISTICS
1784                                  ;       COMMAND WITH THE CVC BIT CLEAR DOES NOT AFFECT THE STATE OF THE
1785                                  ;       VOLUME CHECK BIT.  THE ACTUAL FUNCTION OF VOLUME CHECK, THAT OF
1786                                  ;       PREVENTING OR ALLOWING A TAPE MOTION COMMAND DEPENDING UPON
1787                                  ;       WHETHER VOLUME CHECK IS SET OR CLEAR, IS NOT CHECKED BY THIS
1788                                  ;       TEST;  THIS FUNCTIONALITY IS CHECKED IN THE INDIVIDUAL TESTS OF
1789                                  ;       TAPE MOTION COMMANDS.
1790                                  ;
1791                                  ;       THE TEST PROCEEDS AS FOLLOWS:
1792                                  ;
1793                                  ;            1.  THE CONTROLLER IS INITIALIZED BY WRITING INTO THE TSSR.
1794                                  ;
1795                                  ;            2.  A WRITE CHARACTERISTICS COMMAND IS ISSUED (WITH CVC=0)
1796                                  ;                AND XSTO IN THE RETURNED MESSAGE BUFFER IS EXAMINED;
```

```
                                        ;                   THE VCK BIT SHOULD BE CLEAR (0).
                                        ;
                                        ;               3.  THE PREVIOUS STEP IS REPEATED TO VERIFY THAT VCK DOES
                                        ;                   NOT CHANGE (REMAINS AT 0).
                                        ;
                                        ;               4.  A WRITE CHARACTERISTICS COMMAND IS ISSUED WITH CVC=1
                                        ;                   AND THE VCK BIT IN XST0 IN THE MESSAGE BUFFER IS
                                        ;                   EXAMINED;  THE VCK BIT SHOULD BE CLEAR (0).
                                        ;
                                        ;               5.  A WRITE CHARACTERISTICS COMMAND IS ISSUED WITH CVC=0
                                        ;                   AND THE VCK BIT IN XST0 IN THE MESSAGE BUFFER IS
                                        ;                   EXAMINED;  THE VCK BIT SHOULD REMAIN CLEAR (0).
                                        ;
                                        ;
1811 034462                                     BGNTST
     034462                                                                                 T8::
1816 034462  012700  035347                     MOV     #TST8ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
1817 034466  004737  016570                     JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
1818 034472  012737  000024  002214             MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
1819 034500                             T8LOOP:
1820
1821 034500  012704  035070                     MOV     #T8PACKET,R4            ;PACKET FOR WRITE CHARACTERISTICS
1822 034504  004737  016054             5$:     JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1823 034510  103405                             BCS     10$                     ;BR IF SOFT INIT = OK
1827 034512  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1828 034514                                     ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     034514  104455                                                                     TRAP    C$ERDF
     034516  001441                                                                     .WORD   801
     034520  003652                                                                     .WORD   SFIERR
     034522  012104                                                                     .WORD   SFIMSG
1829 034524  042714  040000             10$:    BIC     #BIT14,(R4)             ;CLEAR THE CVC BIT
1830 034530  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
1831 034534  004737  016416                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1832 034540  103405                             BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1833 034542  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1837 034544                                     ERRDF   ERRNO,T8SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     034544  104455                                                                     TRAP    C$ERDF
     034546  001442                                                                     .WORD   802
     034550  035260                                                                     .WORD   T8SSR
     034552  012116                                                                     .WORD   PKTSSR
1838 034554                             15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034554  104406                                                                     TRAP    C$CLP1
1839 034556                                     ESCAPE  TST                     ;EXIT IF FATAL ERROR
     034556  104410                                                                     TRAP    C$ESCAPE
     034560  000604                                                                     .WORD   L10064-.
1840 034562  012702  035112                     MOV     #T8BFR,R2               ;ADDRESS OF THE MESSAGE BUFFER
1841 034566  032762  000020  000006             BIT     #XSOVCK,XST0(R2)        ;IS VOLUME CHECK SET IN XST0 ?
1842 034574  001406                             BEQ     20$                     ;OKAY IF VOLUME CHECK IS CLEAR
1846 034576  016501  000002                     MOV     TSSR(R5),R1             ;CONTENTS OF TSSR FOR ERROR REPORT
1847 034602                                     ERRHRD  ERRNO,T8NVCK,PKTMES     ;VOLUME CHECK NOT CLEAR
     034602  104456                                                                     TRAP    C$ERHRD
     034604  001443                                                                     .WORD   803
     034606  035167                                                                     .WORD   T8NVCK
     034610  012160                                                                     .WORD   PKTMES
1848 034612                             20$:    CKLOOP                          ;LOOP ON ERROR ?
     034612  104406                                                                     TRAP    C$CLP1
1849 034614  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
```

TEST 8: VOLUME CHECK

```
1850 034620  004737  016416                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1851 034624  103405                        BCS     25$                     ;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    C$ERDF
     034632  001444                                                        .WORD   804
     034634  035260                                                        .WORD   T8SSR
     034636  012116                                                        .WORD   PKTSSR
1857 034640                        25$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034640  104406                                                        TRAP    C$CLP1
1858 034642                                ESCAPE  TST                     ;EXIT IF FATAL ERROR
     034642  104410                                                        TRAP    C$ESCAPE
     034644  000520                                                        .WORD   L10064-.
1859 034646  032762  000020  000006        BIT     #XSOVCK,XSTO(R2)        ;IS VOLUME CHECK SET IN XSTO ?
1860 034654  001406                        BEQ     30$                     ;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    C$ERHRD
     034664  001445                                                        .WORD   805
     034666  035167                                                        .WORD   T8NVCK
     034670  012160                                                        .WORD   PKTMES
1866 034672                        30$:    CKLOOP                          ;LOOP ON ERROR ?
     034672  104406                                                        TRAP    C$CLP1
1867 034674  052714  040000                BIS     #BIT14,(R4)             ;SET THE CVC BIT
1868 034700  010465  000000                MOV     R4,TSD0(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
1869 034704  004737  016416                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1870 034710  103405                        BCS     35$                     ;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    C$ERDF
     034716  001446                                                        .WORD   806
     034720  035260                                                        .WORD   T8SSR
     034722  012116                                                        .WORD   PKTSSR
1876 034724                        35$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034724  104406                                                        TRAP    C$CLP1
1877 034726                                ESCAPE  TST                     ;EXIT IF FATAL ERROR
     034726  104410                                                        TRAP    C$ESCAPE
     034730  000434                                                        .WORD   L10064-.
1878 034732  032762  000020  000006        BIT     #XSOVCK,XSTO(R2)        ;IS VOLUME CHECK CLEAR IN XSTO ?
1879 034740  001406                        BEQ     40$                     ;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    C$ERHRD
     034750  001447                                                        .WORD   807
     034752  035132                                                        .WORD   T8VCK
     034754  012160                                                        .WORD   PKTMES
1885 034756                        40$:    CKLOOP                          ;LOOP ON ERROR ?
     034756  104406                                                        TRAP    C$CLP1
1886 034760  042714  040000                BIC     #BIT14,(R4)             ;CLEAR THE CVC BIT
1887 034764  010465  000000                MOV     R4,TSD0(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
1888 034770  004737  016416                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1889 034774  103405                        BCS     45$                     ;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    C$ERDF
     035002  001450                                                        .WORD   808
     035004  035260                                                        .WORD   T8SSR
```

B12


```
TEST  8: VOLUME CHECK

        035006  012116                                                          .WORD   PKTSSR
1895    035010                          45$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        035010  104406                                                          TRAP    C$CLP1
1896    035012                          ESCAPE  TST                             ;EXIT IF FATAL ERROR
        035012  104410                                                          TRAP    C$ESCAPE
        035014  000350                                                          .WORD   L10064-.
1897    035016  032762  000020  000006        BIT     #XSOVCK,XSTO(R2)        ;IS VOLUME CHECK CLEAR IN XSTO ?
1898    035024  001406                        BEQ     50$                     ;OKAY IF VOLUME CHECK IS CLEARED
1902    035026  016501  000002                MOV     TSSR(R5),R1             ;CONTENTS OF TSSR FOR ERROR REPORT
1903    035032                                ERRHRD  ERRNO,T8VCK,PKTMES      ;VOLUME CHECK NOT CLEARED
        035032  104456                                                          TRAP    C$ERHRD
        035034  001451                                                          .WORD   809
        035036  035132                                                          .WORD   T8VCK
        035040  012160                                                          .WORD   PKTMES
1904    035042                          50$:    CKLOOP                          ;LOOP ON ERROR ?
        035042  104406                                                          TRAP    C$CLP1
1905    035044  004737  016536        60$:    JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
1906    035050  103002                        BCC     62$                     ;BRANCH IF NOT
1907    035052  000137  034500                JMP     T8LOOP                  ;LOOP UNTIL COUNT EXPIRED
1908    035056                          62$:    EXIT    TST                     ;ALL DONE THIS TEST
        035056  104432                                                          TRAP    C$EXIT
        035060  000304                                                          .WORD   L10064-.
1909
1910                                    ;+
1911                                    ;LOCAL STORAGE FOR THIS TEST
1912                                    ;-
1913
1915    035062                                  .BLKB   10-<.-TSV2&7>
1917    035070                          T8PACKET:
1918    035070  100004                          .WORD   100004                  ;COMMAND PACKET FOR TEST
1919    035072  035100                          .WORD   T8DATA                  ;WRITE CHARACTERISTICS COMMAND
1920    035074  000000                          .WORD   0                       ;ADDRESS OF CHARACTERISTICS BLOCK
1921    035076  000010                          .WORD   10                      ;STARTING VALUE OF COUNTER
1922
1923    035100                          T8DATA:                                 ;CHARACTERISTICS DATA BLOCK
1924    035100  035112                          .WORD   T8BFR                   ;ADDRESS OF MESSAGE BUFFER
1925    035102  000000                          .WORD   0
1926    035104  000020                          .WORD   16.                     ;LENGTH OF MESSAGE BUFFER
1927    035106  000000  000000                  .WORD   0,0
1928
1929    035112                          T8BFR:  .BLKW   8.                      ;MESSAGE BUFFER
1930
1931
1932                                    ;+
1933                                    ;LOCAL TEXT MESSAGES FOR TEST
1934                                    ;-
1935
1936    035132      126     157     154 T8VCK:  .ASCIZ  'Volume Check Bit Not Cleared'
1937    035167      126     157     154 T8NVCK: .ASCIZ  'Volume Check Bit (VCK) Not Clear After Initialize (XSTO)'
1938    035260      103     157     156 T8SSR:  .ASCIZ  'Contents of TSSR Incorrect After Write Characteristics'
1939    035347      126     157     154 TST8ID: .ASCIZ  'Volume Check'
1940                                        .EVEN
1941    035364                          ENDTST
        035364                                                                          L10064:
        035364  104401                                                                  TRAP    C$ETST
1942
1943                                    .SBTTL  TEST  9: COMPLETION INTERRUPT
```

TEST 9: COMPLETION INTERRUPT

```
                                        ;       THIS TEST VERIFIES THAT AN INTERRUPT IS GENERATED AT THE
                                        ;       COMPLETION OF THE WRITE CHARACTERISTICS COMMAND IF THE INTERRUPT
                                        ;       ENABLE (IE) BIT IN THE COMMAND HEADER WORD IS SET.  THIS TEST
                                        ;       CHECKS THE FUNCTIONING OF THE INTERRUPT LOGIC AND BASIC
                                        ;       PROCESSING OF THE IE BIT.
                                        ;
                                        ;       THE SEQUENCES OF TEST 7 ARE REPEATED, EXCEPT THAT THE INTERRUPT
                                        ;       SERVICE ROUTINE IS SET UP TO EXPECT INTERRUPTS AND EACH WRITE
                                        ;       CHARACTERISTICS COMMAND IS ISSUED WITH THE IE BIT SET (1).  IT
                                        ;       IS VERIFIED, WHERE APPROPRIATE, THAT THE IE STATUS BIT IN XST0
                                        ;       OF ANY MESSAGE PACKET IS SET AND THAT A COMPLETION INTERRUPT IS
                                        ;       GENERATED.  FINALLY, A SEQUENCE OF TWO COMMANDS ARE ISSUED, THE
                                        ;       FIRST WITH IE=1 AND THE SECOND WITH IE=0.  IT IS VERIFIED THAT
                                        ;       NO INTERRUPT IS GENERATED AFTER THE SECOND COMMAND AND THAT THE
                                        ;       IE BIT IN XST0 IS 0.
                                        ;
035366                                  BGNTST
035366                                                                  T9::
035366  005037  002224                  CLR     EXTFEA                  ;CLEAR EXTENDED FEATURES SWITCH
035372  012700  040421                  MOV     #TST9ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
035376  004737  016570                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
035402  012737  000024  002214          MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
035410                          T9LOOP:
035410                                  BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
035410                                                                          T9.1:
035410  104402                                                                  TRAP    C$BSUB

035412  004737  040446                  JSR     PC,T9REST               ;SET PACKET TO INITIAL VALUES
035416                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
035416  012700  000000                                                          MOV     #PRI00,R0
035422  104441                                                                  TRAP    C$SPRI
035424  012703  002764                  MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
035430  012704  037350                  MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
035434  012764  000010  000006          MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
035442                          5$:
035442                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
035442  104404                                                                  TRAP    C$BSEG

035444  004737  016054                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
035450  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
035452  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
035454                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
035454  104455                                                                  TRAP    C$ERDF
035456  001605                                                                  .WORD   901
035460  003652                                                                  .WORD   SFIERR
035462  012104                                                                  .WORD   SFIMSG
035464  005037  002220          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
035470  005037  002222                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
035474  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
035500  004737  016416                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
035504  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
035506  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
035510                                  ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
035510  104455                                                                  TRAP    C$ERDF
035512  001606                                                                  .WORD   902
```

TEST 9: COMPLETION INTERRUPT

```
        035514  040067                                                          .WORD   T9SSR
        035516  012116                                                          .WORD   PKTSSR
   1999 035520  005237  002220            INC     FATFLG          ;SET FATAL ERROR FLAG
   2000 035524                     15$:   CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
        035524  104406                                                          TRAP    C$CLP1
   2001 035526                            ESCAPE  SEG             ;BY-PASS SUBTEST IF FATAL ERROR
        035526  104410                                                          TRAP    C$ESCAPE
        035530  000056                                                          .WORD   10000$-.
   2002 035532  005737  002222            TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
   2003 035536  001004                    BNE     22$             ;BRANCH IF YES
   2007 035540                            ERRHRD  ERRNO,T9NINT,PKTSSR
        035540  104456                                                          TRAP    C$ERHRD
        035542  001607                                                          .WORD   903
        035544  040156                                                          .WORD   T9NINT
        035546  012116                                                          .WORD   PKTSSR
   2008 035550  016501  000002      22$:   MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
   2009 035554  012702  000200            MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
   2010 035560  032701  000100            BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
   2011 035564  001402                    BEQ     25$             ;BRANCH IF NOT OFF-LINE
   2012 035566  052702  000100            BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
   2013 035572  020201      25$:   CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
   2014 035574  001404                    BEQ     30$             ;OKAY IF MATCH
   2018 035576                            ERRHRD  ERRNO,T9NBA,PKTSSR      ;NBA NOT ZERO
        035576  104456                                                          TRAP    C$ERHRD
        035600  001610                                                          .WORD   904
        035602  037426                                                          .WORD   T9NBA
        035604  012116                                                          .WORD   PKTSSR
   2019 035606                     30$:
   2020 035606                            ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        035606                                                                  10000$:
        035606  104405                                                          TRAP    C$ESEG
   2021
   2022 035610  012364  000006            MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
   2023 035614  020327  003062            CMP     R3,#TBLEND      ;HAS ALL DATA BEEN TESTED ?
   2024 035620  103002                    BHIS    55$             ;BRANCH IF ALL DATA DONE
   2025 035622  000137  035442            JMP     5$              ;BRANCH TILL BACK TO ZERO
   2026
   2027 035626                     55$:   ENDSUB                  ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
        035626                                                                  L10066:
        035626  104403                                                          TRAP    C$ESUB
   2028
   2029 035630  005737  002220            TST     FATFLG          ;ANY FATAL ERRORS ?
   2030 035634  001402                    BEQ     60$             ;BRANCH IF NOT
   2031 035636  004737  017262            JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
   2032 035642                     60$:
   2033 035642  022737  000001  002226    CMP     #1,REV          ;IS IT A NEW MICROCODE
   2034 035650  001004                    BNE     65$             ;NO BR
   2035 035652  033727  037404  000200    BIT     T98FR+12,#BIT7  ;EXTENDED FEATURES SET?
   2036 035660  001402                    BEQ     70$             ;BR IF NO
   2037 035662  005237  002224      65$:   INC     EXTFEA          ;SET EXT FEATURE FLAG
   2038 035666                     70$:
   2039
   2040                            ;+
   2041                            ;
   2042                            ;TEST 9, SUBTEST 2
   2043                            ;
   2044                            ;CHECK THAT UNUSED BITS BEING SET CAUSES
```

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 ?
```

F12

TEST 9: COMPLETION INTERRUPT

```
2091 036044  001404                        BEQ     30$                     ;OKAY IF MATCH
2095 036046                                ERRHRD  ERRNO,T92REJ,PKTSSR     ;COMMAND NOT REJECTED
     036046  104456                                                            TRAP    C$ERHRD
     036050  001614                                                            .WORD   908
     036052  037501                                                            .WORD   T92REJ
     036054  012116                                                            .WORD   PKTSSR
2096 036056                         30$:
2097 036056                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     036056                                                                    10000$:
     036056  104405                                                            TRAP    C$ESEG
2098
2099 036060  062703  000004                ADD     #4,R3                   ;POINT TO NEXT DATA PAIR
2100 036064  020327  037426                CMP     R3,#T92DONE             ;COMPARE TO END OF TEST DATA
2101 036070  103002                        BHIS    59$                     ;BRANCH IF ALL DATA TESTED
2102 036072  000137  035702                JMP     5$                      ;BRANCH TILL BACK TO ZERO
2103
2104 036076                         59$:   ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     036076                                                                    L10067:
     036076  104403                                                            TRAP    C$ESUB
2105
2106                                ;+
2107                                ;
2108                                ;TEST 9, SUBTEST 3
2109                                ;
2110                                ;CHECK THE WRITE CHARACTERISTICS COMMAND IS REJECTED
2111                                ;IF ISSUED WITH AN INVALID DATA BLOCK BYTE COUNT
2112                                ;
2113                                ;-
2114
2115 036100                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     036100                                                                    T9.3:
     036100  104402                                                            TRAP    C$BSUB
2116
2117 036102                                SETPRI  #PRI00                  ;LOWER PRICRITY TO ALLOW INTERRUPTS
     036102  012700  000000                                                    MOV     #PRI00,R0
     036106  104441                                                            TRAP    C$SPRI
2118 036110  012703  000001                MOV     #1,R3                   ;STARTING BYTE COUNT
2119 036114  012704  037350         5$:    MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
2120 036120  004737  040446                JSR     PC,T9REST               ;RESTORE PACKET TO STARTING VALUES
2121
2122 036124                                BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     036124  104404                                                            TRAP    C$BSEG
2123
2124 036126  004737  016054                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2125 036132  103405                        BCS     10$                     ;BR IF SOFT INIT = OK
2129 036134  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2130 036136                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     036136  104455                                                            TRAP    C$ERDF
     036140  001615                                                            .WORD   909
     036142  003652                                                            .WORD   SFIERR
     036144  012104                                                            .WORD   SFIMSG
2131 036146  005037  002222         10$:   CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2132 036152  010364  000006                MOV     R3,PKBCNT(R4)           ;INSERT THE BYTE COUNT FOR TEST
2133 036156  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2134 036162  004737  016330                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
2135 036166  103405                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2136 036170  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

TEST 9: COMPLETION INTERRUPT

```
2140 036172                          ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     036172  104455                                                          TRAP    C$ERDF
     036174  001616                                                          .WORD   910
     036176  040067                                                          .WORD   T9SSR
     036200  012116                                                          .WORD   PKTSSR
2141 036202                    15$:  CKLOOP                                ;LOOP ON ERROR, IF FLAG SET
     036202  104406                                                          TRAP    C$CLP1
2142 036204                          ESCAPE  SEG                           ;BY-PASS SUBTEST IF FATAL ERROR
     036204  104410                                                          TRAP    C$ESCAPE
     036206  000056                                                          .WORD   10000$-.
2143 036210  005737  002222          TST     INTRECV                       ;DID AN INTERRUPT OCCUR ?
2144 036214  001004                  BNE     22$                           ;BRANCH IF YES
2148 036216                          ERRHRD  ERRNO,T9NINT,PKTSSR
     036216  104456                                                          TRAP    C$ERHRD
     036220  001617                                                          .WORD   911
     036222  040156                                                          .WORD   T9NINT
     036224  012116                                                          .WORD   PKTSSR
2149 036226  016501  000002    22$:  MOV     TSSR(R5),R1                   ;GET THE CONTENTS OF TSSR
2150 036232  012702  102206          MOV     #SC!SSR!TSREJ!NBA,R2          ;EXPECTED CONTENTS OF TSSR
2151 036236  032701  000100          BIT     #OFL,R1                       ;IS OFF-LINE BIT SET ?
2152 036242  001402                  BEQ     25$                           ;BRANCH IF NOT OFF-LINE
2153 036244  052702  000100          BIS     #OFL,R2                       ;SET OFF-LINE IN EXPECTED DATA
2154 036250  020201    25$:  CMP     R2,R1                         ;DOES EXPECTED MATCH RECEIVED ?
2155 036252  001404                  BEQ     30$                           ;OKAY IF MATCH
2159 036254                          ERRHRD  ERRNO,T93REJ,PKTSSR           ;COMMAND NOT REJECTED
     036254  104456                                                          TRAP    C$ERHRD
     036256  001620                                                          .WORD   912
     036260  037600                                                          .WORD   T93REJ
     036262  012116                                                          .WORD   PKTSSR
2160 036264                    30$:
2161 036264                          ENDSEG                                ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     036264                                                                 10000$:
     036264  104405                                                          TRAP    C$ESEG
2162
2163 036266  005203                  INC     R3                            ;NEXT BYTE COUNT
2164 036270  020327  000006          CMP     R3,#6                         ;TESTED ALL INVALID ?
2165 036274  002002                  BGE     59$                           ;BRANCH IF TEST DONE
2166 036276  000137  036114          JMP     5$                            ;BRANCH TILL BACK TO ZERO
2167
2168 036302                    59$:  ENDSUB                                ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     036302                                                                 L10070:
     036302  104403                                                          TRAP    C$ESUB
2169
2170                           ;*
2171                           ;
2172                           ;TEST 9, SUBTEST 4
2173                           ;
2174                           ;SUBTEST TO VERIFY THAT A WRITE CHARACTERISTICS COMMAND IS
2175                           ;REJECTED IF AN ILLEGAL DATA BLOCK ADDRESS IS ISSUED.
2176                           ;
2177                           ;-
2178
2179 036304                          BGNSUB                                ;////////////// BEGIN SUBTEST //////////////
     036304                                                                 T9.4:
     036304  104402                                                          TRAP    C$BSUB
2180
2181 036306                          SETPRI  #PRI00                        ;LOWER PRIORITY TO ALLOW INTERRUPTS
```

TEST 9: COMPLETION INTERRUPT

```
        036306  012700  000000                                                  MOV     #PRI00,R0
        036312  104441                                                          TRAP    C$SPRI
  2182  036314  012703  037412            MOV     #T92DATA,R3         ;START OF TEST DATA FOR SUBTEST
  2183  036320  012704  037350      5$:   MOV     #T9PACKET,R4        ;GET THE ADDRESS OF COMMAND PACKET
  2184  036324  004737  040446            JSR     PC,T9REST           ;RESTORE PACKET TO STARTING VALUES
  2185
  2186
  2187  036330  004737  016054            JSR     PC,SOFINIT          ;DO SOFT INIT OF CONTROLLER
  2188  036334  103405                    BCS     10$                 ;BR IF SOFT INIT = OK
  2192  036336  010001                    MOV     R0,R1               ;SAVE CONTENTS OF TSSR
  2193  036340                            ERRDF   ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
        036340  104455                                                          TRAP    C$ERDF
        036342  001621                                                          .WORD   913
        036344  003652                                                          .WORD   SFIERR
        036346  012104                                                          .WORD   SFIMSG
  2194  036350  005037  002222      10$:  CLR     INTRECV             ;CLEAR INTERRUPT RECEIVED FLAG
  2195  036354  052737  000001  037360  BIS     #1,T9DATA           ;MAKE ADDRESS ODD
  2196  036362  010465  000000          MOV     R4,TSDB(R5)         ;SET THE PACKET ADDRESS
  2197  036366  004737  016330          JSR     PC,WAITF            ;WAIT FOR SSR TO SET
  2198  036372  103405                  BCS     15$                 ;BR IF CARRY SET (GOOD RETURN)
  2199  036374  010001                  MOV     R0,R1               ;SAVE CONTENTS OF TSSR
  2203  036376                          ERRDF   ERRNO,T9SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
        036376  104455                                                          TRAP    C$ERDF
        036400  001622                                                          .WORD   914
        036402  040067                                                          .WORD   T9SSR
        036404  012116                                                          .WORD   PKTSSR
  2204  036406                    15$:  CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
        036406  104406                                                          TRAP    C$CLP1
  2205  036410                          ESCAPE  SUB                 ;BY-PASS SUBTEST IF FATAL ERROR
        036410  104410                                                          TRAP    C$ESCAPE
        036412  000056                                                          .WORD   L10071-.
  2206  036414  005737  002222          TST     INTRECV             ;DID AN INTERRUPT OCCUR ?
  2207  036420  001004                  BNE     22$                 ;BRANCH IF YES
  2211  036422                          ERRHRD  ERRNO,T9NINT,PKTSSR
        036422  104456                                                          TRAP    C$ERHRD
        036424  001623                                                          .WORD   915
        036426  040156                                                          .WORD   T9NINT
        036430  012116                                                          .WORD   PKTSSR
  2212  036432  016501  000002      22$:  MOV     TSSR(R5),R1         ;GET THE CONTENTS OF TSSR
  2213  036436  012702  102206          MOV     #SC!SSR!TSREJ!NBA,R2  ;EXPECTED CONTENTS OF TSSR
  2214  036442  032701  000100          BIT     #OFL,R1             ;IS OFF-LINE BIT SET ?
  2215  036446  001402                  BEQ     25$                 ;BRANCH IF NOT OFF-LINE
  2216  036450  052702  000100          BIS     #OFL,R2             ;SET OFF-LINE IN EXPECTED DATA
  2217  036454  020201            25$:  CMP     R2,R1               ;DOES EXPECTED MATCH RECEIVED ?
  2218  036456  001404                  BEQ     30$                 ;OKAY IF MATCH
  2222  036460                          ERRHRD  ERRNO,T94REJ,PKTSSR ;COMMAND NOT REJECTED
        036460  104456                                                          TRAP    C$ERHRD
        036462  001624                                                          .WORD   916
        036464  037673                                                          .WORD   T94REJ
        036466  012116                                                          .WORD   PKTSSR
  2223  036470                    30$:
  2224
  2225  036470                          ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        036470                                                                  L10071:
        036470  104403                                                          TRAP    C$ESUB
  2226
  2227
```

TEST 9: COMPLETION INTERRUPT

```
                                        ;*
                                        ;
                                        ;TEST 9, SUBTEST 5
                                        ;
                                        ;CHECK THAT WRITE CHARACTERISTICS COMMAND IS REJECTED IF THE
                                        ;MESSAGE BUFFER LENGTH SPECIFIES AN INVALID COUNT (LESS THAN 14)
                                        ;
                                        ;-

                                                BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
                                                                                                        T9.5:
036472  104402                                                                                                TRAP    C$BSUB

                                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
036474  012700  000000                                                                                        MOV     #PRI00,R0
036500  104441                                                                                                TRAP    C$SPRI
036502  012703  000001                        MOV     #1,R3                   ;STARTING BUFFER LENGTH
036506  012704  037350                5$:     MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
036512  004737  040446                        JSR     PC,T9REST               ;RESTORE PACKET TO STARTING VALUES

036516                                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
036516  104404                                                                                                TRAP    C$BSEG

036520  004737  016054                        JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
036524  103405                                BCS     10$                     ;BR IF SOFT INIT = OK
036526  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
036530                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
036530  104455                                                                                                TRAP    C$ERDF
036532  001625                                                                                                .WORD   917
036534  003652                                                                                                .WORD   SFIERR
036536  012104                                                                                                .WORD   SFIMSG
036540  005037  002222                10$:     CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
036544  010337  037364                        MOV     R3,T9DATA+4             ;INSERT THE BAD MESSAGE LENGTH
036550  010465  000000                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
036554  004737  016330                        JSR     PC,WAITF                ;WAIT FOR SSR TO SET
036560  103405                                BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
036562  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
036564                                          ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
036564  104455                                                                                                TRAP    C$ERDF
036566  001626                                                                                                .WORD   918
036570  040067                                                                                                .WORD   T9SSR
036572  012116                                                                                                .WORD   PKTSSR
036574                                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
036574  104406                                                                                                TRAP    C$CLP1
036576                                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
036576  104410                                                                                                TRAP    C$ESCAPE
036600  000056                                                                                                .WORD   10000$-.
036602  005737  002222                        TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
036606  001004                                BNE     22$                     ;BRANCH IF YES
036610                                          ERRHRD  ERRNO,T9NINT,PKTSSR
036610  104456                                                                                                TRAP    C$ERHRD
036612  001627                                                                                                .WORD   919
036614  040156                                                                                                .WORD   T9NINT
036616  012116                                                                                                .WORD   PKTSSR
036620  016501  000002                22$:     MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
036624  012702  102206                        MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
036630  032701  000100                        BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
```

```
TEST  9: COMPLETION INTERRUPT

 2274 036634  001402                    BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 2275 036636  052702  000100            BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 2276 036642  020201              25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 2277 036644  001404                    BEQ     30$                     ;OKAY IF MATCH
 2281 036646                            ERRHRD  ERRNO,T95REJ,PKTSSR     ;COMMAND NOT REJECTED
      036646  104456                                                            TRAP    C$ERHRD
      036650  001630                                                            .WORD   920
      036652  037771                                                            .WORD   T95REJ
      036654  012116                                                            .WORD   PKTSSR
 2282 036656                      30$:
 2283 036656                            ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      036656                                                                    10000$:
      036656  104405                                                            TRAP    C$ESEG
 2284
 2285 036660  005203                    INC     R3                      ;NEXT BUFFER LENGTH
 2286 036662  020327  000014            CMP     R3,#14                  ;HAVE ALL BAD VALUES BEEN TESTED ?
 2287 036666  002002                    BGE     59$                     ;BRANCH IF ALL TESTED
 2288 036670  000137  036506            JMP     5$                      ;BRANCH TILL BACK TO ZERO
 2289
 2290 036674                      59$:  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      036674                                                                    L10072:
      036674  104403                                                            TRAP    C$ESUB
 2291                                   ;*
 2292                                   ;
 2293                                   ;TEST 9, SUBTEST 6
 2294                                   ;
 2295                                   ;THIS SUBTEST IS EXECUTED ONLY IF THE EXTENDED
 2296                                   ;FEATURES MODE IS ENABLED (AS DETERMINED BY EXAMINING
 2297                                   ;XST2 AFTER A PREVIOUS EXECUTION OF WRITE CHARACTERISTICS).
 2298                                   ;IT VERIFIES THAT A FIFTH CHARACTERISTICS DATA WORD IS FETCHED
 2299                                   ;IF THE BYTE COUNT PARAMETER IN THE COMMAND PACKET IS 10 DECIMAL
 2300                                   ;OR GREATER.
 2301                                   ;
 2302                                   ;-
 2303 036676                            BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
      036676                                                                    T9.6:
      036676  104402                                                            TRAP    C$BSUB
 2304 036700  005737  002224            TST     EXTFEA                  ;IS EXTENDED FEATURES SOFT. SW SET?
 2305 036704  001002                    BNE     4$                      ;BR, IF SOFTWARE SWITCH IS SET (ON)
 2306 036706  000137  037126            JMP     55$                     ;EXIT SUBTEST
 2307 036712  004737  040446      4$:   JSR     PC,T9REST               ;SET PACKET TO START-UP VALUES
 2308
 2309 036716                            SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      036716  012700  000000                                                    MOV     #PRI00,R0
      036722  104441                                                            TRAP    C$SPRI
 2310 036724  012703  002764            MOV     #T$TSTBLK+10.,R3        ;START OF TEST DATA
 2311 036730  012704  037350            MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
 2312 036734  012764  000012  000006    MOV     #10.,PKBCNT(R4)         ;START WITH EXTENDED FEATURES VALUE
 2313 036742                      5$:
 2314 036742                            BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
      036742  104404                                                            TRAP    C$BSEG
 2315
 2316 036744  004737  016054            JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 2317 036750  103405                    BCS     10$                     ;BR IF SOFT INIT = OK
 2321 036752  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2322 036754                            ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      036754  104455                                                            TRAP    C$ERDF
```

TEST 9: COMPLETION INTERRUPT

```
                036756  001631                                                          .WORD   921
                036760  003652                                                          .WORD   SFIERR
                036762  012104                                                          .WORD   SFIMSG
  2323          036764  005037  002220          10$:    CLR     FATFLG          ;CLEAR FATAL ERROR FLAG
  2324          036770  005037  002222                  CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
  2325          036774  010465  000000                  MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
  2326          037000  004737  016416                  JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
  2327          037004  103407                          BCS     15$             ;BR IF CARRY SET (GOOD RETURN)
  2328          037006  010001                          MOV     R0,R1           ;SAVE CONTENTS OF TSSR
  2332          037010                                  ERRDF   ERRNO,T9SSR,PKTSSR      ;DCVICE FATAL SSR FAILED TO SET
                037010  104455                                                          TRAP    C$ERDF
                037012  001632                                                          .WORD   922
                037014  040067                                                          .WORD   T9SSR
                037016  012116                                                          .WORD   PKTSSR
  2333          037020  005237  002220                  INC     FATFLG          ;SET FATAL ERROR FLAG
  2334          037024                          15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
                037024  104406                                                          TRAP    C$CLP1
  2335          037026                                  ESCAPE  SEG             ;BY-PASS SUBTEST IF FATAL ERROR
                037026  104410                                                          TRAP    C$ESCAPE
                037030  000056                                                          .WORD   10000$-.
  2336          037032  005737  002222                  TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
  2337          037036  001004                          BNE     22$             ;BRANCH IF YES
  2341          037040                                  ERRHRD  ERRNO,T9NINT,PKTSSR
                037040  104456                                                          TRAP    C$ERHRD
                037042  001633                                                          .WORD   923
                037044  040156                                                          .WORD   T9NINT
                037046  012116                                                          .WORD   PKTSSR
  2342          037050  016501  000002          22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
  2343          037054  012702  000200                  MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
  2344          037060  032701  000100                  BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
  2345          037064  001402                          BEQ     25$             ;BRANCH IF NOT OFF-LINE
  2346          037066  052702  000100                  BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
  2347          037072  020201                  25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
  2348          037074  001404                          BEQ     30$             ;OKAY IF MATCH
  2352          037076                                  ERRHRD  ERRNO,T9NBA,PKTSSR      ;NBA NOT ZERO
                037076  104456                                                          TRAP    C$ERHRD
                037100  001634                                                          .WORD   924
                037102  037426                                                          .WORD   T9NBA
                037104  012116                                                          .WORD   PKTSSR
  2353          037106                          30$:
  2354          037106                                  ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                037106                                                                  10000$:
                037106  104405                                                          TRAP    C$ESEG
  2355
  2356          037110  012364  000006                  MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
  2357          037114  020327  003062                  CMP     R3,#TBLEND      ;HAS ALL DATA BEEN TESTED ?
  2358          037120  103002                          BHIS    55$             ;BRANCH IF ALL DATA DONE
  2359          037122  000137  036742                  JMP     5$              ;BRANCH TILL BACK TO ZERO
  2360
  2361          037126                          55$:    ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                037126                                                                  L10073:
                037126  104403                                                          TRAP    C$ESUB
  2362
  2363
  2364                                          ;*
  2365                                          ;
  2366                                          ;TEST 9, SUBTEST 7
```

TEST 9: COMPLETION INTERRUPT

```
2367                                    ;
2368                                    ;TEST WRITE CHARACTERISTICS WITH/WITHOUT INTERRUPTS ENABLED
2369                                    ;
2370                                    ;-
2371
2372 037130                                    BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
      037130                                                                                      T9.7:
      037130  104402                                                                                TRAP    C$BSUB
2373
2374 037132                                    SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      037132  012700  000000                                                                      MOV     #PRI00,R0
      037136  104441                                                                              TRAP    C$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     10$                     ;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$ERDF
      037162  001635                                                                              .WORD   925
      037164  003652                                                                              .WORD   SFIERR
      037166  012104                                                                              .WORD   SFIMSG
2384 037170  005037  002222            10$:    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     15$                     ;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$ERDF
      037216  001636                                                                              .WORD   926
      037220  040067                                                                              .WORD   T9SSR
      037222  012116                                                                              .WORD   PKTSSR
2394 037224                            15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      037224  104406                                                                              TRAP    C$CLP1
2395 037226                                    ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
      037226  104410                                                                              TRAP    C$ESCAPE
      037230  000102                                                                              .WORD   L10074-.
2396 037232  005737  002222                    TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2397 037236  001004                           BNE     22$                     ;BRANCH IF YES
2401 037240                                    ERRHRD  ERRNO,T9NINT,PKTSSR
      037240  104456                                                                              TRAP    C$ERHRD
      037242  001637                                                                              .WORD   927
      037244  040156                                                                              .WORD   T9NINT
      037246  012116                                                                              .WORD   PKTSSR
2402 037250                            22$:    CKLOOP                          ;LOOP ON ERROR ?
      037250  104406                                                                              TRAP    C$CLP1
2403
2404 037252  005037  002222                    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2405 037256  042714  000200                    BIC     #BIT7,(R4)              ;DISABLE INTERRUPTS
2406 037262  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2407 037266  004737  016416                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2408 037272  103405                           BCS     25$                     ;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
```

TEST 9: COMPLETION INTERRUPT

```
        037302  040067                                                          .WORD   T9SSR
        037304  012116                                                          .WORD   PKTSSR
  2414  037306                  25$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        037306  104406                                                          TRAP    C$CLP1
  2415  037310                          ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
        037310  104410                                                          TRAP    C$ESCAPE
        037312  000020                                                          .WORD   L10074-.
  2416  037314  005737  002222        TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
  2417  037320  001404                BEQ     30$                     ;BRANCH IF NOT
  2421  037322                        ERRHRD  ERRNO,T9INT,PKTSSR
        037322  104456                                                          TRAP    C$ERHRD
        037324  001641                                                          .WORD   929
        037326  040247                                                          .WORD   T9INT
        037330  012116                                                          .WORD   PKTSSR
  2422  037332                  30$:
  2423  037332                          ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        037332                                                                  L10074:
        037332  104403                                                          TRAP    C$ESUB
  2424
  2425  037334                          EXIT    TST                     ;ALL DONE THIS TEST
        037334  104432                                                          TRAP    C$EXIT
        037336  001162                                                          .WORD   L10065-.
  2426
  2427                          ;+
  2428                          ;LOCAL STORAGE FOR THIS TEST
  2429                          ;-
  2430
  2432  037340                          .BLKB   10-<.-TSV2&7>
  2434  037350                  T9PACKET:                               ;COMMAND PACKET FOR TEST
  2435  037350  100204                  .WORD   100204                  ;WRITE CHAR COMMAND, WITH IE, ACK
  2436  037352  037360                  .WORD   T9DATA                  ;ADDRESS OF CHARACTERISTICS BLOCK
  2437  037354  000000                  .WORD   0                       ;
  2438  037356  000010                  .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
  2439
  2440  037360                  T9DATA:                                 ;CHARACTERISTICS DATA BLOCK
  2441  037360  037372                  .WORD   T9BFR                   ;ADDRESS OF MESSAGE BUFFER
  2442  037362  000000                  .WORD   0                       ;
  2443  037364  000016                  .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
  2444  037366  000000  000000          .WORD   0,0                     ;
  2445
  2446  037372                  T9BFR:  .BLKW   8.                      ;MESSAGE BUFFER
  2447
  2448                          ;+
  2449                          ;
  2450                          ;TEST DATA FOR SUBTEST TWO
  2451                          ;
  2452                          ;DATA HAS FORMAT:
  2453                          ;
  2454                          ;       1ST WORD        OFFSET TO TEST WORD IN PACKET
  2455                          ;       2ND WORD        BITS TO SET FOR TEST
  2456                          ;
  2457                          ;-
  2458
  2459  037412                  T92DATA:
  2460  037412  000060  037140          .WORD   0,BIT5!BIT6!BIT9!BIT10!BIT11!BIT12!BIT13
  2461  037416  000002  000001          .WORD   2,BIT0
  2462  037422  000004  100100          .WORD   4,BIT6!BIT15
```

N12

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                                    ;
2485                                    ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
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
     040520                                                                  L10065:
     040520  104401                                                                TRAP    C$ETST
2504
2505                                            .SBTTL  TEST  10: BASIC PACKET PROTOCOL
2506
2507                                    ;       THIS TEST VERIFIES BASIC OPERATION OF THE MESSAGE BUFFER RELEASE
2508                                    ;       COMMAND, THE FUNCTION OF THE ACK BIT IN THE COMMAND HEADER WORD,
2509                                    ;       AND THE REGISTER MODIFICATION REFUSED (RMR) LOGIC.
2510                                    ;
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
```

TEST 10: BASIC PACKET PROTOCOL

```
                                        ;
                                        ;-
                        BGNTST                                          T10::
                                                                ;ASCII MESSAGE TO IDENTIFY TEST
                        MOV     #TST10ID,R0
                        JSR     PC,TSTSETUP                     ;DO INITIAL TEST SETUP
                        MOV     #20.,LOOPCNT                    ;PERFORM 20 ITERATIONS
                T10LOOP:
                        BGNSUB                                  ;////////////// BEGIN SUBTEST ////////////
                                                                        T10.1:
                                                                                TRAP    C$BSUB
                        JSR     PC,T10RST                       ;SET PACKET TO INITIAL VALUES
                        SETPRI  #PRI00                          ;LOWER PRIORITY TO ALLOW INTERRUPTS
                                                                                MOV     #PRI00,R0
                                                                                TRAP    C$SPRI
                        MOV     #T10PACKET,R4                   ;GET THE ADDRESS OF COMMAND PACKET
                        MOV     #8.,PKBCNT(R4)                  ;START WITH MINIMUM ALLOWABLE VALUE
                5$:
                        BGNSEG                                  ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
                                                                                TRAP    C$BSEG
                        JSR     PC,SOFINIT                      ;DO SOFT INIT OF CONTROLLER
                        BCS     10$                             ;BR IF SOFT INIT = OK
                        MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
                        ERRDF   ERRNO,SFIERR,SFIMSG             ;DEVICE FATAL ERROR DURING INIT
                                                                                TRAP    C$ERDF
                                                                                .WORD   1001
                                                                                .WORD   SFIERR
                                                                                .WORD   SFIMSG
                10$:    CLR     FATFLG                          ;CLEAR FATAL ERROR FLAG
                        CLR     INTRECV                         ;CLEAR INTERRUPT RECEIVED FLAG
                        MOV     R4,TSDB(R5)                     ;SET THE PACKET ADDRESS
                        JSR     PC,CHKTSSR                      ;WAIT FOR SSR TO SET
                        BCS     15$                             ;BR IF CARRY SET (GOOD RETURN)
                        MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
                        ERRDF   ERRNO,T10SSR,PKTSSR             ;DEVICE FATAL SSR FAILED TO SET
                                                                                TRAP    C$ERDF
                                                                                .WORD   1002
                                                                                .WORD   T10SSR
                                                                                .WORD   PKTSSR
                        INC     FATFLG                          ;SET FATAL ERROR FLAG
                15$:    CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
                                                                                TRAP    C$CLP1
                        ESCAPE  SEG                             ;BY-PASS SUBTEST IF FATAL ERROR
                                                                                TRAP    C$ESCAPE
                                                                                .WORD   10000$-.
                        TST     INTRECV                         ;DID AN INTERRUPT OCCUR ?
                        BNE     22$                             ;BRANCH IF YES
                        ERRHRD  ERRNO,T10NINT,PKTSSR
                                                                                TRAP    C$ERHRD
                                                                                .WORD   1003
                                                                                .WORD   T10NINT
                                                                                .WORD   PKTSSR
                22$:    MOV     TSSR(R5),R1                     ;GET THE CONTENTS OF TSSR
                        MOV     #SSR,R2                         ;EXPECTED CONTENTS OF TSSR
```

TEST 10: BASIC PACKET PROTOCOL

```
 2567 040704  032701  000100            BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 2568 040710  001402                    BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 2569 040712  052702  000100            BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 2570 040716  020201              25$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 2571 040720  001404                    BEQ     30$                     ;OKAY IF MATCH
 2575 040722                            ERRHRD  ERRNO,T10NBA,PKTSSR     ;NBA NOT ZERO
      040722  104456                                                            TRAP    C$ERHRD
      040724  001754                                                            .WORD   1004
      040726  043051                                                            .WORD   T10NBA
      040730  012116                                                            .WORD   PKTSSR
 2576 040732                      30$:
 2577 040732                            ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      040732                                                                10000$:
      040732  104405                                                            TRAP    C$ESEG
 2578 040734                            BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
      040734  104404                                                            TRAP    C$BSEG
 2579
 2580 040736  005037  002222            CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 2581 040742  012737  025252  042672    MOV     #025252,T10BFR          ;WIPE OUT MESSAGE BUFFER AREA
 2582 040750  012714  100212            MOV     #100212,(R4)            ;SET COMMAND PACKET TO MESS BUF REL
 2583 040754  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 2584 040760  004737  016416            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 2585 040764  103407                    BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
 2586 040766  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2590 040770                            ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      040770  104455                                                            TRAP    C$ERDF
      040772  001755                                                            .WORD   1005
      040774  043210                                                            .WORD   T10SSR
      040776  012116                                                            .WORD   PKTSSR
 2591 041000  005237  002220            INC     FATFLG                  ;SET FATAL ERROR FLAG
 2592 041004                      45$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      041004  104406                                                            TRAP    C$CLP1
 2593 041006  005737  002222            TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 2594 041012  001404                    BEQ     52$                     ;BRANCH IF NO
 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             ;GET THE CONTENTS OF TSSR
 2600 041030  012702  000200            MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
 2601 041034  032701  000100            BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 2602 041040  001402                    BEQ     55$                     ;BRANCH IF NOT OFF-LINE
 2603 041042  052702  000100            BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 2604 041046  020201              55$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 2605 041050  001404                    BEQ     60$                     ;OKAY IF MATCH
 2609 041052                            ERRHRD  ERRNO,T10NNBA,PKTSSR    ;NBA  SET
      041052  104456                                                            TRAP    C$ERHRD
      041054  001757                                                            .WORD   1007
      041056  043133                                                            .WORD   T10NNBA
      041060  012116                                                            .WORD   PKTSSR
 2610 041062                      60$:
 2611 041062  013701  042672            MOV     T10BFR,R1               ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
 2612 041066  012702  025252            MOV     #025252,R2              ;SET UP EXPECTED DATA
 2613 041072  020102                    CMP     R1,R2                   ;WAS ANY MESSAGE REC'D
 2614 041074  001404                    BEQ     70$                     ;BR, IF OK (EQUAL)
 2618 041076                            ERRHRD  ERRNO,T10MBF,EXPREC     ;MESSAGE BUFFER WAS MODIFIED
```

```
TEST  10: BASIC PACKET PROTOCOL

        041076  104456                                                          TRAP    C$ERHRD
        041100  001760                                                          .WORD   1008
        041102  042754                                                          .WORD   T10MBF
        041104  015554                                                          .WORD   EXPREC
 2619   041106                  70$:
 2620   041106  005737  002220          TST     FATFLG                  ;ANY FATAL ERRORS
 2621   041112  001403                  BEQ     80$                     ;BR, IF NO FATAL ERRORS
 2622   041114  004737  017262          JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 2623   041120                          ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        041120                                                                  10001$:
        041120  104405                                                          TRAP    C$ESEG
 2624   041122                  80$:
 2625   041122                          ENDSUB                          ;////////////// END SUBTEST ////////////
        041122                                                                  L10076:
        041122  104403                                                          TRAP    C$ESUB
 2626
 2627                           ;+
 2628                           ;TEST 10 SUBTEST 2
 2629                           ;
 2630                           ;CHECKS THAT THE MESSAGE BUFFER RELEASE COMMAND WORKS
 2631                           ;PROPERLY AND THAT THERE IS AN INTERRUPT IF THE "IE"
 2632                           ;BIT IS SET IN THE COMMAND PACKET AND THE "ERI" BIT
 2633                           ;IS SET IN THE CHARACTERISTICS DATA PACKET
 2634                           ;
 2635                           ;-
 2636   041124                          BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
        041124                                                                  T10.2:
        041124  104402                                                          TRAP    C$BSUB
 2637
 2638   041126  004737  043506          JSR     PC,T10RST               ;SET PACKET TO INITIAL VALUES
 2639   041132                          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
        041132  012700  000000                                                  MOV     #PRI00,R0
        041136  104441                                                          TRAP    C$SPRI
 2640   041140  012704  042650          MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 2641   041144  012764  000010  000006  MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
 2642   041152                  5$:
 2643   041152                          BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
        041152  104404                                                          TRAP    C$BSEG
 2644
 2645   041154  004737  016054          JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 2646   041160  103405                  BCS     10$                     ;BR IF SOFT INIT = OK
 2650   041162  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2651   041164                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
        041164  104455                                                          TRAP    C$ERDF
        041166  001761                                                          .WORD   1009
        041170  003652                                                          .WORD   SFIERR
        041172  012104                                                          .WORD   SFIMSG
 2652   041174  005037  002220  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 2653   041200  005037  002222          CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 2654   041204  012737  000020  042666  MOV     #000020,T10DATA+6       ;SET ERI IN CHARACTERISTICS DATA
 2655   041212  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 2656   041216  004737  016416          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 2657   041222  103407                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 2658   041224  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2662   041226                          ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        041226  104455                                                          TRAP    C$ERDF
        041230  001762                                                          .WORD   1010
```

TEST 10: BASIC PACKET PROTOCOL

```
                041232  043210                                                          .WORD   T10SSR
                041234  012116                                                          .WORD   PKTSSR
  2663          041236  005237  002220          INC     FATFLG                  ;SET FATAL ERROR FLAG
  2664          041242                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                041242  104406                                                          TRAP    C$CLP1
  2665          041244                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
                041244  104410                                                          TRAP    C$ESCAPE
                041246  000056                                                          .WORD   10000$-.
  2666          041250  005737  002222          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
  2667          041254  001004                  BNE     22$                     ;BRANCH IF YES
  2671          041256                          ERRHRD  ERRNO,T10NINT,PKTSSR
                041256  104456                                                          TRAP    C$ERHRD
                041260  001763                                                          .WORD   1011
                041262  043277                                                          .WORD   T10NINT
                041264  012116                                                          .WORD   PKTSSR
  2672          041266  016501  000002  22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
  2673          041272  012702  000200          MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
  2674          041276  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
  2675          041302  001402                  BEQ     25$                     ;BRANCH IF NOT OFF-LINE
  2676          041304  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
  2677          041310  020201  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
  2678          041312  001404                  BEQ     30$                     ;OKAY IF MATCH
  2682          041314                          ERRHRD  ERRNO,T10NBA,PKTSSR     ;NBA NOT ZERO
                041314  104456                                                          TRAP    C$ERHRD
                041316  001764                                                          .WORD   1012
                041320  043051                                                          .WORD   T10NBA
                041322  012116                                                          .WORD   PKTSSR
  2683          041324                  30$:
  2684          041324                          ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                041324                                                                  10000$:
                041324  104405                                                          TRAP    C$ESEG
  2685          041326                          BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
                041326  104404                                                          TRAP    C$BSEG
  2686
  2687          041330  005037  002222          CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
  2688          041334  012737  025252  042672  MOV     #025252,T10BFR          ;WIPE JUT MESSAGE BUFFER AREA
  2689          041342  012714  100212          MOV     #100212,(R4)            ;SET COMMAND PACKET TO MESS BUF REL
  2690          041346  010465  000000          MOV     R4,TSD8(R5)             ;SET THE PACKET ADDRESS
  2691          041352  004737  016416          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
  2692          041356  103407                  BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
  2693          041360  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  2697          041362                          ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                041362  104455                                                          TRAP    C$ERDF
                041364  001765                                                          .WORD   1013
                041366  043210                                                          .WORD   T10SSR
                041370  012116                                                          .WORD   PKTSSR
  2698          041372  005237  002220          INC     FATFLG                  ;SET FATAL ERROR FLAG
  2699          041376                  45$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                041376  104406                                                          TRAP    C$CLP1
  2700          041400  005737  002222          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
  2701          041404  001004                  BNE     52$                     ;BRANCH IF YES
  2705          041406                          ERRHRD  ERRNO,T10INT,PKTSSR
                041406  104456                                                          TRAP    C$ERHRD
                041410  001766                                                          .WORD   1014
                041412  043370                                                          .WORD   T10INT
                041414  012116                                                          .WORD   PKTSSR
  2706          041416  016501  000002  52$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
```

F13

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                 55$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2712 041442  001404                        BEQ     60$                     ;OKAY IF MATCH
2716 041444                                ERRHRD  ERRNO,T10NNBA,PKTSSR    ;NBA NOT SET
     041444  104456                                                                TRAP    C$ERHRD
     041446  001767                                                                .WORD   1015
     041450  043133                                                                .WORD   T10NNBA
     041452  012116                                                                .WORD   PKTSSR
2717 041454                         60$:
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$ERHRD
     041472  001770                                                                .WORD   1016
     041474  042754                                                                .WORD   T10MBF
     041476  015554                                                                .WORD   EXPREC
2726
2727 041500                         70$:
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                         80$:
2732 041512                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     041512                                                                     10001$:
     041512  104405                                                                TRAP    C$ESEG
2733 041514                                ENDSUB                          ;////////////// END SUBTEST ////////////
     041514                                                                     L10077:
     041514  104403                                                                TRAP    C$ESUB
2734
2735                                ;+
2736                                ;TEST 10 SUBTEST 3
2737                                ;
2738                                ;CHECKS THAT THE CPU GIVES UP OWNERSHIP OF THE MESSAGE BUFFER
2739                                ;AFTER THE MESSAGE BUFFER RELEASE, AND THAT FOLLOWING COMMANDS
2740                                ;WORK CORRECTLY
2741                                ;
2742                                ;-
2743 041516                                BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
     041516                                                                     T10.3:
     041516  104402                                                                TRAP    C$BSUB
2744
2745 041520  004737  043506                JSR     PC,T10RST               ;SET PACKET TO INITIAL VALUES
2746 041524                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     041524  012700  000000                                                        MOV     #PRI00,R0
     041530  104441                                                                TRAP    C$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                         5$:
2750 041544                                BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     041544  104404                                                                TRAP    C$BSEG
2751
2752 041546  004737  016054                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
```

TEST 10: BASIC PACKET PROTOCOL

```
 2753 041552  103405                    BCS     10$                     ;BR IF SOFT INIT = OK
 2757 041554  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2758 041556                            ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      041556  104455                                                            TRAP    C$ERDF
      041560  001771                                                            .WORD   1017
      041562  003652                                                            .WORD   SFIERR
      041564  012104                                                            .WORD   SFIMSG
 2759 041566  005037  002220      10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 2760 041572  005037  002222              CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 2761 041576  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 2762 041602  004737  016416              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 2763 041606  103407                      BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 2764 041610  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2768 041612                              ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      041612  104455                                                            TRAP    C$ERDF
      041614  001772                                                            .WORD   1018
      041616  043210                                                            .WORD   T10SSR
      041620  012116                                                            .WORD   PKTSSR
 2769 041622  005237  002220              INC     FATFLG                  ;SET FATAL ERROR FLAG
 2770 041626                      15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      041626  104406                                                            TRAP    C$CLP1
 2771 041630                              ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
      041630  104410                                                            TRAP    C$ESCAPE
      041632  000056                                                            .WORD   10000$-.
 2772 041634  005737  002222              TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 2773 041640  001004                      BNE     22$                     ;BRANCH IF YES
 2777 041642                              ERRHRD  ERRNO,T10NINT,PKTSSR
      041642  104456                                                            TRAP    C$ERHRD
      041644  001773                                                            .WORD   1019
      041646  043277                                                            .WORD   T10NINT
      041650  012116                                                            .WORD   PKTSSR
 2778 041652  016501  000002      22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 2779 041656  012702  000200              MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
 2780 041662  032701  000100              BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 2781 041666  001402                      BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 2782 041670  052702  000100              BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 2783 041674  020201      25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 2784 041676  001404                      BEQ     30$                     ;OKAY IF MATCH
 2788 041700                              ERRHRD  ERRNO,T10NBA,PKTSSR     ;NBA NOT ZERO
      041700  104456                                                            TRAP    C$ERHRD
      041702  001774                                                            .WORD   1020
      041704  043051                                                            .WORD   T10NBA
      041706  012116                                                            .WORD   PKTSSR
 2789 041710              30$:
 2790 041710                              ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      041710                                                                10000$:
      041710  104405                                                            TRAP    C$ESEG
 2791 041712                              BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
      041712  104404                                                            TRAP    C$BSEG
 2792 041714  004737  043506              JSR     PC,T10RST               ;RESET PACKETS AND COMMANDS
 2793 041720  005037  002222              CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 2794 041724  012737  025252  042672      MOV     #025252,T10BFR          ;WIPE OUT MESSAGE BUFFER AREA
 2795 041732  012714  100212              MOV     #100212,(R4)            ;SET COMMAND PACKET TO MESS BUF REL
 2796 041736  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 2797 041742  004737  016416              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 2798 041746  103407                      BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
 2799 041750  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

TEST 10: BASIC PACKET PROTOCOL

```
2803 041752                            ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     041752  104455                                                            TRAP    C$ERDF
     041754  001775                                                            .WORD   1021
     041756  043210                                                            .WORD   T10SSR
     041760  012116                                                            .WORD   PKTSSR
2804 041762  005237  002220            INC     FATFLG                  ;SET FATAL ERROR FLAG
2805 041766                    45$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041766  104406                                                            TRAP    C$CLP1
2806 041770  005737  002222            TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2807 041774  001404                    BEQ     52$                     ;BRANCH IF NO
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             ;GET THE CONTENTS OF TSSR
2813 042012  012702  000200            MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
2814 042016  032701  000100            BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
2815 042022  001402                    BEQ     55$                     ;BRANCH IF NOT OFF-LINE
2816 042024  052702  000100            BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2817 042030  020201    55$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2818 042032  001404                    BEQ     60$                     ;OKAY IF MATCH
2822 042034                            ERRHRD  ERRNO,T10NNBA,PKTSSR    ;NBA NOT SET
     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               ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
2825 042050  012702  025252            MOV     #025252,R2              ;SET UP EXPECTED DATA
2826 042054  020102                    CMP     R1,R2                   ;WAS ANY MESSAGE REC'D
2827 042056  001404                    BEQ     70$                     ;BR, IF OK (EQUAL)
2831 042060                            ERRHRD  ERRNO,T10MBF,EXPREC     ;MESSAGE BUFFER WAS MODIFIED
     042060  104456                                                            TRAP    C$ERHRD
     042062  002000                                                            .WORD   1024
     042064  042754                                                            .WORD   T10MBF
     042066  015554                                                            .WORD   EXPREC
2832
2833 042070                    70$:    CKLOOP                          ;LOOP ON ERROR IF FLAG SET
     042070  104406                                                            TRAP    C$CLP1
2834
2835 042072  005037  002222            CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2836 042076  004737  043506            JSR     PC,T10RST               ;RESET THE PACKETS AND COMMANDS
2837 042102  042714  100000            BIC     #100000,(R4)            ;CLEAR THE ACK BIT
2838 042106  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2839 042112  004737  016416            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2840 042116  103407                    BCS     75$                     ;BR IF CARRY SET (GOOD RETURN)
2841 042120  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2845 042122                            ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     042122  104455                                                            TRAP    C$ERDF
     042124  002001                                                            .WORD   1025
     042126  043210                                                            .WORD   T10SSR
     042130  012116                                                            .WORD   PKTSSR
2846 042132  005237  002220            INC     FATFLG                  ;SET FATAL ERROR FLAG
2847 042136                    75$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     042136  104406                                                            TRAP    C$CLP1
2848 042140                            ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
```

TEST 10: BASIC PACKET PROTOCOL

```
                042140  104410                                                          TRAP    C$ESCAPE
                042142  000062                                                          .WORD   10001$ .
2849            042144  005737  002222          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2850            042150  001006                  BNE     82$                     ;BRANCH IF YES
2854            042152  016500  000002          MOV     TSSR(R5),R0             ;GET TSSR FOR ERROR REPORT
2855            042156                          ERRHRD  ERRNO,T10NINT,PKTSSR
                042156  104456                                                          TRAP    C$ERHRD
                042160  002002                                                          .WORD   1026
                042162  043277                                                          .WORD   T10NINT
                042164  012116                                                          .WORD   PKTSSR
2856            042166  016501  000002  82$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
2857            042172  012702  000200          MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
2858            042176  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
2859            042202  001402                  BEQ     85$                     ;BRANCH IF NOT OFF-LINE
2860            042204  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2861            042210  020201  85$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2862            042212  001404                  BEQ     90$                     ;OKAY IF MATCH
2866            042214                          ERRHRD  ERRNO,T10SSR,PKTSSR     ;NBA NOT ZERO
                042214  104456                                                          TRAP    C$ERHRD
                042216  002003                                                          .WORD   1027
                042220  043210                                                          .WORD   T10SSR
                042222  012116                                                          .WORD   PKTSSR
2867            042224                  90$:
2868            042224                          ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                042224                                                                  10001$:
                042224  104405                                                          TRAP    C$ESEG
2869            042226  005737  002220          TST     FATFLG                  ;ANY FATAL ERRORS
2870            042232  001403                  BEQ     95$                     ;BR, IF NO FATAL ERRORS
2871            042234  004737  017262          JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2872
2873            042240                          BGNSEG                          ;<<<<<<<<<<<<< BGN SEGMENT <<<<<<<<<<<<<
                042240  104404                                                          TRAP    C$BSEG
2874            042242  005037  002222  95$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2875            042246  004737  043506          JSR     PC,T10RST               ;RESET THE PACKETS AND COMMANDS
2876            042252  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2877            042256  004737  016416          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2878            042262  103407                  BCS     100$                    ;BR IF CARRY SET (GOOD RETURN)
2879            042264  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2883            042266                          ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                042266  104455                                                          TRAP    C$ERDF
                042270  002004                                                          .WORD   1028
                042272  043210                                                          .WORD   T10SSR
                042274  012116                                                          .WORD   PKTSSR
2884            042276  005237  002220          INC     FATFLG                  ;SET FATAL ERROR FLAG
2885            042302  100$:   CKLOOP                                          ;LOOP ON ERROR, IF FLAG SET
                042302  104406                                                          TRAP    C$CLP1
2886            042304                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
                042304  104410                                                          TRAP    C$ESCAPE
                042306  000062                                                          .WORD   10002$-.
2887            042310  005737  002222          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2888            042314  001006                  BNE     112$                    ;BRANCH IF YES
2892            042316  016500  000002          MOV     TSSR(R5),R0             ;GET TSSR FOR ERROR REPORT
2893            042322                          ERRHRD  ERRNO,T10NINT,PKTSSR
                042322  104456                                                          TRAP    C$ERHRD
                042324  002005                                                          .WORD   1029
                042326  043277                                                          .WORD   T10NINT
                042330  012116                                                          .WORD   PKTSSR
```

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,T10SSR,PKTSSR     ;NBA NOT ZERO
      042360  104456                                                                  TRAP    C$ERHRD
      042362  002006                                                                  .WORD   1030
      042364  043210                                                                  .WORD   T10SSR
      042366  012116                                                                  .WORD   PKTSSR
 2905 042370                                  ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      042370                                                                       10002$:
      042370  104405                                                                   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$:
 2910 042404                                  ENDSUB                          ;////////////// END SUBTEST ////////////
      042404                                                                       L10100:
      042404  104403                                                                  TRAP    C$ESUB
 2911 042406                          30$:
 2912
 2913                                 ;+
 2914                                 ;TEST 10 SUBTEST 4
 2915                                 ;
 2916                                 ;CHECKS THAT THE REGISTER MODIFICATION REFUSED (RMR) BIT IN
 2917                                 ;THE TSSR WILL BE SET IF A WRITE CHARACTERISTICS COMMAND
 2918                                 ;BEING EXECUTED AND ANOTHER "WC" COMMAND IS ATTEMPTED
 2919                                 ;
 2920                                 ;-
 2921
 2922 042406                                  BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
      042406                                                                       T10.4:
      042406  104402                                                                  TRAP    C$BSUB
 2923
 2924 042410  004737  043560                  JSR     PC,T10RT2               ;SET SECOND PACKET UP
 2925 042414  004737  043506                  JSR     PC,T10RST               ;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                                  BGNSEG                          ;>>>>>>>>>>>>> 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                                                                  .WORD   SFIMSG
```

K13

TEST 10: BASIC PACKET PROTOCOL

```
 2940 042474  005037  002220          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 2941 042500  005037  002222                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 2942 042504  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 2943 042510  010365  000000                  MOV     R3,TSDB(R5)             ;SECOND COMMAND PACKET
 2944 042514  004737  016330                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 2945 042520  016501  000002                  MOV     TSSR(R5),R1             ;GET CONTENTS OF TSSR REGISTER
 2946 042524  032701  000200                  BIT     #SSR,R1                 ;CHECK FOR SSR (TSSR) SET
 2947 042530  001006                          BNE     15$                     ;BR, IF SSR SET (GOOD)
 2951 042532                                  ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      042532  104455                                                                  TRAP    C$ERDF
      042534  002010                                                                  .WORD   1032
      042536  043210                                                                  .WORD   T10SSR
      042540  012116                                                                  .WORD   PKTSSR
 2952 042542  005237  002220                  INC     FATFLG                  ;SET FATAL ERROR FLAG
 2953 042546                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      042546  104406                                                                  TRAP    C$CLP1
 2954 042550                                  ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
      042550  104410                                                                  TRAP    C$ESCAPE
      042552  000056                                                                  .WORD   10000$-.
 2955 042554  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 2956 042560  001004                          BNE     22$                     ;BRANCH IF YES
 2957
 2958
 2962 042562                                  ERRHRD  ERRNO,T10NINT,PKTSSR
      042562  104456                                                                  TRAP    C$ERHRD
      042564  002011                                                                  .WORD   1033
      042566  043277                                                                  .WORD   T10NINT
      042570  012116                                                                  .WORD   PKTSSR
 2963 042572  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 2964 042576  012702  110200                  MOV     #SSR!RMR!SC,R2          ;EXPECTED CONTENTS OF TSSR
 2965 042602  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 2966 042606  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 2967 042610  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 2968 042614  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 2969 042616  001404                          BEQ     30$                     ;OKAY IF MATCH
 2973 042620                                  ERRHRD  ERRNO,T10SSR,PKTSSR     ;NBA NOT ZERO
      042620  104456                                                                  TRAP    C$ERHRD
      042622  002012                                                                  .WORD   1034
      042624  043210                                                                  .WORD   T10SSR
      042626  012116                                                                  .WORD   PKTSSR
 2974 042630                          30$:
 2975 042630                                  ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      042630                                                                  10000$:
      042630  104405                                                                  TRAP    C$ESEG
 2976 042632                                  ENDSUB                          ;////////////// END SUBTEST ////////////
      042632                                                                  L10101:
      042632  104403                                                                  TRAP    C$ESUB
 2977 042634                                  EXIT    TST                     ;ALL DONE WITH THIS TEST
      042634  104432                                                                  TRAP    C$EXIT
      042636  000774                                                                  .WORD   L10075-.
 2978
 2979                                 ;+
 2980                                 ;LOCAL STORAGE FOR THIS TEST
 2981                                 ;-
 2982
 2984 042640                                  .BLKB   10-<.-TSV2&7>
 2986 042650                          T10PACKET:                              ;COMMAND PACKET FOR TEST
```

```
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                                 ;
3036                                 ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
3037                                 ;
3038                                 ;-
3039
3040 043506                          T10RST:
3041 043506                                 SAVREG                        ;SAVE THE REGISTERS
3042 043512  012701  042650                MOV     #T10PACKET,R1         ;START OF THE PACKET
3043 043516  012721  100204                MOV     #100204,(R1)+         ;WRITE CHARACTERISTICS WITH ACK, IE
```

```
TEST  10: BASIC PACKET PROTOCOL

 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                                 ;ROUTINE TO RESTORE COMMAND PACKET #2 TO START-UP (DEFAULT) VALUES
 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
      043632                                                                  L10075:
      043632  104401                                                                  TRAP    C$ETST
 3075
 3076                                         .SBTTL  TEST  11: NON-TAPE MOTION COMMANDS
 3077
 3078                                 ;+
 3079                                 ;
 3080                                 ;THIS TEST VERIFIES PROPER OPERATION OF THE INITIALIZE
 3081                                 ;COMMAND. TWO SUBTESTS ARE USED. THE FIRST VERIFIES THAT
 3082                                 ;THE COMMAND RUNS TO COMPLETION AND STORES A VALID
 3083                                 ;MESSAGE PACKET. THE SECOND VERIFIES THAT NON-ZERO
 3084                                 ;VALUES IN THE COMMAND MODE FIELD CAUSES COMMAND REJECT.
 3085                                 ;
 3086                                 ;-
 3087
 3088 043634                                  BGNTST
      043634                                                                  T11::
 3093 043634  012700  045642                  MOV     #TST11ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
 3094 043640  004737  016570                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
 3095 043644  012737  000024  002214          MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
 3096 043652                          T11LOOP:
 3097 043652                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
      043652                                                                  T11.1:
      043652  104402                                                                  TRAP    C$BSUB
 3098
 3099 043654                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
```

TEST 11: NON-TAPE MOTION COMMANDS

```
        043654  012700  000000                                                  MOV     #PRI00,R0
        043660  104441                                                          TRAP    C$SPRI
  3100  043662  004737  016054          JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
  3101  043666  103405                  BCS     3$              ;BR IF SOFT INIT = OK
  3105  043670  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
  3106  043672                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
        043672  104455                                                          TRAP    C$ERDF
        043674  002115                                                          .WORD   1101
        043676  003652                                                          .WORD   SFIERR
        043700  012104                                                          .WORD   SFIMSG
  3107  043702                  3$:
  3108  043702  012704  045070          MOV     #T11PK2,R4      ;WRITE CHARACTERISTICS PACKET
  3109  043706  004737  010662          JSR     PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
  3110  043712  103404                  BCS     4$              ;BR, IF COMMAND ISSUED OK
  3114  043714                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
        043714  104456                                                          TRAP    C$ERHRD
        043716  002116                                                          .WORD   1102
        043720  005056                                                          .WORD   WRTMSG
        043722  012104                                                          .WORD   SFIMSG
  3115  043724                  4$:
  3116  043724  004737  045674          JSR     PC,T11REST      ;SET UP PACKET FOR COMMAND
  3117  043730  012704  045020          MOV     #T11PACKET,R4   ;GET THE ADDRESS OF COMMAND PACKET
  3118  043734                  5$:
  3119  043734                          BGNSEG                  ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
        043734  104404                                                          TRAP    C$BSEG
  3120
  3121  043736  005037  002220  10$:    CLR     FATFLG          ;CLEAR FATAL ERROR FLAG
  3122  043742  005037  002222          CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
  3123  043746  010465  000000          MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
  3124  043752  004737  016416          JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
  3125  043756  103407                  BCS     15$             ;BR IF CARRY SET (GOOD RETURN)
  3126  043760  010001                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
  3130  043762                          ERRDF   ERRNO,T11SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        043762  104455                                                          TRAP    C$ERDF
        043764  002117                                                          .WORD   1103
        043766  045364                                                          .WORD   T11SSR
        043770  012116                                                          .WORD   PKTSSR
  3131  043772  005237  002220          INC     FATFLG          ;SET FATAL ERROR FLAG
  3132  043776                  15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
        043776  104406                                                          TRAP    C$CLP1
  3133  044000                          ESCAPE  SEG             ;BY-PASS SUBTEST IF FATAL ERROR
        044000  104410                                                          TRAP    C$ESCAPE
        044002  000074                                                          .WORD   10000$-.
  3134  044004  005737  002222          TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
  3135  044010  001004                  BNE     22$             ;BRANCH IF YES
  3139  044012                          ERRHRD  ERRNO,T11NINT,PKTSSR
        044012  104456                                                          TRAP    C$ERHRD
        044014  002120                                                          .WORD   1104
        044016  045514                                                          .WORD   T11NINT
        044020  012116                                                          .WORD   PKTSSR
  3140  044022  016501  000002  22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
  3141  044026  012702  000200          MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
  3142  044032  032701  000100          BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
  3143  044036  001402                  BEQ     25$             ;BRANCH IF NOT OFF-LINE
  3144  044040  052702  000100          BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
  3145  044044  020201          25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
  3146  044046  001404                  BEQ     30$             ;OKAY IF MATCH
```

```
TEST  11: NON-TAPE MOTION COMMANDS

3150 044050                                ERRHRD  ERRNO,T11NBA,PKTSSR     ;NBA NOT ZERO
     044050  104456                                                                TRAP    C$ERHRD
     044052  002121                                                                .WORD   1105
     044054  045132                                                                .WORD   T11NBA
     044056  012116                                                                .WORD   PKTSSR
3151 044060                          30$:
3152 044060  004737  011154          35$:  JSR     PC,CKRAM                  ;CHECK RAM TO MEMORY
3153 044064  103405                       BCS     59$                       ;RAM OK GO ON
3157 044066                                ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
     044066  104456                                                                TRAP    C$ERHRD
     044070  002122                                                                .WORD   1106
     044072  004745                                                                .WORD   PKTRAM
     044074  015570                                                                .WORD   RAMERR
3158 044076                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     044076                                                                   10000$:
     044076  104405                                                                TRAP    C$ESEG
3159
3160
3161 044100                          59$:  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     044100                                                                   L10103:
     044100  104403                                                                TRAP    C$ESUB
3162
3163 044102  005737  002220                TST     FATFLG                  ;ANY FATAL ERRORS ?
3164 044106  001402                       BEQ     60$                     ;BRANCH IF NOT
3165 044110  004737  017262                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
3166 044114                          60$:
3167
3168                                 ;*
3169                                 ;
3170                                 ;TEST 11, SUBTEST 2
3171                                 ;
3172                                 ;CHECK THAT NON-ZERO MODE BITS BEING SET CAUSES
3173                                 ;INITIALIZE COMMAND TO BE REJECTED
3174                                 ;
3175                                 ;-
3176
3177 044114                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     044114                                                                   T11.2:
     044114  104402                                                                TRAP    C$BSUB
3178
3179 044116                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     044116  012700  000000                                                        MOV     #PRI00,R0
     044122  104441                                                                TRAP    C$SPRI
3180 044124                                BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     044124  104404                                                                TRAP    C$BSEG
3181
3182
3183 044126  004737  016054                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
3184 044132  103405                       BCS     3$                      ;BR IF SOFT INIT = OK
3188 044134  010001                       MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3189 044136                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     044136  104455                                                                TRAP    C$ERDF
     044140  002123                                                                .WORD   1107
     044142  003652                                                                .WORD   SFIERR
     044144  012104                                                                .WORD   SFIMSG
3190 044146                          3$:
3191 044146  012704  045070                MOV     #T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
```

TEST 11: NON-TAPE MOTION COMMANDS

```
3192 044152  004737  010662            JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
3193 044156  103404                    BCS     4$                      ;BR, IF COMMAND ISSUED OK
3197 044160                            ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     044160  104456                                                            TRAP    C$ERHRD
     044162  002124                                                            .WORD   1108
     044164  005056                                                            .WORD   WRTMSG
     044166  012104                                                            .WORD   SFIMSG
3198 044170                        4$:
3199 044170  004737  045674            JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
3200 044174  012704  045020            MOV     #T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3201 044200                        5$:
3202 044200  005037  002222        10$: CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
3203 044204  052714  007400            BIS     #P.MODE,(R4)            ;NON-ZERO COMMAND MODE BITS
3204 044210  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
3205 044214  004737  016416            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3206 044220  103405                    BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
3207 044222  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3211 044224                            ERRDF   ERRNO,T11SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     044224  104455                                                            TRAP    C$ERDF
     044226  002125                                                            .WORD   1109
     044230  045364                                                            .WORD   T11SSR
     044232  012116                                                            .WORD   PKTSSR
3212 044234                        15$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044234  104406                                                            TRAP    C$CLP1
3213 044236                            ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
     044236  104410                                                            TRAP    C$ESCAPE
     044240  000074                                                            .WORD   10000$-.
3214 044242  005737  002222            TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
3215 044246  001004                    BNE     22$                     ;BRANCH IF YES
3219 044250                            ERRHRD  ERRNO,T11NINT,PKTSSR
     044250  104456                                                            TRAP    C$ERHRD
     044252  002126                                                            .WORD   1110
     044254  045514                                                            .WORD   T11NINT
     044256  012116                                                            .WORD   PKTSSR
3220 044260  016501  000002        22$: MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
3221 044264  012702  100206            MOV     #SC!SSR!TSREJ,R2        ;EXPECTED CONTENTS OF TSSR
3222 044270  032701  000100            BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
3223 044274  001402                    BEQ     25$                     ;BRANCH IF NOT OFF-LINE
3224 044276  052702  000100            BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
3225 044302  020201                25$: CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
3226 044304  001404                    BEQ     30$                     ;OKAY IF MATCH
3230 044306                            ERRHRD  ERRNO,T112REJ,PKTSSR    ;COMMAND NOT REJECTED
     044306  104456                                                            TRAP    C$ERHRD
     044310  002127                                                            .WORD   1111
     044312  045172                                                            .WORD   T112REJ
     044314  012116                                                            .WORD   PKTSSR
3231 044316                        30$:
3232 044316  004737  011154        35$: JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
3233 044322  103405                    BCS     59$                     ;RAM OK GO ON
3237 044324                            ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
     044324  104456                                                            TRAP    C$ERHRD
     044326  002130                                                            .WORD   1112
     044330  004745                                                            .WORD   PKTRAM
     044332  015570                                                            .WORD   RAMERR
3238 044334                            ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     044334                                                                 10000$:
     044334  104405                                                            TRAP    C$ESEG
```

```
3239
3240 044336                          59$:    ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     044336                                                                          L10104:
     044336  104403                                                                        TRAP    C$ESUB
3241
3242                                 ;+
3243                                 ;
3244                                 ;TEST 11, SUBTEST 3
3245                                 ;
3246                                 ;CHECK THAT THE GET STATUS COMMAND IS ACCEPTED
3247                                 ;
3248                                 ;-
3249
3250 044340                                 BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     044340                                                                          T11.3:
     044340  104402                                                                  TRAP    C$BSUB
3251
3252 044342                                 SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     044342  012700  000000                                                          MOV     #PRI00,R0
     044346  104441                                                                  TRAP    C$SPRI
3253 044350                                 BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     044350  104404                                                                  TRAP    C$BSEG
3254 044352  004737  016054                 JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
3255 044356  103405                       BCS     3$                      ;BR IF SOFT INIT = OK
3259 044360  010001                       MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3260 044362                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     044362  104455                                                                  TRAP    C$ERDF
     044364  002131                                                                  .WORD   1113
     044366  003652                                                                  .WORD   SFIERR
     044370  012104                                                                  .WORD   SFIMSG
3261 044372                          3$:
3262 044372  012704  045070                 MOV     #T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
3263 044376  004737  010662                 JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
3264 044402  103404                       BCS     4$                      ;BR, IF COMMAND ISSUED OK
3268 044404                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     044404  104456                                                                  TRAP    C$ERHRD
     044406  002132                                                                  .WORD   1114
     044410  005056                                                                  .WORD   WRTMSG
     044412  012104                                                                  .WORD   SFIMSG
3269 044414                          4$:
3270 044414  004737  045674                 JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
3271 044420  012704  045020                 MOV     #T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3272 044424                          5$:
3273 044424  005037  002222          10$:   CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
3274 044430  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
3275 044434  004737  016416                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3276 044440  103405                       BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
3277 044442  010001                       MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3281 044444                                ERRDF   ERRNO,T11SR2,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     044444  104455                                                                  TRAP    C$ERDF
     044446  002133                                                                  .WORD   1115
     044450  045440                                                                  .WORD   T11SR2
     044452  012116                                                                  .WORD   PKTSSR
3282 044454                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044454  104406                                                                  TRAP    C$CLP1
3283 044456                                 ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     044456  104410                                                                  TRAP    C$ESCAPE
```

```
TEST  11: NON-TAPE MOTION COMMANDS

      044460  000074                                                          .WORD   10000$-.
 3284 044462  005737  002222            TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 3285 044466  001004                    BNE     22$                     ;BRANCH IF YES
 3289 044470                            ERRHRD  ERRNO,T11NINT,PKTSSR
      044470  104456                                                          TRAP    C$ERHRD
      044472  002134                                                          .WORD   1116
      044474  045514                                                          .WORD   T11NINT
      044476  012116                                                          .WORD   PKTSSR
 3290 044500  016501  000002    22$:    MOV     TSAR(R5),R1             ;GET THE CONTENTS OF TSSR
 3291 044504  012702  000200            MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
 3292 044510  032701  000100            BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 3293 044514  001402                    BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 3294 044516  052702  000100            BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 3295 044522  020201    25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 3296 044524  001404                    BEQ     30$                     ;OKAY IF MATCH
 3300 044526                            ERRHRD  ERRNO,T113REJ,PKTSSR    ;COMMAND NOT ACCEPTED
      044526  104456                                                          TRAP    C$ERHRD
      044530  002135                                                          .WORD   1117
      044532  045253                                                          .WORD   T113REJ
      044534  012116                                                          .WORD   PKTSSR
 3301 044536            30$:
 3302 044536  004737  011154    35$:    JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
 3303 044542  103405                    BCS     59$                     ;RAM OK GO ON
 3307 044544                            ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
      044544  104456                                                          TRAP    C$ERHRD
      044546  002136                                                          .WORD   1118
      044550  004745                                                          .WORD   PKTRAM
      044552  015570                                                          .WORD   RAMERR
 3308 044554                            ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      044554                                                            10000$:
      044554  104405                                                          TRAP    C$ESEG
 3309
 3310
 3311 044556            59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      044556                                                            L10105:
      044556  104403                                                          TRAP    C$ESUB
 3312
 3313                           ;*
 3314                           ;
 3315                           ;TEST 11, SUBTEST 4
 3316                           ;
 3317                           ;SUBTEST TO VERIFY THAT A GET STATUS COMMAND IS
 3318                           ;REJECTED IF A NON-ZERO COMMAND MODE FIELD IS USED
 3319                           ;
 3320                           ;-
 3321
 3322 044560                            BGNSUB                  ;////////////// BEGIN SUBTEST //////////////
      044560                                                            T11.4:
      044560  104402                                                          TRAP    C$BSUB
 3323
 3324 044562                            SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      044562  012700  000000                                                    MOV     #PRI00,R0
      044566  104441                                                          TRAP    C$SPRI
 3325 044570                            BGNSEG                  ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
      044570  104404                                                          TRAP    C$BSEG
 3326 044572  004737  016054            JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 3327 044576  103405                    BCS     3$                      ;BR IF SOFT INIT = OK
```

F14

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$ERDF
     044604  002137                                                                .WORD   1119
     044606  003652                                                                .WORD   SFIERR
     044610  012104                                                                .WORD   SFIMSG
3333 044612                          3$:
3334 044612  012704  045070                  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 CHARACTERISTISC FAILED
     044624  104456                                                                TRAP    C$ERHRD
     044626  002140                                                                .WORD   1120
     044630  005056                                                                .WORD   WRTMSG
     044632  012104                                                                .WORD   SFIMSG
3341 044634                          4$:
3342 044634  004737  045674                  JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
3343 044640  012704  045020                  MOV     #T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3344 044644                          5$:
3345 044644  005037  002222          10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
3346 044650  052714  007000                  BIS     #007000,(R4)            ;SET TO NON-ZERO MODE
3347 044654  010465  000000                  MOV     R4,TSDB(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     #OFL,R1                 ;IS OFF-LINE BIT SET ?
3366 044740  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
3367 044742  052702  000100                  BIS     #OFL,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$ERHRD
     044754  002143                                                                .WORD   1123
     044756  045303                                                                .WORD   T114REJ
     044760  012116                                                                .WORD   PKTSSR
3374 044762                          30$:
3375
3376 044762  004737  011154          35$:    JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
```

```
3377 044766  103404                        BCS     59$                     ;RAM OK GO ON
3381 044770                                ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
     044770  104456                                                            TRAP    C$ERHRD
     044772  002144                                                            .WORD   1124
     044774  004745                                                            .WORD   PKTRAM
     044776  015570                                                            .WORD   RAMERR
3382 045000                           59$:
3383 045000                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<
     045000                                                                  10000$:
     045000  104405                                                            TRAP    C$ESEG
3384 045002                                ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     045002                                                                  L10106:
     045002  104403                                                            TRAP    C$ESUB
3385
3386 045004                                EXIT    TST                     ;ALL DONE THIS TEST
     045004  104432                                                            TRAP    C$EXIT
     045006  000772                                                            .WORD   L10102-.
3387
3388                                  ;*
3389                                  ;LOCAL STORAGE FOR THIS TEST
3390                                  ;-
3391
3393 045010                                .BLKB   10-<.-TSV2&7>
3395 045020                           T11PACKET:
3396 045020  100204                        .WORD   100204                  ;COMMAND PACKET FOR TEST
3397 045022  045030                        .WORD   T11DATA                 ;WRITE CHAR COMMAND, WITH IE, ACK
3398 045024  000000                        .WORD   0                       ;ADDRESS OF CHARACTERISTICS BLOCK
3399 045026  000010                        .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
3400
3401 045030                           T11DATA:                                ;CHARACTERISTICS DATA BLOCK
3402 045030  045042                        .WORD   T11BFR                  ;ADDRESS OF MESSAGE BUFFER
3403 045032  000000                        .WORD   0
3404 045034  000016                        .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
3405 045036  000000  000000                .WORD   0,0
3406
3407 045042                           T11BFR: .BLKW   8.                     ;MESSAGE BUFFER
3408
3409
3411 045062                                .BLKB   10-<.-TSV2&7>
3413 045070                           T11PK2:                                 ;COMMAND PACKET FOR TEST
3414 045070  100204                        .WORD   100204                  ;WRITE CHAR COMMAND, WITH IE, ACK
3415 045072  045100                        .WORD   T11DTA                  ;ADDRESS OF CHARACTERISTICS BLOCK
3416 045074  000000                        .WORD   0
3417 045076  000010                        .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
3418
3419 045100                           T11DTA:                                 ;CHARACTERISTICS DATA BLOCK
3420 045100  045112                        .WORD   T11BF2                  ;ADDRESS OF MESSAGE BUFFER
3421 045102  000000                        .WORD   0
3422 045104  000016                        .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
3423 045106  000000  000000                .WORD   0,0
3424
3425 045112                           T11BF2: .BLKW   8.                     ;MESSAGE BUFFER
3426
3427
3428
3429                                  ;*
3430                                  ;LOCAL TEXT MESSAGES FOR TEST
```

TEST 11: NON-TAPE MOTION COMMANDS

```
3431                                    ;-
3432
3433 045132  111  116  111  T11NBA: .ASCIZ  'INITIALIZE Command Not Accepted'
3434 045172  111  116  111  T112REJ:        .ASCIZ  'INITIALIZE Not Rejected With Non-Zero Mode Field'
3435 045253  107  105  124  T113REJ:        .ASCIZ  'GET STATUS Not Accepted'
3436 045303  107  105  124  T114REJ:        .ASCIZ  'GET STATUS Not Rejected With Non-Zero Mode Field'
3437 045364  103  157  156  T11SSR: .ASCIZ  'Contents of TSSR Incorrect After INITIALIZE'
3438 045440  103  157  156  T11SR2: .ASCIZ  'Contents of TSSR Incorrect After GET STATUS'
3439 045514  105  170  160  T11NINT:        .ASCIZ  'Expected Interrupt Not Received On INITIALIZE'
3440 045572  111  156  143  T11TSBA:        .ASCIZ  'Incorrect TSBA Address After INITIALIZE'
3441 045642  116  157  156  TST11ID:        .ASCIZ  'Non-Tape Motion Commands'
3442                                            .EVEN
3443
3444
3445                                    ;*
3446                                    ;
3447                                    ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
3448                                    ;INITIALIZE COMMAND
3449                                    ;
3450                                    ;-
3451
3452 045674                             T11REST:
3453 045674                                     SAVREG                          ;SAVE THE REGISTERS
3454 045700  012701  045020                      MOV     #T11PACKET,R1   ;START OF THE PACKET
3455 045704  012721  100213                      MOV     #100213,(R1)+   ;INITIALIZE WITH ACK, IE
3456 045710  005021                              CLR     (R1)+           ;ADDRESS OF CHAR DATA BLOCK
3457 045712  005021                              CLR     (R1)+           ;EXTENDED ADDRESS
3458 045714  005021                              CLR     (R1)+           ;SIZE OF DATA BLOCK IN BYTES
3459 045716  005021                              CLR     (R1)+           ;ADDRESS OF MESSAGE BUFFER
3460 045720  005021                              CLR     (R1)+
3461 045722  005021                              CLR     (R1)+           ;LENGTH OF MESSAGE BUFFER
3462 045724  005021                              CLR     (R1)+
3463 045726  005011                              CLR     (R1)
3464 045730  005037  045042                      CLR     T11BFR          ;CLEAR 1ST LOC IN MESSAGE BUFFER
3465 045734  000207                              RTS     PC              ;RETURN
3466                                    ;*
3467                                    ;
3468                                    ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
3469                                    ;GET STATUS COMMAND
3470                                    ;
3471                                    ;-
3472
3473 045736                             T11RT2:
3474 045736                                     SAVREG                          ;SAVE THE REGISTERS
3475 045742  012701  045020                      MOV     #T11PACKET,R1   ;START OF THE PACKET
3476 045746  012721  100217                      MOV     #100217,(R1)+   ;GET STATUS WITH ACK, IE
3477 045752  005021                              CLR     (R1)+           ;ADDRESS OF CHAR DATA BLOCK
3478 045754  005021                              CLR     (R1)+           ;EXTENDED ADDRESS
3479 045756  005021                              CLR     (R1)+           ;SIZE OF DATA BLOCK IN BYTES
3480 045760  005021                              CLR     (R1)+           ;ADDRESS OF MESSAGE BUFFER
3481 045762  005021                              CLR     (R1)+
3482 045764  005021                              CLR     (R1)+           ;LENGTH OF MESSAGE BUFFER
3483 045766  005021                              CLR     (R1)+
3484 045770  005011                              CLR     (R1)
3485 045772  005037  045042                      CLR     T11BFR          ;CLEAR 1ST LOC IN MESSAGE BUFFER
3486 045776  000207                              RTS     PC              ;RETURN
3487 046000                                      ENDTST
```

TEST 11: NON-TAPE MOTION COMMANDS

```
                046000
                046000  104401
  3488          046002                          ENDMOD

                                                L10102:  TRAP    C$ETST
```

```
      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  000010                          .WORD L10107-L$HARD/2
        046004                          L$HARD::
     32
     33 046004                                  GPRMA   HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
        046004  000031                          .WORD   T$CODE
        046006  046024                          .WORD   HPM1
        046010  160010                          .WORD   T$LOLIM
        046012  177776                          .WORD   T$HILIM
     34 046014                                  GPRMA   HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
        046014  001031                          .WORD   T$CODE
        046016  046060                          .WORD   HPM2
        046020  000000                          .WORD   T$LOLIM
        046022  000776                          .WORD   T$HILIM
     35                                         ;GPRMD  HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
     36 046024                                  ENDHRD
                                                .EVEN
        046024                          L10107:
     37 046024       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

```
                                        .SBTTL  SOFTWARE PARAMETER CODING SECTION

                                ;++
                                ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--
046134                                  BGNSFT
046134  000003                          .WORD L10110-L$SOFT/2
046136                          L$SOFT::
                                ;       GPRML   SPM1,0,-1,YES                   ; GET TRANSPORT TEST FLAG.
046136                                  GPRML   SPM4,2,-1,YES                   ; GET ITERATION CONTROL.
046136  001130                          .WORD   T$CODE
046140  046174                          .WORD   SPM4
046142  177777                          .WORD   -1
                                ;       GPRMD   SPM6,4,D,7777,0,7777,YES        ; GET LOCAL ERROR LIMIT
                                ;       GPRMD   SPM7,6,D,7777,0,7777,YES        ; GET GLOBAL ERROR LIMIT
046144                                  ENDSFT
                                        .EVEN
046144                          L10110:

046144  105 116 101             SPM1:   .ASCIZ  'ENABLE TRANSPORT TESTS '
046174  111 116 110             SPM4:   .ASCIZ  'INHIBIT ITERATIONS     '
046224  120 105 122             SPM6:   .ASCIZ  'PER TEST ERROR LIMIT   '
046254  120 105 122             SPM7:   .ASCIZ  'PER UNIT ERROR LIMIT   '
                                        .SBTTL  PATCH AREA


                                ;
                                ; FINALLY A GENEROUS PATCH AREA.
                                ;
                                ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
                                ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
                                ;

046304                          PATCH::

                                        .BLKW   32.

        047000                          .=.!377+1
047000                                  LASTAD          ;SET LAST USED ADDRESS.
                                        .EVEN
047000  000000                          .WORD   0
047002  000000                          .WORD   0
047004                          L$LAST::
047004                                  ENDMOD
        000001                          .END
```

L14
TSV6 - PARAMETER CODING MACRO V05.03 Tuesday 28-Apr-87 10:28 Page 87-1
SEQ 0180

Symbol table

```
ADDSSR  012176 C        C$AU  = 000052        DEVDRO  023412        FRESIZ  003126 G        INTFLA  016225
ADR   = 000020 G        C$AUTO= 000061        DEVNRD  023331        FUSI    004117          INTMAS  016224
AMBTSS  006635          C$BRK = 000022        DEVNXR  023247        F$AU  = 000015          INTR    016276 G
ASSEMB= 000010          C$BSEG= 000004        DEVONL  023177        F$AUTO= 000020          INTREC  002222 G
A1716 = 000003          C$BSUB= 000002        DEVSUM  023142        F$BGN = 000040          INTVEC  016226
BADDAT  003156 G        C$CEFG= 000045        DFPTBL  002154 G      F$CLEA= 000007          INTX    004300
BADSSR  015760 G        C$CLCK= 000062        DIAGMC= 000000        F$DU  = 000016          INVERT  021222 G
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$CVEC= 000036        DUFLG   003112 G      F$INIT= 000006          ISR   = 000100 G
BIT00 = 000001 G        C$DCLN= 000044        DUMMY   003062        F$JMP = 000050          IVEC    002206 G
BIT01 = 000002 G        C$DODU= 000051        EF.CON= 000036 G      F$MOD = 000000          IXE   = 004000 G
BIT02 = 000004 G        C$DRPT= 000024        EF.NEW= 000035 G      F$MSG = 000011          I$AU  = 000041
BIT03 = 000010 G        C$DU  = 000053        EF.PWR= 000034 G      F$PROT= 000021          I$AUTO= 000041
BIT04 = 000020 G        C$EDIT= 000003        EF.RES= 000037 G      F$PWR = 000017          I$CLN = 000041
BIT05 = 000040 G        C$ERDF= 000055        EF.STA= 000040 G      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$ERSF= 000054        ENAINT  016232        F$SRV = 000010          I$MOD = 000041
BIT09 = 001000 G        C$ERSO= 000057        ENVIRN  020710        F$SUB = 000002          I$MSG = 000041
BIT1  = 000002 G        C$ESCA= 000010        EPRTSW  002176 G      F$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 = 000041
BIT13 = 020000 G        C$EXIT= 000032        ERRHI   002236 G      GETPAT  020254 G        I$SEG = 000041
BIT14 = 040000 G        C$GETB= 000026        ERRK    017036        GETSEL  020336 G        I$SETU= 000041
BIT15 = 100000 G        C$GETW= 000027        ERRLO   002240 G      G$CNTO= 000200          I$SFT = 000041
BIT2  = 000004 G        C$GMAN= 000043        ERRNO = 002144        G$DELM= 000372          I$SRV = 000041
BIT3  = 000010 G        C$GPHR= 000042        ERRVEC= 000004 G      G$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      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      G$OFSI= 000376          KIPAR3= 172346
BOE   = 000400 G        C$MSG = 000023        EXPD    002232 G      G$PRMA= 000001          KIPAR4= 172350
BRINIT  004457          C$OPEN= 000034        EXPGOT  004533        G$PRMD= 000002          KIPAR5= 172352
BSEL0 = 000000          C$PNTB= 000014        EXPGT2  004567        G$PRML= 000000          KIPAR6= 172354
BSEL1 = 000001          C$PNTF= 000017        EXPMSG  002322 G      G$RADA= 000140          KIPAR7= 172356
CHKAMB  016124          C$PNTS= 000016        EXPREC  015554 G      G$RADB= 000000          KIPDR0= 172300
CHKMAN  020560 G        C$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      G$RADO= 000020          KIPDR3= 172306
CKEMAX  017162          C$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$RFLA= 000021        FATFLG  002220 G      HOE   = 100000 G        KIPDR7= 172316
CKRAM2  011264 G        C$RPT = 000025        FERCM   011772        HPM1    046024          KTENAB  003134 G
CMDPKT  021274 G        C$SEFG= 000046        FIFEXP  012240 G      HPM2    046060          KTFLG   003132 G
CMPMEM  017740          C$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 G      IFAULT  004256          LISTAL= 000001
CTABE   003176 G        DEBUGM  011702        FREE    003124 G      INCERR  017124          LOE   = 040000 G
CTABM   003164 G        DEVCNT  002216 G      FREEMI  003130        INTCPC  016230          LOOPCN  002214 G
```

M14

Symbol table

```
LOOPCO  013176          L10001  002174          L10073  037126          NXR     003740          PRIO5 = 000240 G
LOOPFL  003162 G        L10002  005766          L10074  037332          NXRERR  005736 G        PRIO6 = 000300 G
LOT   = 000010 G        L10003  012114          L10075  043632          NXRX    003777          PRIO7 = 000340 G
L$ACP   002110 G        L10004  012132          L10076  041122          NXTU    022036          PRMESS  014312
L$APT   002036 G        L10005  012150          L10077  041514          OFL   = 000100          PRMNO   002320 G
L$AU    022366 G        L10006  012156          L10100  042404          ONEFIL= 000000          PRMSGE  014632 G
L$AUT   002070 G        L10007  012174          L10101  042632          O$APTS= 000000          PRMSGO  015012
L$AUTO  022572 G        L10010  012212          L10102  046000          O$AU  = 000001          PRMSG1  015057
L$CCP   002106 G        L10011  012236          L10103  044100          O$BGNR= 000001          PRMSG2  015115
L$CLEA  022652 G        L10012  012310          L10104  044336          O$BGNS= 000001          PROASC  014500
L$CO    002032 G        L10013  012460          L10105  044556          O$DU  = 000001          PR1ASC  014545
L$DEPO  002011 G        L10014  013174          L10106  045002          O$ERRT= 000000          PST32W  003152 G
L$DESC  003410 G        L10015  014022          L10107  046024          O$GNSW= 000001          PUNIT   022320
L$DESP  002076 G        L10016  014044          L10110  046144          O$POIN= 000001          PW.D11= 000021
L$DEVP  002060 G        L10017  015560          MEMADD  014024 G        O$SETU= 000000          PW.D13= 000022
L$DISP  002124 G        L10020  015566          MEMCK   021312 G        PASRPT  022070          PW.D22= 000020
L$DLY   002116 G        L10021  015574          MENASC  020527          PATCH   046304 G        PW.NOP= 000000
L$DTP   002040 G        L10022  015606          MENERR  020454          PATDAT  020310          PW.NO1= 000023
L$DTYP  002034 G        L10023  015630          MENRES  020556          PC.ERA= 002400          PW.RDE= 000024
L$DU    022464 G        L10024  015656          MMVEC = 000250          PC.IER= 002000          PW.RDR= 000001
L$DUT   002072 G        L10025  016016          MSA.FR= 000006          PC.NOO= 001000          PW.RDS= 000005
L$DVTY  003402 G        L10026  016326          MSA.NO= 000000          PC.REL= 000000          PW.RFI= 000003
L$EF    002052 G        L10030  022316          MSA.NR= 000004          PC.REW= 000400          PW.WCT= 000006
L$ENVI  002044 G        L10031  022462          MSA.VO= 000002          PKBCNT= 000006          PW.WFI= 000004
L$ETP   002102 G        L10032  022570          MSGEXP  012214 G        PKHI  = 000004          PW.WFM= 000007
L$EXP1  002046 G        L10033  022650          MSGLOO  013134 G        PKLOW = 000002          PW.WMI= 000010
L$EXP4  002064 G        L10034  022676          MSGSTA  012420 G        PKTADD  007554          PW.WNP= 000011
L$EXP5  002066 G        L10035  023140          MSGSUB  014012 G        PKTFRM  007516          PW.WTR= 000002
L$HARD  046004 G        L10036  023700          MS.ATT= 000006          PKTGET  012134 G        P.ACK = 100000
L$HIME  002120 G        L10037  023546          MS.EXT= 000200          PKTMES  012160 G        P.CMD = 000037
L$HPCP  002016 G        L10040  023630          MS.RSD= 000001          PKTRAM  004745 G        P.CONT= 000012
L$HPTP  002022 G        L10041  024376          MS.RSF= 000020          PKTSSR  012116 G        P.CVC = 040000
L$HW    002154 G        L10042  025070          MS.RST= 000010          PNT   = 001000 G        P.FMT = 000140
L$ICP   002104 G        L10043  026424          M2901   026150          PRAMPK  014046          P.FORM= 000011
L$INIT  021572 G        L10044  025332          M8186   005554          PRASC   014603          P.GETS= 000017
L$LADP  002026 G        L10045  025632          M8189   005645          PRBEXP  015550          P.IE  = 000200
L$LAST  047004 G        L10046  026130          NBA   = 002000          PRBMSG  015416          P.INIT= 000013
L$LOAD  002100 G        L10047  027530          NEWPAS  022024          PRBREC  015552          P.MODE= 007400
L$LUN   002074 G        L10050  026776          NODEV   003114 G        PRBTOT  015503          P.OPP = 020000
L$MREV  002050 G        L10051  027346          NOINIT  004335          PRBYTE  015202 G        P.POSI= 000010
L$NAME  002000 G        L10052  031002          NOINTR  004221          PRI   = 002000 G        P.READ= 000001
L$PRIO  002042 G        L10053  030060          NOITS   002166 G        PRIADD  010160          P.SWB = 010000
L$PROT  021562 G        L10054  030404          NOMAN   020614          PRIAO   010230          P.WRIT= 000005
L$PRT   002112 G        L10055  034460          NOMEM   005460          PRIBXO  007612 G        P.WRTC= 000004
L$REPP  002062 G        L10056  031410          NP.IR = 000200          PRIEQU  010060          P.WRTS= 000006
L$REV   002010 G        L10057  031674          NP.LOO= 000040          PRIPKT  007370 G        QVP     002202 G
L$RPT   022700 G        L10060  032140          NP.OUT= 000100          PRIRAM  010066          RAMASC  014226
L$SOFT  046136 G        L10061  032366          NP.WRP= 000020          PRITAD  010274          RAMDAT  002242 G
L$SPC   002056 G        L10062  032632          NSI     004152          PRITSS  006024          RAMERR  015570 G
L$SPCP  002020 G        L10063  033202          NSINIT  004407          PRITO   010356          RAMEXP  015610 G
L$SPTP  002024 G        L10064  035364          NUL     004527          PRIT1   010421          RAMFOR  010116
L$STA   002030 G        L10065  040520          NULCR   004530          PRIXOR  007742 G        RAMSIZ  002302 G
L$SW    002164 G        L10066  035626          NXM   = 004000          PRIO0 = 000000 G        RAMTAD  015576 G
L$TEST  002114 G        L10067  036076          NXMFLG  003136 G        PRIO1 = 000040 G        RCVHIA  002304 G
L$TIML  002014 G        L10070  036302          NXMHI   003142 G        PRIO2 = 000100 G        RCVLOA  002306 G
L$UNIT  002012 G        L10071  036470          NXMLO   003140 G        PRIO3 = 000140 G        RDERR   005206
L10000  002162          L10072  036674          NXMTST  021466          PRIO4 = 000200 G        RECMSG  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 G	T4.1	025072		
REWIND	011054 G	S1.IID=	004000	TST9ID	040421	T10BFR	042672	T4.2	025334		
RMCHBE=	000167	S1.I1R=	020000	TSV2	002000 G	T10BUF	042734	T4.3	025634		
RMCHEN=	000200	S1.I2R=	040000	TSV3	002174 G	T10DAT	042660	T5	026426	G	
RMMSGB=	000215	S1.PAR=	100000	TSV4	021562 G	T10DTA	042722	T5ADDR	027466		
RMMSGE=	000234	S2.ATI=	000010	TSV5	023462 G	T10INT	043370	T5LOOP	026444		
RMPKTB=	000201	S2.BTI=	000004	TSV6	046002 G	T10LOO	040540	T5MEM	027430		
RMPKTE=	000210	S2.DIM=	000200	TTIBFR=	177562 G	T10MBF	042754	T5.1	026450		
RMR =	010000	S2.ILW=	000100	TTICSR=	177560 G	T10NBA	043051	T5.2	027016		
RUPACK	011150	S2.INR=	000020	TTIVEC=	000060 G	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		
SCHERR	005300	TBLEND=	003062 G	T$ERRN=	002144	T10PKT	042712	T6NBA	030450		
SCME	005013	TCOASC	006476	T$EXCP=	000000	T10RST	043506	T6NINT	030631		
SDELAY	010660	TCOCOD	006676	T$FLAG=	000040	T10RT2	043560	T6PACK	030440		
SELASC	020522	TEMP1	003116 G	T$GMAN=	000000	T10SSR	043210	T6SSR	030475		
SELDAT=	000004	TEMP2	003120 G	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$LTNO=	000013	T11	043634 G	T7BFR	033256		
SFHERR	003705	TIMEXP	015632 G	T$NEST=	177777	T11BFR	045042	T7DATA	033240		
SFIERR	003652	TIMSGO	015660	T$NS0 =	000000	T11BF2	045112	T7INT	034211		
SFIMSG	012104 G	TINERR	012071	T$NS1 =	000005	T11DAT	045030	T7LOOP	031022		
SFPTBL	002164 G	TMPBFR	002632 G	T$NS2 =	000002	T11DTA	045100	T7NBA	033370		
SIFLAG	003154 G	TNAM	016764	T$NS3 =	000003	T11LOO	043652	T7NINT	034120		
SIMSG	012036	TRANST	002164 G	T$PTNU=	000000	T11NBA	045132	T7PACK	033230		
SKIPT	003400	TSBA =	000000 G	T$SAVL=	177777	T11NIN	045514	T7REST	034412		
SOFINI	016054 G	TSBAH =	000001 G	T$SEGL=	177777	T11PAC	045020	T7SP	033250		
SPACE	010466 G	TSBAM2	026240	T$SEKO=	010000	T11PK2	045070	T7SSR	034031		
SPM1	046144	TSBAM3	026322	T$SUBN=	000004	T11RES	045674	T7TSBA	034300		
SPM4	046174	TSDB =	000000 G	T$TAGL=	177777	T11RT2	045736	T7.1	031022		
SPM6	046224	TSDBH =	000001 G	T$TAGN=	010111	T11SR2	045440	T7.2	031424		
SPM7	046254	TSFCOD	007236	T$TEMP=	000000	T11SSR	045364	T7.3	031676		
SR0 =	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 G	T$TSTS=	000001	T11.2	044114	T7.6	032634		
SR3 =	172516	TSSRBI	003502 G	T$$AU =	010031	T11.3	044340	T72DAT	033276		
SSR =	000200	TSSRFO	006415	T$$AUT=	010033	T11.4	044560	T72DON=	033312		
STATCO	012462	TSSRH =	000003 G	T$$CLE=	010034	T112RE	045172	T72NBA	033312		
SVCGBL=	000000	TSSX	004020	T$$DU =	010032	T113RE	045253	T72REJ	033443		
SVCINS=	000000	TSTBLK	002752 G	T$$HAR=	010107	T114RE	045303	T73REJ	033542		
SVCSUB=	000001	TSTCNT	002212 G	T$$HW =	010000	T2	023702 G	T74REJ	033635		
SVCTAG=	000000	TSTEND	017000	T$$INI=	010030	T2LOOP	023720	T75REJ	033733		
SVCTST=	000001	TSTFLA	002314 G	T$$MSG=	010025	T2SSR	024276	T8	034462	G	
S$LSYM=	010000	TSTLOO	016536 G	T$$PRO=	010027	T2TSBA	024164	T8BFR	035112		
SO.IDB=	000010	TSTPTR	002316 G	T$$RPT=	010035	T2TSSR	024231	T8DATA	035100		
SO.IFB=	000002	TSTSET	016570 G	T$$SEG=	010000	T23A	003144 G	T8LOOP	034500		
SO.IFP=	000001	TST1ID	023660	T$$SOF=	010110	T23B	003146 G	T8NVCK	035167		
SO.ILD=	000020	TST10I	043457	T$$SRV=	010026	T3	024400 G	T8PACK	035070		
SO.ION=	000040	TST11I	045642	T$$SUB=	010106	T3BFLG	003150 G	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 G	T3TSBA	024662	T9BFR	037372		

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          XSOTMK= 100000          X2.UNI= 000007
T9REST  040446          WC.IFA= 000200          XNXM    016456          XSOVCK= 000020          X2.WCF= 002000
T9SSR   040067          WC.IFE= 000002          XORBFO  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.I1T= 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/ML,TSV1A,TSV22A,TSV3B,TSV4,TSV5A,TSV6
```