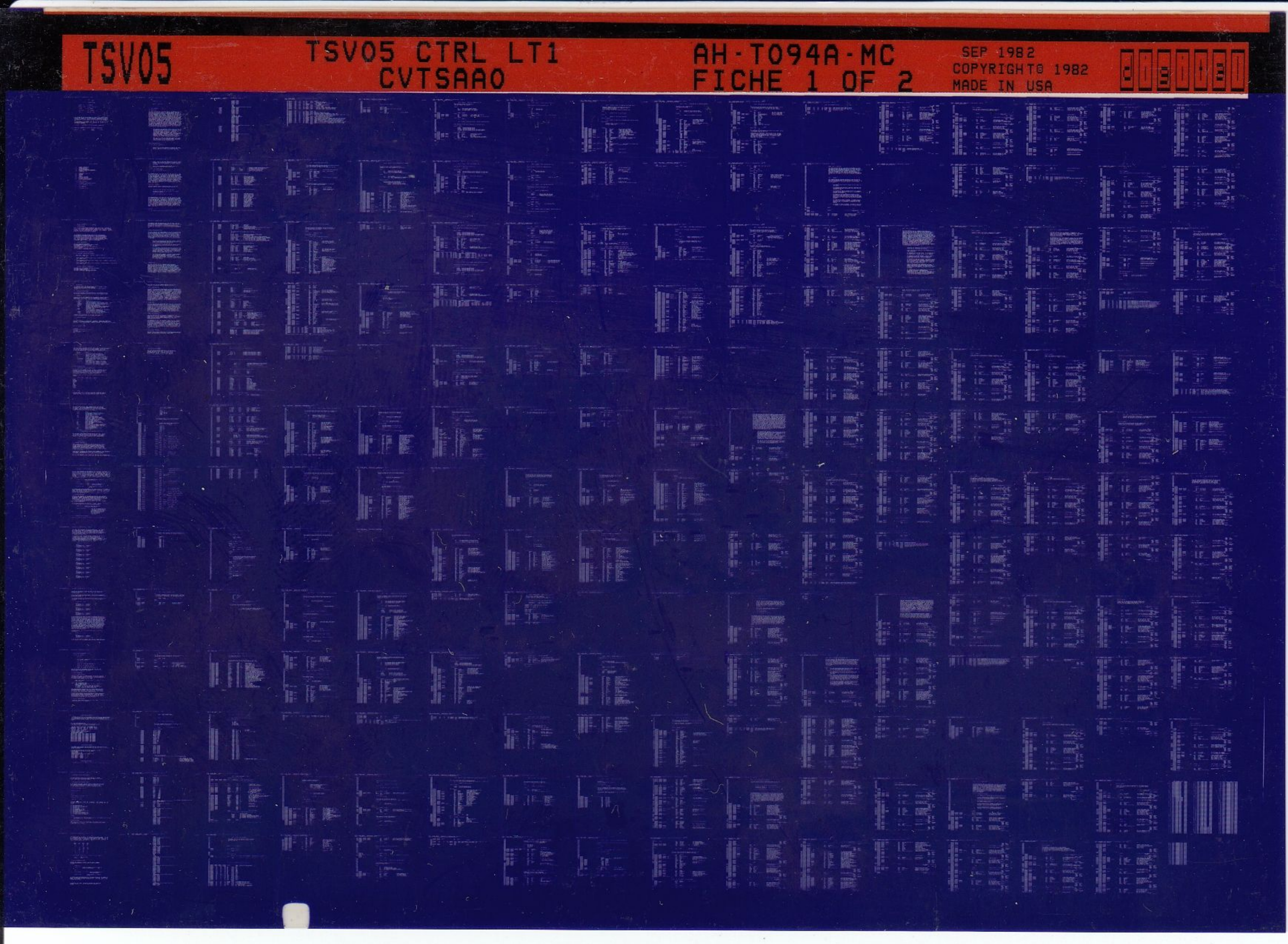
TSV05

TSV05 CTRL LT1
CVTSAA0

AH-T094A-MC
FICHE 1 OF 2



TSV05

TSV05 CTRL LT1
CVTSAA0

AH-T094A-MC
FICHE 2 OF 2

.REM_

IDENTIFICATION

PRODUCT ID: AC-T093A-MC

PRODUCT TITLE: CVTSAA0 TSV05 CTRL LT1

AUTHOR: DICK GORDON

MAINTAINER: SCOTT SNOWDON

DATE: MARCH 08, 1982

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS IS A PDP-11/23 RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF A TSV05 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A PDP-11//23 SYSTEM (Q-BUS). THE PROGRAM PROVIDES ERROR MESSAGES WHICH IDENTIFY FAILING FUNCTIONS THAT AID IN THE REPAIR OF THE DEVICE. THIS DIAGNOSTIC CONSIST OF 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

PDP-11/23 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 1982

3. TSV05 TRANSPORT SUBSYSTEM TECHNICAL MANUAL; DOCUMENT NUMBER EK-TSV05-TM-001
   DATE: AUGUST 1982

4. TSV05 TRANSPORT SUBSYSTEM INSTALLATION MANUAL; DOCUMENT NUMBER EK-TSV05-IN-001
   DATE: AUGUST 1982

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

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

1.5 ASSUMPTIONS

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

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

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

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

2.1.1 OPERATOR COMMANDS

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

BOOT THE DIAGNOSTIC MEDIA

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

2.2 SWITCHES

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

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

EXAMPLE OF SWITCH USAGE:

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

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

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

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

2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS

ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

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

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

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

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

2.4 HARDWARE QUESTIONS

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

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

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

```
        TSBA/TSDB = 172520, VECTOR = 224


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

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

        UNIT 0

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

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

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

2.5  SOFTWARE QUESTIONS

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

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

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

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

2.6  EXTENDED P-TABLE DIALOGUE

WHEN YOU ANSWER THE HARDWARE QUESTIONS, YOU ARE BUILDING ENTRIES
```

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

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

```
# UNITS (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: 100020  XOR: 000000
WORD #1 EXPD: 000012  RECV: 000012  XOR: 000000
WORD #2 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #3 EXPD: 000010  RECV: 000010  XOR: 000000
WORD #4 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #5 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #6 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #7 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #8 EXPD: 070217  RECV: 070217  XOR: 000000
WORD #9 EXPD: 000074  RECV: 000034  XOR: 000040
```

ERROR MESSAGE EXAMPLE 2

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

```
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 REWING

```
 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 (PDP-11/23)

 DR>STA/FLA:PNT:HOE

 UNITS (D) ?  1

 UNIT 0

 DEVICE ADDRESS  (O) 172520 ? <CR>

 VECTOR (O) 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 PDP-11/23
PROCESSOR WITH A LA34 CONSOLE.
```

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

TEST NUMBER	N/I SECS.	ITER SECS	DEF SECS.
1	1	30	29
2	1	10	9
3	1	8	7
4	25	120	95
5	5	140	135
6	25	475	450
7	20	20	0
8	1	10	9
9	20	20	0
10	1	2	1
11	8	11	3

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

Q.V. 1 MIN 57 SECONDS
DEFAULT 12 MINS

5.0 DEVICE INFORMATION TABLES

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

CHANGE HW (L) ?

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

UNIT 0

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

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

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

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

CHANGE SW (L) ?

INHIBIT ITERATIONS (L) N ?

6.0 TEST SUMMARIES

TEST 1: BUS RESET TEST

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

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

2. SSR = 0, SC = 0 AND THE ERROR CODE IN BITS 0-5 IS IN THE RANGE 17-13: THIS IS A FATAL ERROR. THE ERROR CODE IS DECODED AND THE APPROPRIATE DESCRIPTION GIVEN. INDICATES THAT A SERIOUS PROBLEM EXISTS.

TEST 2: WRAP DATA - HIGH BYTE

THIS TEST VERIFIES OPERATION OF:

1. PART OF THE 11/23 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 11/23 BUS INTERFACE SECTION OR IN THE ROM OR PIPELINE.

TEST 8: VOLUME CHECK

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

TEST 9: COMPLETION INTERRUPT

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

THE SEQUENCES OF TEST 7 ARE REPEATED, EXCEPT THAT THE INTERRUPT SERVICE ROUTINE IS SET UP TO EXPECT INTERRUPTS AND EACH WRITE CHARACTERISTICS COMMAND IS ISSUED WITH THE IE BIT SET (1). IT IS VERIFIED, WHERE APPROPRIATE, THAT THE IE STATUS BIT IN 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

```
                                        .TITLE  TSV2 - PROGRAM HEADER
                                        .SBTTL  PROGRAM HEADER

                                        .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,A,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      101                         .ASCII  /A/
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  045634                          .WORD   L$HARD
002020                          L$SPCP::                ;POINTER TO S.W. QUES.
002020  045766                          .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  046404                          .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
002040  002124                          .WORD   L$DISPATCH
```

```
002042
002042  000000                  L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
                                .WORD   0
002044
002044  000000                  L$ENVI::                ;FLAGS DESCRIBE HOW IT WAS SETUP
                                .WORD   0
002046
002046  000000                  L$EXP1::                ;EXPANSION WORD
                                .WORD   0
002050
002050    003                   L$MREV::                ;SVC REV AND EDIT #
                                .BYTE   C$REVISION
002051    003                   .BYTE   C$EDIT
002052
002052  000000                  L$EF::                  ;DIAG. EVENT FLAGS
                                .WORD   0
002054  000000                  .WORD   0
002056
002056  000000                  L$SPC::
                                .WORD   0
002060
002060  003376                  L$DEVP::                ; POINTER TO DEVICE TYPE LIST
                                .WORD   L$DVTYP
002062
002062  022654                  L$REPP::                ;PTR. TO REPORT CODE
                                .WORD   L$RPT
002064
002064  000000                  L$EXP4::
                                .WORD   0
002066
002066  000000                  L$EXP5::
                                .WORD   0
002070
002070  022342                  L$AUT::                 ;PTR. TO ADD UNIT CODE
                                .WORD   L$AU
002072
002072  022440                  L$DUT::                 ;PTR. TO DROP UNIT CODE
                                .WORD   L$DU
002074
002074  000000                  L$LUN::                 ;LUN FOR EXERCISERS TO FILL
                                .WORD   0
002076
002076  003404                  L$DESP::                ;POINTER TO DIAG. DESCRIPTION
                                .WORD   L$DESC
002100
002100  104035                  L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
                                EMT     E$LOAD
002102
002102  000000                  L$ETP::                 ;POINTER TO ERRTBL
                                .WORD   0
002104
002104  021546                  L$ICP::                 ;PTR. TO INIT CODE
                                .WORD   L$INIT
002106
002106  022626                  L$CCP::                 ;PTR. TO CLEAN-UP CODE
                                .WORD   L$CLEAN
002110
002110  022546                  L$ACP::                 ;PTR. TO AUTO CODE
                                .WORD   L$AUTO
002112
002112  021536                  L$PRT::                 ;PTR. TO PROTECT TABLE
                                .WORD   L$PROT
002114
002114  000000                  L$TEST::                ;TEST NUMBER
                                .WORD   0
002116
002116  000000                  L$DLY::                 ;DELAY COUNT
                                .WORD   0
002120
002120  000000                  L$HIME::                ;PTR. TO HIGH MEM
                                .WORD   0
```

```
                        .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  023436                  .WORD   T1
002126  023656                  .WORD   T2
002130  024354                  .WORD   T3
002132  025046                  .WORD   T4
002134  026402                  .WORD   T5
002136  027506                  .WORD   T6
002140  030764                  .WORD   T7
002142  034352                  .WORD   T8
002144  035256                  .WORD   T9
002146  040372                  .WORD   T10
002150  043474                  .WORD   T11

```

SEQ 0022

```
                                .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:
```

```
                                .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
```

```
                                .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
                        ;
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
34
35 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
;*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
```

```
                        .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                  XS0TMK= BIT15           ;TAPE MARK DETECTED
040000                  XS0RLS= BIT14           ;RECORD LENGTH SHORT
020000                  XS0LET= BIT13           ;LOGICAL END OF TAPE
010000                  XS0RLL= BIT12           ;RECORD LENGTH LONG
004000                  XS0WLE= BIT11           ;WRITE LOCK ERROR
002000                  XS0NEF= BIT10           ;NON EXECUTABLE FUNCTION
001000                  XS0ILC= BIT9            ;ILLEGAL COMMAND
000400                  XS0ILA= BIT8            ;ILLEGAL ADDRESS
000200                  XS0MOT= BIT7            ;TAPE IN MOTION
000100                  XS0ONL= BIT6            ;TRANSPORT ON LINE
000040                  XS0IE=  BIT5            ;INTERRUPT ENABLE
000020                  XS0VCK= BIT4            ;VOLUME CHECK BIT
000010                  XS0PED= BIT3            ;PHASE ENCODED DRIVE
000004                  XS0WLK= BIT2            ;WRITE LOCKED
000002                  XS0BOT= BIT1            ;BEGINNING OF TAPE
000001                  XS0EOT= BIT0            ;END OF TAPE


                        ;+
                        ;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)
                        ;
                        ;-

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

                        ;+
                        ;
                        ;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

                        ;+
                        ; 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.WNPR         = 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          ;ITAD0  - TRANSPORT ADDRESS BIT 0
```

```
000040          WC.I1TAD        = BIT5          ;ITAD1  - TRANSPORT ADDRESS BIT 1
000020          WC.I5RESV       = 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
                ;+
                ; 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
```

```
325          004000          S1.IIDENT       = BIT11         ;                IIDENT H
326          002000          S1.ICER         = BIT10         ;                ICER H
327          001000          S1.IFMK         = BIT9          ;                IFMK H
328          000400          S1.IHER         = BIT8          ;                IHER H
329          000200          S0.ISPEED       = BIT7          ;WORD #8 BYTE 0  ISPEED H
330          000100          S0.IRDY         = BIT6          ;                IRDY L
331          000040          S0.IONL         = BIT5          ;                IONL L
332          000020          S0.ILDP         = BIT4          ;                ILDP L
333          000010          S0.IDBY         = BIT3          ;                IDBY L
334          000004          S0.IRWD         = BIT2          ;                IRWD L
335          000002          S0.IFBY         = BIT1          ;                IFBY L
336          000001          S0.IFPT         = BIT0          ;                IFPT L
337
338
```

```
                        .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

                ;+
```

```
                                ;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.
```

```
                                .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          BENBSW::        .WORD   0       ;BUFFER ENABLE SWITCH SW 0=OFF;1=ON
002230  000000          EXPD::          .WORD   0       ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
002232  000000          RECV::          .WORD   0       ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
002234  000000          ERRHI::         .WORD   0       ;HIGH ADDRESS MEMORY ERROR
002236  000000          ERRLO::         .WORD   0       ;LOW ADDRESS MEMORY ERROR
002240                  RAMDATA::       .BLKW   16.     ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
002300  000000          RAMSIZ::        .WORD   0       ;RAM DATA SIZE FOR PRAMPKT ROUTINE
002302  000000          RCVHIADD::      .WORD   0       ;RECEIVED BUFFER HIGH ADDRESS
002304  000000          RCVLOADD::      .WORD   0       ;RECEIVED BUFFER LOW ADDRESS
002306  000000          COUNT::         .WORD   0       ;TEST COUNT PATTERN
002310  000000          DATA::          .WORD   0       ;TEST DATA
002312  000000          TSTFLAG::       .WORD   0       ;TEST FLAG WORD
002314  000000          TSTPTR::        .WORD   0       ;TSTBLK POINTER
002316  000000          PRMNO::         .WORD   0       ;PRINT ROUTINE TEMP
002320                  EXPMSG::        .BLKB   100.    ;EXPECTED MESSAGE BUFFER DATA
002464                  RECMSG::        .BLKB   100.    ;RECEIVED MESSAGE BUFFER DATA
002630                  TMPBFR::        .BLKB   80.     ;TEMPORARY STORAGE FOR PRINT
```

```
                                .SBTTL  TSTBLK  - TEST DATA TABLE

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

002750                  TSTBLK::
002750  000000                  .WORD   0               ;ALL ZEROS
002752  177777                  .WORD   177777          ;ALL ONES
002754  000001                  .WORD   BIT0            ;DATA FOR WALKING ONES
002756  000002                  .WORD   BIT1
002760  000004                  .WORD   BIT2
002762  000010                  .WORD   BIT3
002764  000020                  .WORD   BIT4
002766  000040                  .WORD   BIT5
002770  000100                  .WORD   BIT6
002772  000200                  .WORD   BIT7
002774  000400                  .WORD   BIT8
002776  001000                  .WORD   BIT9
003000  002000                  .WORD   BIT10
003002  004000                  .WORD   BIT11
003004  010000                  .WORD   BIT12
003006  020000                  .WORD   BIT13
003010  040000                  .WORD   BIT14
003012  100000                  .WORD   BIT15
003014  177776                  .WORD   ^CBIT0          ;DATA FOR WALKING ZEROS
003016  177775                  .WORD   ^CBIT1
003020  177773                  .WORD   ^CBIT2
003022  177767                  .WORD   ^CBIT3
003024  177737                  .WORD   ^CBIT5
003026  177677                  .WORD   ^CBIT6
003030  177577                  .WORD   ^CBIT7
003032  177377                  .WORD   ^CBIT8
003034  176777                  .WORD   ^CBIT9
003036  175777                  .WORD   ^CBIT10
003040  173777                  .WORD   ^CBIT11
003042  167777                  .WORD   ^CBIT12
003044  157777                  .WORD   ^CBIT13
003046  137777                  .WORD   ^CBIT14
003050  077777                  .WORD   ^CBIT15
003052  125252                  .WORD   125252          ;ALTERNATING ONES, ZEROS
003054  052525                  .WORD   052525          ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
003056          TBLEND==.
```

```
                                        .SBTTL  GLOBAL ENVIRONMENT STORAGE
                                ;
                                ;STORAGE FOR DEVICE REGISTERS
                                ;
003056  000000  100000  000000  DUMMY:  0,100000,0,0    ;DUMMY DEVICE REGISTERS...
003066  000000  000000  000000          0,0,0,0,0,0,0,0
                                                        ;...FOR MULTI-UNIT CHECKOUT.


003106  000000                  DUFLG::         .WORD   0               ;"DROPPED UNIT" FLAG.
                                                                        ;INHIBITS CODE IN "CLEAN-UP".
003110  000000                  NODEV::         .WORD   0               ;FLAG TO SAY NO DEVICE.

003112  000000                  TEMP1::         .WORD   0               ;SOME TEMP LOCATIONS.
003114  000000                  TEMP2::         .WORD   0
003116  000000                  XXCOMM::        .WORD   0               ;XXDP+ COMM BLOCK POINTER.
003120  000000                  FREE::          .WORD   0               ;1ST FREE MEMORY ADDRESS...
003122  000000                  FRESIZ::        .WORD   0               ;...AND SIZE (IN WORDS).
003124  000000                  FREEHI: .WORD   0               ;LAST WORD IN FREE SPACE
003126  000000                  KTFLG::         .WORD   0               ;KT11, MEM AVAIL FLAG -
                                                                        ;- .WORD        0 = <24K OR NO KT -
                                                                        ;- NZ = >24K AND KT.
003130  000000                  KTENABLE::      .WORD   0               ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
003132  000000                  NXMFLG::        .WORD   0               ;SET IF WE CAN TEST CLEARED OTHERWISE
003134  000000                  NXMLO::         .WORD   0               ;NXM LO ADDRESS BITS
003136  000000                  NXMHI::         .WORD   0               ;NXM HI ADDRESS BITS FOR DAL'S 16-21
003140  000000                  T23A::          .WORD   0               ;11/23A FLAG
003142  000000                  T23B::          .WORD   0               ;11/23B FLAG
003144  000000                  T3BFLG::        .WORD   0               ;TEST 3B FLAG ^O
003146  002000                  PST32W::        .WORD   2000            ;32W BLOCK ADDRESS FOR 32K START
003150  000000                  SIFLAG::        .WORD   0               ;
003152  000000                  BADDAT::        .WORD   0               ;ACTUAL DATA
003154  000000                  GDDAT::         .WORD   0               ;EXPECTED DATA
003156  000000                  LOOPFL::        .WORD   0
003160                          CTAB::                                  ;CONFIGURATION TABLES.
003160  000000                  CTABM::         .WORD   0               ;CONFIG WORK.
003162  000000                                  .WORD   0
003164  000000                                  .WORD   0
003166  000000                                  .WORD   0
003170  177777                                  .WORD   -1              ;END OF MEM TABLE.
003172                          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
                                ;
003172                          ERTABL:         .BLKW   64.
003372  000000                  ERTABE:         .WORD   0

003374  000000                  SKIPT:  .WORD   0                ;1=SKIP SUBTEST 0=NO SKIP OF SUBTEST
```

```
                                        .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
                                ;-

003376                                  DEVTYP  <TSV05>
003376                          L$DVTYP::
003376  124  123  126                    .ASCIZ  /TSV05/
                                        .EVEN

                                ;+
                                ;TEST DESCRIPTION
                                ;-
003404                                  DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****>
003404                          L$DESC::
003404  052  052  052                    .ASCIZ  /**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****/
                                        .EVEN



                                ;+
                                ;BIT TO ASCII CONVERSION FOR TSSR REGISTER
                                ;-

003476  003536  003541  003545  TSSRBIT::        .WORD   1$,2$,3$,4$,5$,6$,7$,8$
003516  003577  003603  003607                  .WORD   9$,10$,11$,12$,13$,14$,15$,16$
003536     123     103     000  1$:     .ASCIZ  'SC'
003541     102     111     105  2$:     .ASCIZ  'BIE'
003545     123     103     105  3$:     .ASCIZ  'SCE'
003551     122     115     122  4$:     .ASCIZ  'RMR'
003555     116     130     115  5$:     .ASCIZ  'NXM'
003561     116     102     101  6$:     .ASCIZ  'NBA'
003565     102     111     124  7$:     .ASCIZ  'BIT9'
003572     102     111     124  8$:     .ASCIZ  'BIT8'
003577     123     123     122  9$:     .ASCIZ  'SSR'
003603     117     106     114  10$:    .ASCIZ  'OFL'
003607     102     111     124  11$:    .ASCIZ  'BIT5'
003614     102     111     124  12$:    .ASCIZ  'BIT4'
003621     102     111     124  13$:    .ASCIZ  'BIT3'
003626     102     111     124  14$:    .ASCIZ  'BIT2'
003633     102     111     124  15$:    .ASCIZ  'BIT1'
003640     102     111     124  16$:    .ASCIZ  'BIT0'
                                        .EVEN
003646     124     123     123  SFIERR: .ASCIZ  'TSSR ERROR AFTER SOFT INIT'
003701     124     123     123  SFHERR: .ASCIZ  'TSSR ERROR AFTER BUS RESET'
003734     040     040     116  NXR:    .ASCIZ  /  NON-EXISTANT DEVICE REGISTER/
003773     045     101     040  NXRX:   .ASCIZ  /%A  ADDRESS: %O6/
004014     045     101     040  TSSX:   .ASCII  /%A  TSBA,TSSR EXP'D: %O6%A,%O6%N/
004054     045     101     040          .ASCIZ  /%A  TSBA,TSSR REC'D: %O6%A,%O6/
```

```
                                        .ASCII  /%N%A/
004113  045  116  045  FUSI:    .ASCIZ  /%N%A/
004117  040  040  125  USI:     .ASCIZ  /  UNEXPECTED INTERRUPT/
004146  040  040  111  NSI:     .ASCIZ  /  INTERRUPT EXPECTED, NOT RECEIVED/
004211  045  116  045  FNOINTR:          .ASCII  /%N%A/
004215  040  040  116  NOINTR:  .ASCIZ  /  NO INTERRUPT WAS GENERATED/
004252  040  040  111  IFAULT:  .ASCIZ  /  INTERRUPT FAULT/
004274  045  101  040  INTX:    .ASCIZ  /%A  CPU PC: %O6%A  TSBA: %O6/
004331  040  040  042  NOINIT:  .ASCIZ  /  "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
004403  040  040  042  NSINIT:  .ASCIZ  /  "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
004453  040  040  042  BRINIT:  .ASCIZ  /  "BUS-RESET" DIDN'T INITIALIZE THE DPU/

004523  000          NUL:     .ASCIZ  //
004524  045  116  000  NULCR:   .ASCIZ  /%N/
004527  045  101  040  EXPGOT:  .ASCIZ  /%A EXP'D: %O6%A, REC'D: %O6/
004563  045  116  045  EXPGT2:  .ASCIZ  /%N%A EXP'D: %O6%A, %O6%N%A REC'D: %O%A, %O6/
004637  045  101  040  DUAD12:  .ASCIZ  /%A REG(W) WRITTEN TO: %O6%A REG(R) READ; EXP'D: %O6%A, REC'D: %O6/
004741  122  101  115  PKTRAM::         .ASCIZ  'RAM Contents Do Not Match Packet Sent'
005007  040  040  103  SCME:    .ASCIZ  /  CONFIG DOESN'T MATCH MFG. MASTER/
005052  127  122  111  WRTMSG:  .ASCIZ  'WRITE CHARACTERISTICS Failed'
005107  124  123  123  WRTERR:  .ASCIZ  'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
005202  124  123  123  RDERR:   .ASCIZ  'TSSR Incorrect After READ Command, More Bits Set Than SSR'
005274  106  101  124  SCHERR:  .ASCIZ  'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
005366  105  122  122  RETERR:  .ASCIZ  'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
005454  045  116  045  NOMEM:   .ASCIZ  '%N%A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****%N'
005550  045  116  045  M8186:   .ASCIZ  '%N%A ****************** 11/23A SYSTEM ******************%N'
005641  045  116  045  M8189:   .ASCIZ  '%N%A ****************** 11/23B SYSTEM ******************%N'
                                .EVEN
```

```
                        .SBTTL  GLOBAL ERROR REPORT SECTION

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

005732                          BGNMSG  NXRERR          ;NON-EXISTANT DEVICE REGISTER.
005732                  NXRERR::
005732                          PRINTX  #NXRX,NODEV     ;NODEV = NEXM ADDRESS.
005732  013746  003110          MOV     NODEV,-(SP)
005736  012746  003773          MOV     #NXRX,-(SP)
005742  012746  000002          MOV     #2,-(SP)
005746  010600                  MOV     SP,R0
005750  104415                  TRAP    C$PNTX
005752  062706  000006          ADD     #6,SP
005756  004737  005764          JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
005762                          ENDMSG
005762                  L10002:
005762  104423                  TRAP    C$MSG


                        ;
                        ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
                        ; TO ANY OF THE ABOVE ERROR SIGNATURES.
                        ;
005764  005727         EXTEND: TST     (PC)+
005766  000000         EXTA:   0                       ; 0 = NO EXTENSION.
005770  001402                 BEQ     1$
005772  004777  177770         JSR     PC,@EXTA        ; APPEND EXTENSION TEXT.
005776                  1$:     PRINTX  #NULCR          ; PRINT A BLANK LINE
005776  012746  004524          MOV     #NULCR,-(SP)
006002  012746  000001          MOV     #1,-(SP)
006006  010600                  MOV     SP,R0
006010  104415                  TRAP    C$PNTX
006012  062706  000004          ADD     #4,SP
006016  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
                ;
                ;-

006020          PRITSSR:
006020                  SAVREG                          ;SAVE GENERAL REGISTERS
006024  010104          MOV     R1,R4                   ;SAVE THE TSSR CONTENTS
006026                  PRINTB  #TSSRFOR,R4             ;PRINT THE CONTENTS OF TSSR
006026  010446          MOV     R4,-(SP)
006030  012746  006411  MOV     #TSSRFOR,-(SP)
006034  012746  000002  MOV     #2,-(SP)
006040  010600          MOV     SP,R0
006042  104414          TRAP    C$PNTB
006044  062706  000006  ADD     #6,SP
006050  010400          MOV     R4,R0                   ;GET TSSR BACK FOR CHKAMB
006052  004737  016034  JSR     PC,CHKAMB               ;ARE CONTENTS AMBIGUOUS ?
006056  103410          BCS     5$                      ;BRANCH IF NOT
006060                  PRINTX  #AMBTSSR                ;SHOW CONTENTS ARE AMBIGUOUS
006060  012746  006631  MOV     #AMBTSSR,-(SP)
006064  012746  000001  MOV     #1,-(SP)
006070  010600          MOV     SP,R0
006072  104415          TRAP    C$PNTX
006074  062706  000004  ADD     #4,SP
006100  010403  5$:     MOV     R4,R3                   ;CONTENTS OF TSSR
006102  042703  001476  BIC     #HIADDR!FATERR!TERCLS,R3        ;CLEAR ALL MULTIPLE BIT FIELDS
006106  001434          BEQ     20$                     ;NO BITS ARE SET
006110  012702  002630  MOV     #TMPBFR,R2              ;TEMPORARY ASCII BUFFER
006114  012701  003476  MOV     #TSSRBIT,R1             ;ASCII EQUIVALENT OF BITS
006120  005703  10$:    TST     R3                      ;REMAINING BITS TO CONVERT
006122  001413          BEQ     15$                     ;BRANCH WHEN ALL ARE DONE
006124  000241          CLC                             ;CLEAR CARRY FOR SHIFT
006126  006103          ROL     R3                      ;SHIFT NEXT BIT TO CARRY
006130  103006          BCC     13$                     ;BRANCH IF BIT NOT SET
006132  011100          MOV     (R1),R0                 ;POINTER TO BIT DEFINITION
006134  112022  11$:     MOVB    (R0)+,(R2)+             ;MOVE ASCIZ TO BUFFER
006136  001376          BNE     11$                     ;MOVE ALL BITS
006140  112762  000054  177777  MOVB    #',,-1(R2)      ;INSERT A COMMA TO TERMINATE
006146  005721  13$:     TST     (R1)+                   ;POINT TO NEXT DESCRIPTION
006150  000763          BR      10$                     ;GET THE REMAINING BITS
006152  105042  15$:     CLRB    -(R2)                   ;TERMINATE THE LINE
006154                  PRINTX  #TSSDEF,#TMPBFR         ;PRINT THE BIT DEFINITIONS
006154  012746  002630  MOV     #TMPBFR,-(SP)
006160  012746  006602  MOV     #TSSDEF,-(SP)
```

```
                006164  012746  000002                  MOV     #2,-(SP)
                006170  010600                          MOV     SP,R0
                006172  104415                          TRAP    C$PNTX
                006174  062706  000006                  ADD     #6,SP
    745
    746         006200  010403              20$:        MOV     R4,R3           ;GET THE TSSR CONTENTS
    747         006202  042703  177761                  BIC     #^CTERCLS,R3    ;CLEAR ALL BUT TERMINATION
    748         006206  016303  006672                  MOV     TCOCOD(R3),R3   ;GET THE TERMINATION CODE MEANING
    749         006212                                  PRINTX  #TCOASC,R3      ;PRINT THE TERMINATION CODE
                006212  010346                          MOV     R3,-(SP)
                006214  012746  006472                  MOV     #TCOASC,-(SP)
                006220  012746  000002                  MOV     #2,-(SP)
                006224  010600                          MOV     SP,R0
                006226  104415                          TRAP    C$PNTX
                006230  062706  000006                  ADD     #6,SP
    750         006234  010403                          MOV     R4,R3           ;TSSR CONTENTS AGAIN
    751         006236  042703  177717                  BIC     #^CFATERR,R3    ;CLEAR ALL BUT FATAL TERMINATION
    752         006242  001416                          BEQ     25$             ;DON'T PRINT IF ZERO
    753         006244  006203                          ASR     R3
    754         006246  006203                          ASR     R3
    755         006250  006203                          ASR     R3              ;ALINE TERMINATION CODE FOR INDEX
    756         006252  016303  007232                  MOV     TSFCOD(R3),R3   ;GET THE FATAL TERMINATION CODE
    757         006256                                  PRINTX  #TFCASC,R3      ;PRINT THE FATAL TERMINATION CODE
                006256  010346                          MOV     R3,-(SP)
                006260  012746  006533                  MOV     #TFCASC,-(SP)
                006264  012746  000002                  MOV     #2,-(SP)
                006270  010600                          MOV     SP,R0
                006272  104415                          TRAP    C$PNTX
                006274  062706  000006                  ADD     #6,SP
    758         006300  042704  176377      25$:        BIC     #^CHIADDR,R4    ;CLEAR ALL BUT EXTENDED ADDRESS
    759         006304  001411                          BEQ     30$             ;DON'T PRINT IF ZERO
    760         006306                                  PRINTX  #TEXASC,R4      ;PRINT THE EXTENDED ADDRESS BITS
                006306  010446                          MOV     R4,-(SP)
                006310  012746  006431                  MOV     #TEXASC,-(SP)
                006314  012746  000002                  MOV     #2,-(SP)
                006320  010600                          MOV     SP,R0
                006322  104415                          TRAP    C$PNTX
                006324  062706  000006                  ADD     #6,SP
    761         006330  013703  002176      30$:        MOV     EPRTSW,R3               ;PRINT MEAASGE BUFFER ADDRESS
    762         006334                                  PRINTX  R3                      ;PRINT PROPER MESSAGE
                006334  010346                          MOV     R3,-(SP)
                006336  012746  000001                  MOV     #1,-(SP)
                006342  010600                          MOV     SP,R0
                006344  104415                          TRAP    C$PNTX
                006346  062706  000004                  ADD     #4,SP
    763         006352  000207                          RTS     PC              ;RETURN TO CALLER
    764
    766         006354                      EPRT2:
    767         006354      045     116     045 EPRT1:  .ASCIZ   '%N%A *****REPLACE M7196*****'
    782         006411      045     116     045 TSSRFOR:        .ASCIZ   '%N%A TSSR = %O6'
    783         006431      045     116     045 TEXASC:         .ASCIZ   '%N%A Extended Address Bits = %O6'
    784         006472      045     116     045 TCOASC:         .ASCIZ   '%N%A Termination Class Code = %T'
    785         006533      045     116     045 TFCASC:         .ASCIZ   '%N%A Fatal Termination Class Code = %T'
    786         006602      045     116     045 TSSDEF:         .ASCIZ   '%N%A TSSR Bits Set: %T'
    787         006631      045     116     045 AMBTSSR:        .ASCIZ   '%N%A TSSR Contents Are Ambiguous'
    788                                             .EVEN
    789         006672  006712  006735  006763 TCOCOD:  .WORD   1$,2$,3$,4$,5$,6$,7$,8$
```

```
 790 006712  116  157  162  1$:     .ASCIZ  'Normal Termination'
 791 006735  124  145  162  2$:     .ASCIZ  'Termination Condition'
 792 006763  124  141  160  3$:     .ASCIZ  'Tape Status Alert'
 793 007005  106  165  156  4$:     .ASCIZ  'Function Reject'
 794 007025  122  145  143  5$:     .ASCIZ  'Recoverable Error - Tape Position One Record Down'
 795 007107  122  145  143  6$:     .ASCIZ  'Recoverable Error - Tape Was Not Moved'
 796 007156  125  156  162  7$:     .ASCIZ  'Unrecoverable Error'
 797 007202  106  141  164  8$:     .ASCIZ  'Fatal Controller Error'
 798                                 .EVEN
 799
 800 007232 007242 007276 007307 TSFCOD: .WORD   1$,2$,3$,4$
 801 007242  111  156  164  1$:     .ASCIZ  'Internal Diagnostic Failure'
 802 007276  122  145  163  2$:     .ASCIZ  'Reserved'
 803 007307  102  165  163  3$:     .ASCIZ  'Bus Interface or Sanity Check Error'
 804 007353  122  145  163  4$:     .ASCIZ  'Reserved'
 805                                 .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.
                ;-

007364          PRIPKT::
007364                  SAVREG                          ;SAVE THE REGISTERS
007370  010005          MOV     R0,R5                   ;SAVE NO. OF WORDS IN PACKET
007372  005737  003130  TST     KTENABLE                ;ABOVE 28K UNDER TEST?
007376  001001          BNE     10$                     ;BR IF YES
007400  005003          CLR     R3                      ;SET HIGH ORDER ADDRESS TO 0
007402  010301  10$:    MOV     R3,R1                   ;COPY HIGH ORDER ADDRESS
007404  010400          MOV     R4,R0                   ;GET LOWER ADDRESS
007406  006100          ROL     R0                      ;SHIFT BIT 15 INTO C BIT
007410  006101          ROL     R1                      ;AND INTO HIGH ORDER.
007412                  PRINTB  #PKTADD,R1,R4           ;PRINT PACKET ADDRESS
007412  010446          MOV     R4,-(SP)
007414  010146          MOV     R1,-(SP)
007416  012746  007550  MOV     #PKTADD,-(SP)
007422  012746  000003  MOV     #3,-(SP)
007426  010600          MOV     SP,R0
007430  104414          TRAP    C$PNTB
007432  062706  000010  ADD     #10,SP
007436  010300  15$:    MOV     R3,R0                   ;GET HIGH ORDER ADDRESS
007440  001404          BEQ     20$                     ;BR IF NOT ABOVE 28K.
007442  010401          MOV     R4,R1                   ;GET LOW ORDER ADDRESS
007444  004737  017306  JSR     PC,SETMAP               ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
007450  010004          MOV     R0,R4                   ;GET RETURNED PAR6 ADDRESS BIAS
007452  005001  20$:    CLR     R1                      ;SAVE WORD NUMBER
007454  012402  25$:    MOV     (R4)+,R2                ;GET PACKET CONTENTS
007456                  PRINTB  #PKTFRM,R1,R2           ;PRINT THE DATA
007456  010246          MOV     R2,-(SP)
007460  010146          MOV     R1,-(SP)
007462  012746  007512  MOV     #PKTFRM,-(SP)
007466  012746  000003  MOV     #3,-(SP)
007472  010600          MOV     SP,R0
007474  104414          TRAP    C$PNTB
007476  062706  000010  ADD     #10,SP
007502  005201          INC     R1                      ;NEXT WORD NUMBER
007504  020105          CMP     R1,R5                   ;DONE ALL PACKET WORDS?
007506  002762          BLT     25$                     ;LOOP TILL ALL DONE
007510  000207          RTS     PC                      ;RETURN

007512  045  116  045  PKTFRM: .ASCIZ  '%N%A Packet Word #%D1%A = %O6'
007550  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
                ;
                ;-

007606          PRIBXOR::
007606                  SAVREG                          ;SAVE THE REGISTERS
007612  010203          MOV     R2,R3                   ;EXPECTED DATA
007614                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007624  012700  177400  MOV     #^C<377>,R0             ;BYTE MASK
007630  040001          BIC     R0,R1                   ;SAVE LOW BYTE RECV
007632  040002          BIC     R0,R2                   ;SAVE LOW BYTE EXPD
007634  040003          BIC     R0,R3                   ;SAVE LOW BYTE XOR
007636                  PRINTB  #XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
007636  010346          MOV     R3,-(SP)
007640  010146          MOV     R1,-(SP)
007642  010246          MOV     R2,-(SP)
007644  012746  007670  MOV     #XORBFOR,-(SP)
007650  012746  000004  MOV     #4,-(SP)
007654  010600          MOV     SP,R0
007656  104414          TRAP    C$PNTB
007660  062706  000012  ADD     #12,SP
007664  010300          MOV     R3,R0                   ;R0 HAS XOR ON RETURN
007666  000207          RTS     PC                      ;RETURN TO CALLER

007670      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
                        ;
                        ;-

007736                  PRIXOR::
007736                          SAVREG                          ;SAVE THE REGISTERS
007742  010203                  MOV     R2,R3                   ;EXPECTED DATA
007744                          XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007754                          PRINTB  #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
007754  010346                  MOV     R3,-(SP)
007756  010146                  MOV     R1,-(SP)
007760  010246                  MOV     R2,-(SP)
007762  012746  010006          MOV     #XORFOR,-(SP)
007766  012746  000004          MOV     #4,-(SP)
007772  010600                  MOV     SP,R0
007774  104414                  TRAP    C$PNTB
007776  062706  000012          ADD     #12,SP
010002  010300                  MOV     R3,R0                   ;R0 HAS XOR ON RETURN
010004  000207                  RTS     PC                      ;RETURN TO CALLER

010006      045     116     045 XORFOR: .ASCIZ   '%N%A EXPD: %06%A RECV: %06%A XOR: %06'
                                .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
                                ;
                                ;-

010054                          PRIEQU:
010054                                  SAVREG                          ;SAVE THE REGISTERS
010060  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
                                ;
                                ;-
010062                          PRIRAM:
010062                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
010066                                  PRINTB  #RAMFOR,R4              ;PRINT RAM ADDRESS IN ERROR
010066  010446                          MOV     R4,-(SP)
010070  012746  010112                  MOV     #RAMFOR,-(SP)
010074  012746  000002                  MOV     #2,-(SP)
010100  010600                          MOV     SP,R0
010102  104414                          TRAP    C$PNTB
010104  062706  000006                  ADD     #6,SP
010110  000207                          RTS     PC                      ;RETURN

010112      045     116     045 RAMFOR: .ASCIZ  '%N%A CONTROLLER RAM ADDRESS = %06'
                                        .EVEN


                                        .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
```

```
                                        ;
                                        ;-
010154                                  PRIADD:
010154                                          SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
010160  013700  002234                          MOV     ERRHI,R0        ;GET HIGH ADDRESSS
010164  013701  002236                          MOV     ERRLO,R1        ;GET LOW ADDRESS
010170  010102                                  MOV     R1,R2           ;COPY LOW ADDRESS
010172  006101                                  ROL     R1              ;SHIFT BIT 15 TO C BIT
010174  006100                                  ROL     R0              ;SHIFT INTO HIGH ORDER
010176                                          PRINTB  #PRIA0,R0,R2    ;PRINT MEMORY ADDRESS IN ERROR
010176  010246                                  MOV     R2,-(SP)
010200  010046                                  MOV     R0,-(SP)
010202  012746  010224                          MOV     #PRIA0,-(SP)
010206  012746  000003                          MOV     #3,-(SP)
010212  010600                                  MOV     SP,R0
010214  104414                                  TRAP    C$PNTB
010216  062706  000010                          ADD     #10,SP
010222  000207                                  RTS     PC              ;RETURN

010224      045     116     045 PRIA0:  .ASCIZ  '%N%A MEMORY ERROR ADDRESS = %01%05'
                                    .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
                                        ;
                                        ;-
010270                                  PRITADD:
010270                                          SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
010274  013702  002234                          MOV     ERRHI,R2        ;GET HIGH ADDRESSS
010300  013701  002236                          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
010304                                          PRINTB  #PRIT0,R1       ;PRINT MEMORY ADDRESS LOW IN ERROR
010304  010146                                  MOV     R1,-(SP)
010306  012746  010352                          MOV     #PRIT0,-(SP)
010312  012746  000002                          MOV     #2,-(SP)
010316  010600                                  MOV     SP,R0
010320  104414                                  TRAP    C$PNTB
010322  062706  000006                          ADD     #6,SP
010326                                          PRINTB  #PRIT1,R2       ;PRINT MEMORY ADDRESS HIGH IN ERROR
010326  010246                                  MOV     R2,-(SP)
010330  012746  010415                          MOV     #PRIT1,-(SP)
010334  012746  000002                          MOV     #2,-(SP)
010340  010600                                  MOV     SP,R0
010342  104414                                  TRAP    C$PNTB
010344  062706  000006                          ADD     #6,SP
010350  000207                                  RTS     PC              ;RETURN
```

```
   1003
   1004 010352     045     116     045  PRIT0:  .ASCIZ  '%N%A MEMORY TEST ADDRESS LOW = %06'
   1005 010415     045     116     045  PRIT1:  .ASCIZ  '%N%A MEMORY TEST ADDRESS HIGH = %06'
   1006                                         .EVEN
   1007
   1008
   1009
```

```
                        .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:
                ;
                ;
                ;-

010462                  SPACE::
010462                          SAVREG                          ;SAVE THE GENERAL REGISTERS
010466  012737  000764  010650          MOV     #500.,SDELAY    ;SET UP DELAY
010474  012737  140010  010640          MOV     #140010,80$     ;SET UP COMMAND, SPACE FORWARD
010502  005703                          TST     R3              ;CHECK FOR DIRECTION
010504  100403                          BMI     5$              ;BR, IF REVERSE INDICATED
010506  010337  010642                  MOV     R3,90$          ;LOAD UP NUMBER OF RECORDS TO SPACE
010512  000407                          BR      10$             ;GO DO COMMAND
010514  042703  100000          5$:     BIC     #BIT15,R3       ;CLEAR DIRECTION BIT
010520  010337  010642                  MOV     R3,90$          ;LOAD UP NUMBER OF RECORDS TO SPACE
010524  052737  000400  010640          BIS     #BIT8,80$       ;SET REVERSE BIT IN COMMAND PACKET
010532  012704  010640          10$:    MOV     #80$,R4         ;SET UP R4 WITH PACKET ADDRESS
010536  010465  000000                  MOV     R4,TSDB(R5)     ;SEND OUT COMMAND
010542  004737  016240          15$:    JSR     PC,WAITF        ;WAIT FOR SSR
010546  103420                          BCS     20$             ;BR, IF SSR IS SET AND OK
010550                                  DELAY   250     ;DELAY ABOUT .25 SECONDS
010550  012727  000250                  MOV     #250.,(PC)+
010554  000000                          .WORD   0
010556  013727  002116                  MOV     L$DLY,(PC)+
010562  000000                          .WORD   0
010564  005367  177772                  DEC     -6(PC)
010570  001375                          BNE     .-4
```

```
        010572  005367  177756                  DEC     -22(PC)
        010576  001367                          BNE     .-20
   1062 010600  005337  010650                  DEC     $DELAY          ;BUMP DELAY COUNTER DOWN
   1063 010604  001356                          BNE     15$             ;BR, IF MORE DELAY
   1064 010606  000411                          BR      60$             ;BR IF TROUBLE CARRY = CLEAR
   1065 010610  016501  000002          20$:    MOV     TSSR(R5),R1     ;READ TSSR
   1066 010614  012702  000200                  MOV     #SSR,R2         ;SET UP EXPECTED
   1067 010620  020201                  25$:    CMP     R2,R1           ;ARE THEY OK
   1068 010622  001401                          BEQ     40$             ;BR, IF EQUAL = OK
   1069 010624  000402                          BR      60$             ;TROUBLE EXIT
   1070 010626  000261                  40$:    SEC                     ;SET CARRY NO TROUBLE
   1071 010630  000401                          BR      70$             ;EXIT
   1072 010632  000241                  60$:    CLC                     ;CARRY CLEAR = ERROR
   1073 010634                          70$:
   1074 010634  010400                          MOV     R4,R0           ;PASS PACKET ADDRESS
   1075 010636  000207                          RTS     PC              ;RETURN
```

```
                                        ;
                                        ;
                                        ;
                                        ;PACKET FOR SPACE COMMAND
                                        ;
                                        ;
                                        ;COMMAND WORD
1087 010640  000000                     80$:    .WORD
                                        ;NUMBER OF RECORDS TO BE SPACED OVER WORD
1089 010642  000000                     90$:    .WORD
1090 010644  000000                             .WORD
1091 010646  000000                             .WORD
1092 010650  000000                     SDELAY: .WORD   0                       ;DELAY COUNTER
                                                .EVEN
```

```
                        .SBTTL  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:
                ;
                ;-

010652                  WRTCHR::
010652                          SAVREG                          ;SAVE THE GENERAL REGISTERS
010656  005037  002226          CLR     BENBSW          ;CLEAR BUFFER ENABLE SWITCH
010662  005037  002224          CLR     EXTFEA                  ;CLEAR EXTENDED FEATURES SW SWITCH
010666  010465  000000  10$:    MOV     R4,TSDB(R5)             ;SEND OUT COMMAND
010672  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR SSR
010676  103401                  BCS     20$                     ;BR, IF SSR IS SET AND OK
010700  000435                  BR      60$                     ;BR IF TROUBLE CARRY = CLEAR
010702  016501  000002  20$:    MOV     TSSR(R5),R1             ;READ TSSR
010706  012702  000200          MOV     #SSR,R2                 ;SET UP EXPECTED
010712  032701  000100          BIT     #OFL,R1                 ;WAS OFF LINE SET IN TSSR
010716  001402                  BEQ     25$                     ;BR, IF NO OFL SET
010720  052702  000100          BIS     #OFL,R2                 ;MAKE THEM LOOK ALIKE
010724  020201          25$:    CMP     R2,R1                   ;ARE THEY OK
010726  001401                  BEQ     40$                     ;BR, IF EQUAL = OK
010730  000421                  BR      60$                     ;TROUBLE EXIT
010732  062704  000010  40$:    ADD     #8.,R4                  ;POINT TO WRT CHARA DATA PACKET
010736  011403                  MOV     (R4),R3                 ;GET ADDRESS OF MESSAGE BUFFER
010740  032763  000200  000012  BIT     #X2.EXTF,XST2(R3)       ;EXTENDED FEATURES BIT SET?
010746  001402                  BEQ     45$                     ;BR IF NO
010750  005237  002224          INC     EXTFEA                  ;SET EXTENDED FEATURES SW SWITCH
010754                  45$:
010754  032763  000100  000012  BIT     #X2.BUFE,XST2(R3)                       ;BUFFER ENABLE SWITCH SET
010762  001402                  BEQ     50$                     ;BR, IF SWITCH NOT SET
010764  005237  002226          INC     BENBSW                  ;SET SOFTWARE SWITCH FOR ENABLED
```

```
1152 010770                        50$:
1153 010770  000261                    SEC                     ;SET CARRY NO TROUBLE
1154 010772  000401                    BR      70$             ;EXIT
1155 010774  000241                60$: CLC                     ;CARRY CLEAR = ERROR
1156 010776  016500  000002       70$: MOV     TSSR(R5),R0     ;RETURN TSSR CONTENTS
1157 011002  000207                    RTS     PC              ;RETURN
1158
1159
```

```
                                        .SBTTL  REWIND  - POSITION TAPE (REWIND) COMMAND

                                ;+
                                ;
                                ;THIS ROUTINE WILL REWIND THE SELECTED TAPE.
                                ;
                                ;       CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
                                ;                TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
                                ;                SSR TO SET IN THE TSSR
                                ;
                                ;
                                ;CALLING SEQUENCE:
                                ;
                                ;       DO A SOFT INIT
                                ;       DO A WRITE CHARACTERISTICS
                                ;       JSR     PC,REWIND
                                ;
                                ;INPUT:
                                ;
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;
                                ;
                                ;OUTPUT
                                ;
                                ;       R0      THE CONTENTS OF R4 IS PASSED TO R0
                                ;
                                ;
                                ;-
                                REWIND::
                                        SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011004                                  MOV     #RWPACK,R4              ;GET PACKET ADDRESS
011004  012704  011100                  MOV     R4,TSDB(R5)             ;SEND PACKET ADDRESS TO EXECUTE
011010  010465  000000                  MOV     #360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
011014  012703  000550          10$:    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
011020  004737  016240                  BCS     20$                     ;LEAVE WHEN SSR IS SET
011024  103417                          DELAY   250.                    ;WAIT FOR .25 SECONDS
011030                                  MOV     #250.,(PC)+
011032  012727  000372                  .WORD   0
011032  000000                          MOV     L$DLY,(PC)+
011036  013727  002116                  .WORD   0
011040  000000                          DEC     -6(PC)
011044  005367  177772                  BNE     .-4
011046  001375                          DEC     -22(PC)
011052  005367  177756                  BNE     .-20
011054  001367                          DEC     R3                      ;BUMP COUNTER DOWN
011060  005303                          BNE     10$                     ;KEEP GOING
011062  001357                          CLC                             ;CLEAR CARRY TO SET ERROR
011064  000241          20$:    MOV     R4,R0                           ;PASS THE PACKET ADDRESS
011066  010400                  RTS     PC                              ;RETURN
011070  000207
011072
                                .=<.+10>&177770
011100          011100          RWPACK:
011100                                  .WORD   102010                  ;POSTION COMMAND (REWIND)
011100  102010                          .WORD   0                       ;NOT USED
011102  000000
```

```
1212
1213
1214
```

```
                        .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
                ;
                ;-

011104                  CKRAM::
011104                          SAVREG                  ;SAVE THE GENERAL REGISTERS
011110  012701  002240          MOV     #RAMDATA,R1     ;ADDRESS TO SAVE THE RAM DATA
011114  012702  000201          MOV     #RMPKTBEG,R2    ;BYTE ADDRESS OF FIRST RAM DATA
011120  005003                  CLR     R3              ;CLEAR THE ERROR FLAG
011122  004737  016326          JSR     PC,CHKTSSR      ;WAIT FOR SSR
011126  112765  000000  000000  MOVB    #0,TSDB(R5)     ;SET MAINTENANCE MODE
011134  004737  016326  10$:    JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
011140  010265  000000          MOV     R2,TSDB(R5)     ;SELECT NEXT RAM ADDRESS
011144  004737  016326          JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
011150  116511  000000          MOVB    TSBA(R5),(R1)   ;READ THE RAM DATA
011154  122124                  CMPB    (R1)+,(R4)+     ;COMPARE TO EXPECTED
011156  001401                  BEQ     20$             ;BRANCH IF OK
011160  005203                  INC     R3              ;SET ERROR FLAG
011162  005202          20$:    INC     R2              ;ADDRESS OF NEXT RAM LOCATION
011164  020227  000210          CMP     R2,#RMPKTEND    ;REACHED END YET ?
011170  003761                  BLE     10$             ;BRANCH TILL ALL READ
011172  005703                  TST     R3              ;WAS AN ERROR FOUND ?
011174  001402                  BEQ     30$             ;BRANCH IF NOT
011176  000241                  CLC                     ;CLEAR CARRY TO SHOW ERROR
011200  000401                  BR      50$             ;AND EXIT
011202  000261          30$:    SEC                     ;SHOW GOOD COMPARE
011204  012737  000010  002300  50$:    MOV     #8.,RAMSIZ      ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
011212  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
                        ;
                        ;-

011214                  CKRAM2::
011214                          SAVREG                          ;SAVE THE GENERAL REGISTERS
011220  012701  002240          MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
011224  012702  000167          MOV     #RMCHBEG,R2             ;BYTE ADDRESS OF FIRST RAM DATA
011230  005003                  CLR     R3                      ;CLEAR THE ERROR FLAG
011232  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR SSR
011236  112765  000000  000000  MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
011244  004737  016326  10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011250  010265  000000          MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
011254  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011260  116511  000000          MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
011264  122124                  CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
011266  001401                  BEQ     20$                     ;BRANCH IF OK
011270  005203                  INC     R3                      ;SET ERROR FLAG
011272  005202          20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
011274  012737  000010  002300  MOV     #8.,RAMSIZ              ;ASSUME EXTFEA NOT SET
011302  005737  002224          TST     EXTFEA                  ;IS THE SOFTWARE EXTENDED FEATURES SET
011306  001407                  BEQ     25$                     ;BR, IF NOT SET
011310  012737  000012  002300  MOV     #10.,RAMSIZ             ;SET RAMSIZ FOR EXTEND FEATURES
011316  020227  000200          CMP     R2,#RMCHEND             ;AT END OF EXTENDED BUFFER
011322  003750                  BLE     10$                     ;BR, IF NOT AT END YET
011324  000403                  BR      27$                     ;AT END BRANCH
011326  020227  000176  25$:    CMP     R2,#RMCHEND-2           ;REACHED END YET ?
011332  003744                  BLE     10$                     ;BRANCH TILL ALL READ
011334  005703          27$:    TST     R3                      ;WAS AN ERROR FOUND ?
011336  001402                  BEQ     30$                     ;BRANCH IF NOT
011340  000241                  CLC                             ;CLEAR CARRY TO SHOW ERROR
011342  000401                  BR      50$                     ;AND EXIT
011344  000261          30$:    SEC                             ;SHOW GOOD COMPARE
```

```
 1328 011346  000207                50$:    RTS     PC                      ;RETURN
 1329
```

```
                                .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
                        ;
                        ;-
011350                  CKMSG::
011350                          SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011354  010037  002302          MOV     R0,RCVHIADD             ;SAVE RECV HIGH ADDRESS
011360  010137  002304          MOV     R1,RCVLOAD              ;SAVE RECV LOW ADDRESS
011364  005737  003130          TST     KTENABLE                ;TESTING ABOVE 28K?
011370  001403                  BEQ     10$                     ;BR IF NO
011372  004737  017306          JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
011376  010001                  MOV     R0,R1                   ;GET RETURNED ADDRESS BIASED TO PAR6
011400  005004          10$:    CLR     R4                      ;WORD IN BUFFER
011402  005003                  CLR     R3                      ;CLEAR ERROR SEEN FLAG
011404  010205                  MOV     R2,R5                   ;GET EXPD BUFFER ADDRESS
011406  011264  002320  15$:    MOV     (R2),EXPMSG(R4)         ;SAVE EXPD FOR ERROR REPORT
011412  011164  002464          MOV     (R1),RECMSG(R4)         ;SAVE RECV FOR ERROR REPORT
011416  022221                  CMP     (R2)+,(R1)+             ;EXPD EQUAL RECV?
011420  001401                  BEQ     25$                     ;BR IF YES
011422  005203                  INC     R3                      ;SET ERROR SEEN FLAG
011424  062704  000002  25$:    ADD     #2,R4                   ;POINT TO NEXT WORD ADDRESS
011430  020427  000014          CMP     R4,#14                  ;DONE FIRST 7 WORDS?
011434  003764                  BLE     15$                     ;BR IF NO
011436  032765  000200  000012  BIT     #X2.EXTF,XST2(R5);IS EXTENDED FEATURES SET IN EXPD?
011444  001403                  BEQ     50$                     ;BR IF NO
011446  020427  000016          CMP     R4,#16                  ;DONE EXTENDED FEATURES WORD?
011452  003755                  BLE     15$                     ;BR IF NO
011454  005703          50$:    TST     R3                      ;ANY ERRORS SEEN?
011456  001402                  BEQ     55$                     ;BR IF NO
011460  000241                  CLC                             ;SET FAILURE
011462  000401                  BR      60$                     ;
011464  000261          55$:    SEC                             ;SET SUCCESS
011466  000207          60$:    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
                        ;
                        ;-
                        CKMSG2::
                                SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
                                CMP     R3,#RECMSG-EXPMSG;@@D IS COUNT ABOVE MAX ALLOWED?
                                BLE     5$                      ;@@D BR IF NO
                                MOV     #RECMSG-EXPMSG,R3;@@D
                                PRINTF  #DEBUGMSG               ;@@D
                                MOV     #DEBUGMSG,-(SP)
                                MOV     #1,-(SP)
                                MOV     SP,R0
                                TRAP    C$PNTF
                                ADD     #4,SP
                        5$:     MOV     R0,RCVHIADD             ;SAVE RECV HIGH ADDRESS
                                MOV     R1,RCVLOAD              ;SAVE RECV LOW ADDRESS
                                TST     KTENABLE                ;TESTING ABOVE 28K?
                                BEQ     10$                     ;BR IF NO
                                JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
                                MOV     R0,R1                   ;GET RETURNED ADDRESS BIASED TO PAR6
                        10$:    CLR     R4                      ;WORD IN BUFFER
                                CLR     R5                      ;CLEAR ERROR SEEN FLAG
                        15$:    MOVB    (R2),EXPMSG(R4)         ;SAVE EXPD FOR ERROR REPORT
                                MOVB    (R1),RECMSG(R4)         ;SAVE RECV FOR ERROR REPORT
                                CMPB    (R2)+,(R1)+             ;EXPD EQUAL RECV?
                                BEQ     25$                     ;BR IF YES
                                INC     R5                      ;SET ERROR SEEN FLAG
                        25$:    ADD     #1,R4                   ;POINT TO NEXT BYTE
                                CMP     R4,R3                   ;DONE ALL BYTES?
                                BGE     50$                     ;BR IF YES
                                BR      15$                     ;DO NEXT BYTE
                        50$:    TST     R5                      ;ANY ERRORS SEEN?
                                BEQ     55$                     ;BR IF NO
```

```
1439 011612  000241                        CLC                     ;SET FAILURE
1440 011614  000401                        BR      60$             ;
1441 011616  000261                55$:     SEC                     ;SET SUCCESS
1442 011620  000207                60$:     RTS     PC              ;RETURN
1443
1444 011622      120      122      117  DEBUGMSG:       .ASCIZ  'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-';@@D
1445 011712      045      116      045  FERCM:  .ASCII  /%N%A  ***/
1446 011723      040      040      124  ERCM:   .ASCIZ  /  TSSR ERROR CODE REC'D = /
1447 011756      056      056      056  SIMSG:  .ASCIZ  /.... AFTER DOING SOFT INIT/
1448 012011      124      105      123  TINERR: .ASCIZ  /TEST: .../
1449                                     .EVEN
```

```
                                ;+
                                ;
                                ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
                                ;
                                ;INPUT:
                                ;
                                ;       R1      CONTENTS OF TSSR AT ERROR
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;       EXECUTES DROP UNIT TO CEASE TESTING
                                ;
                                ;-

012024                                  BGNMSG  SFIMSG
012024                          SFIMSG::
012024  004737  006020                  JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
012030  004737  017172                  JSR     PC,CKDROP       ;DROP UNIT, IF ALLOWED
012034                                  ENDMSG
012034                          L10003:
012034  104423                          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
                                ;
                                ;-

012036                                  BGNMSG  PKTSSR
012036                          PKTSSR::
012036  004737  006020                  JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
012042  012700  000004                  MOV     #4,R0           ;NO. OF WORDS IN PACKET
012046  004737  007364                  JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
012052                                  ENDMSG
012052                          L10004:
012052  104423                          TRAP    C$MSG

                                ;+
                                ;PRINT ROUTINE TO PRINT THE CONTENTS OF
                                ;TSSR AND A GET STATUS COMMAND PACKET.
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      TSSR CONTENTS
                                ;       R4      ADDRESS OF COMMAND PACKET
                                ;
                                ;-

012054                                  BGNMSG  PKTGETS
012054                          PKTGETS::
```

```
1501 012054  004737  006020                    JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1502 012060  012700  000002                    MOV     #2,R0           ;NO. OF WORDS IN GET STATUS PACKET
1503 012064  004737  007364                    JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
1504 012070                                    ENDMSG
     012070                              L10005:
     012070  104423                            TRAP    C$MSG
1505
1506
1507                                        ;+
1508                                        ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
1509                                        ;
1510                                        ;INPUTS:
1511                                        ;
1512                                        ;       R1      TSSR CONTENTS
1513                                        ;       R4      ADDRESS OF COMMAND PACKET
1514                                        ;-
1515
1516 012072                                    BGNMSG  SFFMSG
     012072                              SFFMSG::
1517 012072  004737  006020                    JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1518 012076                                    ENDMSG
     012076                              L10006:
     012076  104423                            TRAP    C$MSG
1519
1520
1521                                            .SBTTL  PKTMES  - PRINT TSSR AND MESSAGE BUFFER
1522                                        ;+
1523                                        ;
1524                                        ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
1525                                        ;BUFFER FOR ERROR REPORTS
1526                                        ;
1527                                        ;INPUTS:
1528                                        ;
1529                                        ;       R1      CONTENTS OF TSSR
1530                                        ;       R2      LOW ORDER MESSAGE BUFFER
1531                                        ;       R3      HIGH ORDER MESSAGE BUFFER ADDRESS
1532                                        ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
1533                                        ;-
1534 012100                                    BGNMSG  PKTMES
     012100                              PKTMES::
1535 012100  004737  006020                    JSR     PC,PRITSSR              ;PRINT CONTENTS OF TSSR
1536 012104  010200                            MOV     R2,R0                   ;LOW ORDER ADDRESS
1537 012106  010301                            MOV     R3,R1                   ;HIGH ORDER ADDRESS
1538 012110  004737  014232                    JSR     PC,PRMESS               ;PRINT THE MESSAGE BUFFER
1539 012114                                    ENDMSG
     012114                              L10007:
     012114  104423                            TRAP    C$MSG
1540
```

```
                                .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
                        ;-
                        
012116                          BGNMSG  ADDSSR
012116                  ADDSSR::
012116  004737  010270          JSR     PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
012122  016501  000002          MOV     TSSR(R5),R1     ;GET CURRENT TSSR
012126  004737  006C20          JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
012132                          ENDMSG
012132                  L10010:
012132  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
                        ;-
012134                          BGNMSG  MSGEXP
012134                  MSGEXP::
012134  012700  000007          MOV     #7,R0           ;ASSUME NO EXT FEATURES
012140  005737  002224          TST     EXTFEA          ;EXT FEATURES SET?
012144  001402                  BEQ     5$              ;BR IF NO
012146  012700  000010          MOV     #8.,R0          ;EXT FEATURE BUFFER IS 8 WORDS
012152  004737  014542  5$:     JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012156                          ENDMSG
012156                  L10011:
012156  104423                  TRAP    C$MSG
```

```
                                        .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
                                FIFEXP::
                                        PRINTX  #FIF1MSG,R1     ;PRINT BYTES TRANSFERRED
012160  010146                          MOV     R1,-(SP)
012162  012746  012232                  MOV     #FIF1MSG,-(SP)
012166  012746  000002                  MOV     #2.-(SP)
012172  010600                          MOV     SP,R0
012174  104415                          TRAP    C$PNTX
012176  062706  000006                  ADD     #6,SP
                                        PRINTX  #FIF2MSG        ;PRINT HEADER MSG
012202  012746  012301                  MOV     #FIF2MSG,-(SP)
012206  012746  000001                  MOV     #1,-(SP)
012212  010600                          MOV     SP,R0
012214  104415                          TRAP    C$PNTX
012216  062706  000004                  ADD     #4,SP
012222  010100                          MOV     R1,R0           ;GET BYTE COUNT
012224  004737  015112                  JSR     PC,PRBYTEXP     ;PRINT FIFO BYTES IN ERROR
                                        ENDMSG
                                L10012:
012230  104423                          TRAP    C$MSG
012232  045     116     045     FIF1MSG:        .ASCIZ  '%N%A NUMBER OF BYTES TRANSFERRED = %D2'
012301  045     116     045     FIF2MSG:        .ASCIZ  '%N%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
                ;-
012340                  BGNMSG  MSGSTAT
012340          MSGSTAT::
012340 012701 012402    MOV     #STATCOD,R1     ;ASCII ADDRESS TABLE
012344 012100   10$:    MOV     (R1)+,R0        ;DONE ALL MSG LINES?
012346 001410           BEQ     20$             ;BR IF YES
012350                  PRINTX  R0              ;PRINT STATUS BIT NAMES
012350 010046           MOV     R0,-(SP)
012352 012746 000001    MOV     #1,-(SP)
012356 010600           MOV     SP,R0
012360 104415           TRAP    C$PNTX
012362 062706 000004    ADD     #4,SP
012366 000766           BR      10$             ;DO ANOTHER MSG LINE
012370 012700 000012    20$:    MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
012374 004737 014542    JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012400                  ENDMSG
012400          L10013:
012400 104423           TRAP    C$MSG

012402 012420 012462 012553 STATCOD:    .WORD   1$,2$,3$,4$,5$,6$,0
012420   045  116   045   1$:.ASCIZ  '%N%A Tape Bus Signals in Word #8:'
012462   045  116   045   2$:.ASCIZ  '%N%A         PARERR<15>  IEOT  <12>  IFMK  <9>  IRDY<6>  IRWD<2>'
012553   045  116   045   3$:.ASCIZ  '%N%A         IRESV2<14>  IIDENT<11>  IHER  <8>  IONL<5>  IFBY<1>'
012644   045  116   045   4$:.ASCIZ  '%N%A         IRESV1<13>  ICER  <10>  ISPEED<7>  ILDP<4>  IFPT<0>'
012735   045  116   045   5$:.ASCIZ  '%N%A Tape Bus Signals in Word #9:'
012777   045  116   045   6$:.ASCIZ  '%N%A         DATMIS<7>  ILW<6>  OUTRDY<5>  INRDY<4>'
                        .EVEN


                        .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
                ;-
013054                  BGNMSG  MSGLOOP
013054          MSGLOOP::
013054 012701 013116    MOV     #LOOPCOD,R1     ;ASCII ADDRESS TABLE
```

```
 1654 013060  012100                  10$:   MOV     (R1)+,R0        ;DONE ALL MSG LINES?
 1655 013062  001410                         BEQ     20$             ;BR IF YES
 1656 013064                                 PRINTX  R0              ;PRINT STATUS BIT NAMES
      013064  010046                         MOV     R0,-(SP)
      013066  012746  000001                 MOV     #1,-(SP)
      013072  010600                         MOV     SP,R0
      013074  104415                         TRAP    C$PNTX
      013076  062706  000004                 ADD     #4,SP
 1657 013102  000766                         BR      10$             ;DO ANOTHER MSG LINE
 1658 013104  012700  000012          20$:   MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
 1659 013110  004737  014542                 JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
 1660 013114                                 ENDMSG
      013114                          L10014:
      013114  104423                         TRAP    C$MSG
 1661
 1662 013116  013136  013211  013310  LOOPCOD:        .WORD   1$,2$,3$,4$,5$,6$,7$,0
 1663 013136     045     116     045  1$:.ASCIZ  '%N%A Tape Bus Loopback Signals in Word #8:'
 1664 013211     045     116     045  2$:.ASCIZ  '%N%A      PARERR<15>          IRESV2<14>          IRESV1<13>'
 1665 013310     045     116     045  3$:.ASCIZ  '%N%A IHISP=>IEOT<12>    IWRT=>IIDENT<11>    IREV  =>ICER  <10>'
 1666 013407     045     116     045  4$:.ASCIZ  '%N%A IWFM =>IFMK<09>    IEDIT=>IHER <08>    IFAD  =>ISPEED<07>'
 1667 013506     045     116     045  5$:.ASCIZ  '%N%A ITAD0=>IRDY<06>    ITAD1=>IONL <05>    IERASE=>ILDP  <04>'
 1668 013605     045     116     045  6$:.ASCIZ  '%N%A IREW =>IDBY<03>    IRWU =>IRWD <02>    IFEN  =>IFBY  <01>'
 1669 013704     045     116     045  7$:.ASCIZ  '%N%A IGO  =>IFPT<00>'
 1670                                        .EVEN
 1671
```

```
                                        .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
013732                          MSGSUB::
013732
013732  012700  000012                  MOV     #10.,R0         ;SIZE OF WRITE SUBSYSTEM BUFFER
013736  004737  014542                  JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
013742                                  ENDMSG
013742                          L10015:
013742  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
013744                          MEMADD::
013744
013744  004737  010154                  JSR     PC,PRIADD       ;PRINT MEMORY ADDRESS IN ERROR
013750  013701  002230                  MOV     EXPD,R1         ;GET EXPD DATA
013754  013702  002232                  MOV     RECV,R2         ;GET RECEIVED DATA
013760  004737  007736                  JSR     PC,PRIXOR       ;PRINT EXPD/RECV
013764                                  ENDMSG
013764                          L10016:
013764  104423                          TRAP    C$MSG
```

```
                        .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:
013766                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
013766 012701 002240    MOV     #RAMDATA,R1             ;DATA FROM THE RAM
013772 005002           CLR     R2                      ;INIT BYTE NUMBER
013776 122124   5$:     CMPB    (R1)+,(R4)+             ;COMPARE EXPECTED, RECEIVED
014000 001005           BNE     7$                      ;BR IF NO MATCH
014002                  FORCERROR       7$,NOTSSR
014004 000436           BR      10$                     ;@@D
014014 116105 177777 7$:MOVB    -1(R1),R5               ;GET RECV RAM DATA
014016 116403 177777    MOVB    -1(R4),R3               ;GET EXPD PACKET DATA
014022                  XOR     R5,R3                   ;XOR EXPD/RECV
014026 042703 177400    BIC     #177400,R3              ;LOW BYTE ONLY
014036 116137 177777 002232     MOVB    -1(R1),RECV     ;GET RECEIVED RAM DATA
014042 116437 177777 002230     MOVB    -1(R4),EXPD     ;GET EXPECTED RAM DATA
014050                  PRINTB  #RAMASC,R2,RECV,EXPD,R3
014056 010346           MOV     R3,-(SP)
014060 013746 002230    MOV     EXPD,-(SP)
014064 013746 002232    MOV     RECV,-(SP)
014070 010246           MOV     R2,-(SP)
014072 012746 014146    MOV     #RAMASC,-(SP)
014076 012746 000005    MOV     #5,-(SP)
014102 010600           MOV     SP,R0
014104 104414           TRAP    C$PNTB
014106 062706 000014    ADD     #14,SP
014112 005202   10$:    INC     R2                      ;UPDATE BYTE COUNT
014114 005737 002300    TST     RAMSIZ                  ;DEFAULT TO 8.?
014120 001404           BEQ     15$                     ;BR IF YES
014122 020237 002300    CMP     R2,RAMSIZ               ;DONE ALL BYTES?
014126 003724           BLE     5$                      ;BR IF NO
014130 000403           BR      25$                     ;
014132 020227 000010 15$:CMP    R2,#8.                  ;DONE DEFAULT NUMBER OF BYTES?
014136 002720   20$:    BLT     5$                      ;BR IF NO
014140 005037 002300 25$:CLR    RAMSIZ                  ;SET DEFAULT RAMSIZ
014144 000207           RTS     PC                      ;RETURN

014146  045  116  045  RAMASC: .ASCIZ  '%N%A  BYTE: %D2%A RAM  %03%A Packet: %03%A XOR:%03'
```

```
1763                                    .EVEN
1764
1765
```

```
                        .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
                ;
                ;-

014232          PRMESS:
014232                  SAVREG                          ;SAVE THE REGISTERS
014236  010005          MOV     R0,R5                   ;SAVE LOW ORDER ADDRESS
014240  005737  003130  TST     KTENABLE                ;ADDRESS ABOVE 28K?
014244  001001          BNE     10$                     ;BR IF YES
014246  005001          CLR     R1                      ;SET HIGH ORDER ADDRESS TO 0
014250  010103  10$:     MOV     R1,R3                   ;SAVE HIGH ORDER ADDRESS
014252  006100          ROL     R0                      ;SHIFT BIT15 TO C BIT
014254  006101          ROL     R1                      ;SHIFT TO HIGH ORDER FOR PRINTOUT
014256                  PRINTX  #PROASC,R1,R5           ;PRINT MESSAGE BUFFER ADDRESS
014256  010546          MOV     R5,-(SP)
014260  010146          MOV     R1,-(SP)
014262  012746  014410  MOV     #PROASC,-(SP)
014266  012746  000003  MOV     #3,-(SP)
014272  010600          MOV     SP,R0
014274  104415          TRAP    C$PNTX
014276  062706  000010  ADD     #10,SP
014302                  PRINTX  #PR1ASC                 ;PRINT HEADER FOR CONTENTS
014302  012746  014455  MOV     #PR1ASC,-(SP)
014306  012746  000001  MOV     #1,-(SP)
014312  010600          MOV     SP,R0
014314  104415          TRAP    C$PNTX
014316  062706  000004  ADD     #4,SP
014322  005004          CLR     R4                      ;NUMBER OF THE NEXT WORD
014324  010501          MOV     R5,R1                   ;COPY LOW ORDER ADDRESS
014326  010300          MOV     R3,R0                   ;COPY HIGH ORDER ADDRESS
014330  001403          BEQ     20$                     ;BR IF NOT ABOVE 28K
014332  004737  017306  JSR     PC,SETMAP               ;SETUP PAR ADDRESS IN R0
014336  010005          MOV     R0,R5                   ;GET PAR FORMAT ADDRESS ABOVE 28K
014340          20$:    PRINTX  #PRASC,R4,(R5)+         ;PRINT THE CONTENTS OF MEMORY BUFFER
014340  012546          MOV     (R5)+,-(SP)
014342  010446          MOV     R4,-(SP)
014344  012746  014513  MOV     #PRASC,-(SP)
014350  012746  000003  MOV     #3,-(SP)
014354  010600          MOV     SP,R0
014356  104415          TRAP    C$PNTX
014360  062706  000010  ADD     #10,SP
014364  005204          INC     R4                      ;NUMBER OF THE NEXT
014366  020427  000007  CMP     R4,#7                   ;DONE ALL YET ?
014372  003005          BGT     50$                     ;BRANCH IF ALL DONE
```

```
1805 014374  002761                        BLT     20$                ;PRINT FIRST 7 WORDS
1806 014376  032763  000200  000012        BIT     #X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
1807 014404  001355                        BNE     20$                ;PRINT EXTENDED STATUS WORD
1808 014406  000207                 50$:   RTS     PC                 ;RETURN
1809
1810 014410     045     116     045  PROASC: .ASCIZ  '%N%A Message Buffer Address = %O1%O5'
1811 014455     045     116     045  PR1ASC: .ASCIZ  '%N%A Message Buffer Contents:'
1812 014513     045     116     045  PRASC:  .ASCIZ  '%N%A    Word%D1%A: %O'
1813                                        .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
                ;-
014542          PRMSGEXP::
                        SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
014546  010005          MOV     R0,R5                   ;SAVE NUMBER OF WORDS
014550  013700  002304  MOV     RCVLOADD,R0             ;GET RECV LOW ADDRESS
014554  010004          MOV     R0,R4                   ;COPY LOW ADDRESS
014556  013701  002302  MOV     RCVHIADD,R1             ;GET RECV HIGH ADDRESS
014562  006100          ROL     R0                      ;SHIFT BIT15 TO C BIT
014564  006101          ROL     R1                      ;SHIFT TO HIGH ORDER FOR PRINTOUT
014566                  PRINTX  #PRMSG0,R1,R4           ;PRINT MESSAGE BUFFER ADDRESS
014566  010446          MOV     R4,-(SP)
014570  010146          MOV     R1,-(SP)
014572  012746  014722  MOV     #PRMSG0,-(SP)
014576  012746  000003  MOV     #3,-(SP)
014602  010600          MOV     SP,R0
014604  104415          TRAP    C$PNTX
014606  062706  000010  ADD     #10,SP
014612                  PRINTX  #PRMSG1                 ;PRINT HEADER FOR CONTENTS
014612  012746  014767  MOV     #PRMSG1,-(SP)
014616  012746  000001  MOV     #1,-(SP)
014622  010600          MOV     SP,R0
014624  104415          TRAP    C$PNTX
014626  062706  000004  ADD     #4,SP
014632  005004          CLR     R4                      ;NUMBER OF THE CURRENT WORD
014634  012701  002320  MOV     #EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
014640  012702  002464  MOV     #RECMSG,R2              ;GET RECV BUFFER ADDRESS
014644  011100  20$:    MOV     (R1),R0                 ;GET EXPD
014646  011203          MOV     (R2),R3                 ;GET RECV
014650                  XOR     R0,R3                   ;XOR EXPD/RECV
014660                  PRINTX  #PRMSG2,R4,(R1)+,(R2)+,R3
014660  010346          MOV     R3,-(SP)
014662  012246          MOV     (R2)+,-(SP)
014664  012146          MOV     (R1)+,-(SP)
014666  010446          MOV     R4,-(SP)
014670  012746  015025  MOV     #PRMSG2,-(SP)
014674  012746  000005  MOV     #5,-(SP)
014700  010600          MOV     SP,R0
014702  104415          TRAP    C$PNTX
014704  062706  000014  ADD     #14,SP
014710  005204          INC     R4                      ;NUMBER OF THE NEXT
014712  020405          CMP     R4,R5                   ;DONE ALL YET?
014714  002001          BGE     50$                     ;BR IF YES
014716  000752          BR      20$                     ;DO ANOTHER
014720  000207  50$:    RTS     PC                      ;RETURN
```

```
 1851
 1852 014722     045     116     045  PRMSG0: .ASCIZ  '%N%A Message Buffer Address = %01%05'
 1853 014767     045     116     045  PRMSG1: .ASCIZ  '%N%A Message Buffer Contents:'
 1854 015025     045     116     045  PRMSG2: .ASCIZ  '%N%A   WORD #%D2%A  EXPD: %06%A  RECV: %06%A  XOR: %06'
 1855                                          .EVEN
 1856
```

```
                        .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
                ;-
015112          PRBYTEXP::
                        SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
015116  010005          MOV     R0,R5                   ;SAVE NUMBER OF BYTES
015120  005037  002316  CLR     PRMNO                   ;INIT ERROR COUNT
015124  005004          CLR     R4                      ;NUMBER OF THE CURRENT BYTE
015126  012701  002320  MOV     #EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
015132  012702  002464  MOV     #RECMSG,R2              ;GET RECV BUFFER ADDRESS
015136  111100          20$:    MOVB    (R1),R0         ;GET EXPD BYTE
015140  042700  177400  BIC     #^C<377>,R0             ;CLEAR UPPER BYTE
015144  110037  015460  MOVB    R0,PRBEXP               ;SAVE FOR ERROR REPORT
015150  111203          MOVB    (R2),R3                 ;GET RECV BYTE
015152  042703  177400  BIC     #^C<377>,R3             ;CLEAR UPPER BYTE
015156  110337  015462  MOVB    R3,PRBREC               ;FOR ERROR REPORT
015162                  XOR     R0,R3                   ;XOR EXPD/RECV
015172  122122          CMPB    (R1)+,(R2)+             ;EXPD = RECV?
015174  001431          BEQ     30$                     ;BR IF YES
015176  005237  002316  INC     PRMNO                   ;UPDATE ERROR COUNT
015202  023727  002316  000010  CMP     PRMNO,#8.       ;PRINTED 8?
015210  101023          BHI     30$                     ;BR IF YES
015212          27$:    PRINTX  #PRBMSG,R4,PRBEXP,PRBREC,R3
015212  010346          MOV     R3,-(SP)
015214  013746  015462  MOV     PRBREC,-(SP)
015220  013746  015460  MOV     PRBEXP,-(SP)
015224  010446          MOV     R4,-(SP)
015226  012746  015326  MOV     #PRBMSG,-(SP)
015232  012746  000005  MOV     #5,-(SP)
015236  010600          MOV     SP,R0
015240  104415          TRAP    C$PNTX
015242  062706  000014  ADD     #14,SP
015246                  FORCEXIT        50$             ;aad
015256  000404          BR      35$                     ;aad
015260          30$:
015260                  FORCERROR  27$,NOTSSR           ;aad
015270          35$:                                    ;aad
015270  005204          INC     R4                      ;NUMBER OF THE NEXT
015272  020405          CMP     R4,R5                   ;DONE ALL YET?
015274  002001          BGE     50$                     ;BR IF YES
015276  000717          BR      20$                     ;DO ANOTHER
015300          50$:    PRINTX  #PRBTOT,PRMNO           ;PRINT TOTAL ERROR COUNT
015300  013746  002316  MOV     PRMNO,-(SP)
015304  012746  015413  MOV     #PRBTOT,-(SP)
015310  012746  000002  MOV     #2,-(SP)
015314  010600          MOV     SP,R0
015316  104415          TRAP    C$PNTX
```

```
        015320  062706  000006                  ADD     #6,SP
1901    015324  000207                          RTS     PC              ;RETURN
1902
1903    015326      045     116     045 PRBMSG: .ASCIZ  '%N%A   BYTE #%D2%A  EXPD: %03%A  RECV: %03%A  XOR: %03'
1904    015413      045     116     045 PRBTOT: .ASCIZ  '%N%A NUMBER OF BYTES IN ERROR = %D2'
1905                                            .EVEN
1906    015460  000000                  PRBEXP: .WORD   0               ;EXPD
1907    015462  000000                  PRBREC: .WORD   0               ;RECV
1908
```

```
                                .SBTTL  EXPREC  - PRINT EXPD/RECV WORD DATA
                        ;+
                        ;
                        ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                        ;
                        ;INPUTS:
                        ;
                        ;       R1      RECEIVED DATA
                        ;       R2      EXPECTED DATA
                        ;
                        ;-
                        ;
                                BGNMSG  EXPREC
015464                  EXPREC::
015464  004737  007736          JSR     PC,PRIXOR               ;PRINT THE DATA
015470                          ENDMSG
015470                  L10017:
015470  104423                  TRAP    C$MSG
```

```
                                        .SBTTL  EXPBREC - PRINT EXPD/RECV BYTE DATA
                                ;+
                                ;
                                ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA BYTE
                                ;       R2      EXPECTED DATA BYTE
                                ;
                                ;-
                                        BGNMSG  EXPBREC
                                EXPBREC::
015472  004737  007606                  JSR     PC,PRIBXOR              ;PRINT THE DATA
                                        ENDMSG
                                L10020:
015476  104423                          TRAP    C$MSG


                                        .SBTTL  RAMERR  - PRINT RAM AND PACKET DATA
                                ;+
                                ;
                                ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
                                ;
                                ;INPUTS:
                                ;
                                ;       R4      POINTER TO COMMAND PACKET
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       RAMDATA         DATA AS READ FROM THE RAM
                                ;       RAMSIZ          NUMBER OF BYTES IN PACKET
                                ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
                                ;
                                ;IMPLICIT OUTPUTS:
                                ;
                                ;       RAMSIZ  SET TO 0
                                ;-
                                        BGNMSG  RAMERR
                                RAMERR::
015500  004737  013766                  JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
                                        ENDMSG
                                L10021:
015504  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::
015506 004737 010270                    JSR     PC,PRITADD      ;PRINT TEST ADDRESS
015512 004737 013766                    JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
                                        ENDMSG
                                L10022:
015516 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
                                ;       R4      CONTROLLER RAM ADDRESS
                                ;-
                                
                                        BGNMSG  RAMEXP
                                RAMEXP::
015520 042701 177400                    BIC     #^C<377>,R1             ;SAVE EXPD RAM DATA BYTE
015524 042702 177400                    BIC     #^C<377>,R2             ;SAVE EXPD RAM DATA BYTE
015530 004737 010062                    JSR     PC,PRIRAM               ;PRINT THE RAM ADDRESS
015534 004737 007736                    JSR     PC,PRIXOR               ;PRINT THE DATA
                                        ENDMSG
                                L10023:
015540 104423                           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
```

```
2030                                    ;-
2031
2032 015542                                     BGNMSG  TIMEXP
     015542                             TIMEXP::
2033 015542                                     PRINTX  #TIMSG0                 ;PRINT HEADER
     015542  012746  015570                     MOV     #TIMSG0,-(SP)
     015546  012746  000001                     MOV     #1,-(SP)
     015552  010600                             MOV     SP,R0
     015554  104415                             TRAP    C$PNTX
     015556  062706  000004                     ADD     #4,SP
2034 015562  004737  007736                     JSR     PC,PRIXOR               ;PRINT THE DATA
2035 015566                                     ENDMSG
     015566                             L10024:
     015566  104423                             TRAP    C$MSG
2036
2037
2038 015570    045    116    045  TIMSG0: .ASCIZ  '%N%A TIMER A STATUS IS IN BIT 3%N%A TIMER B STATUS IS IN BIT 2'
2039                                    .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)
                ;
                ;-

015670                  BGNMSG  BADSSR
015670          BADSSR::
015670  010246          MOV     R2,-(SP)                ;SAVE DATA TRANSFERRED
015672  042702  177400  BIC     #177400,R2              ;GET JUST ONE BYTE
015676          PRINTB  #XFERASC,R2
015676  010246          MOV     R2,-(SP)
015700  012746  015730  MOV     #XFERASC,-(SP)
015704  012746  000002  MOV     #2,-(SP)
015710  010600          MOV     SP,R0
015712  104414          TRAP    C$PNTB
015714  062706  000006  ADD     #6,SP
015720  012602          MOV     (SP)+,R2                ;RESTORE R2
015722  004737  006020  JSR     PC,PRITSSR              ;DECODE TSSR CONTENTS
015726          ENDMSG
015726          L10025:
015726  104423          TRAP    C$MSG
015730    045   116   045 XFERASC:      .ASCIZ  '%N%A Data Transferred = %03'
```

```
                        .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
                        ;
                        ;-

015764                  SOFINIT::
015764                          SAVREG                  ; SAVE THE REGISTERS
015770  012765  000000  000002          MOV     #0,TSSR(R5)     ; DO THE INIT.
015776  004737  016240                  JSR     PC,WAITF        ; WAIT FOR SSR
016002  016500  000002                  MOV     TSSR(R5),R0     ;GET THE TSSR REGISTER
016006  010004                          MOV     R0,R4           ;TSSR CONTENTS
016010  042704  176277                  BIC     #^C<HIADDR!OFL>,R4
016014  052704  002200                  BIS     #SSR!NBA,R4     ;R4 HAS EXPECTED CONTENTS
016020  020400                          CMP     R4,R0           ;ONLY EXPECTED BITS SET ?
016022  001402                          BEQ     5$              ;BRANCH IF OKAY
016024  000241                          CLC                     ;CLEAR THE CARRY FOR ERROR
016026  000401                          BR      10$             ;GO TO EXIT
016030  000261                  5$:     SEC                     ;SET THE CARRY BIT
016032  000207                  10$:    RTS     PC              ;RETURN TO CALLER
```

```
                                        .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
                                ;
                                ;-

016034                          CHKAMB:
016034                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
016040  010004                          MOV     R0,R4                   ;CONTENTS OF TSSR
016042  032700  100000                  BIT     #SC,R0                  ;IS BIT 15 SET ?
016046  001004                          BNE     5$                      ;BRANCH IF YES
016050  032700  174077                  BIT     #^C<NBA!OFL!SSR!HIADDR>,R0      ;ANY OTHER BITS SET ?
016054  001023                          BNE     40$                     ;MUST BE AN ERROR
016056  000424                          BR      45$                     ;RETURN WITH SUCCESS
016060  032700  000200          5$:     BIT     #SSR,R0                 ;IS READY BIT SET ?
016064  001011                          BNE     10$                     ;BRANCH IF READY BIT IS SET.
016066  032700  000040                  BIT     #BIT5,R0                ;IS FATAL ERROR BIT SET ?
016072  001414                          BEQ     40$                     ;ERROR IF NOT
016074  042704  177761                  BIC     #^CTERCLS,R4            ;CLEAR ALL BUT TERMINATION CODE
016100  020427  000016                  CMP     R4,#16                  ;ALL THREE BITS MUST BE SET
016104  001007                          BNE     40$                     ;ERROR IF NOT SET
016106  000410                          BR      45$                     ;OK IF ALL ARE SET
016110  032700  000040          10$:    BIT     #BIT5,R0                ;IS FATAL ERROR BIT SET ?
016114  001405                          BEQ     45$                     ;ERROR IF BIT IS SET WITH SSR
016116  032700  000006                  BIT     #BIT2!BIT1,R0           ;IS THIS A FUNCTION REJECT
016122  001002                          BNE     45$                     ;BR, IF TSSR IS OK
016124  000241                  40$:    CLC                             ;AMBIGUOUS CONTENTS
016126  000401                          BR      50$
016130  000261                  45$:    SEC                             ;SHOW SUCCESS - NO AMBIGUITY
016132  000207                  50$:    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
016134      000                 INTMASK:        .BYTE   0
                                ;INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
016135      000                 INTFLAG:        .BYTE   0

                                ;SAVED INTERRUPT VECTOR:
016136  000000                  INTVEC: .WORD   0
                                ;SAVE CPU PC
016140  000000                  INTCPC: .WORD   0

                                ;SUBROUTINE TO ENABLE INTERRUPTS:
016142  010046                  ENAINT: MOV     R0,-(SP)        ;SAVE R0
016144  013700  002206                  MOV     IVEC,R0         ;GET POINTER TO VECTORS
016150  012720  016206                  MOV     #INTR,(R0)+     ;SET UP INTERRUPT VECTOR
016154  012720  000340                  MOV     #PRI07,(R0)+
016160  012600                          MOV     (SP)+,R0        ;RESTORE R0
016162  011646                          MOV     (SP),-(SP)
016164  012766  000000  000002          MOV     #0,2(SP)        ;SET CPU TO LEVEL 0
016172  000002                          RTI

                                ;SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
016174  011646                  DSBINT: MOV     (SP),-(SP)
016176  012766  000340  000002          MOV     #PRI07,2(SP)
016204  000002                          RTI
```

```
2199                                            .SBTTL  INTR    - INTERRUPT HANDLERS
2200
2201 016206                                     BGNSRV  INTR            ;DEFINE INTERRUPT ENTRY
     016206                                 INTR::
2202 016206  012737  000001  002222            MOV     #1,INTRECV      ;SET FLAG TO SHOW INTERRUPT RECEIVED
2203 016214  105037  016135                    CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2204 016220  132737  000001  016134            BITB    #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
2205 016226  001003                            BNE     1$              ;BR IF YES
2206 016230  152737  000001  016135            BISB    #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.
2207
2208                                        ;SAVE REGISTERS, MSG BUFFER, ETC.
2209 016236                                 1$:
2210 016236                                     ENDSRV
     016236                                 L10026:
     016236  000002                            RTI
2211
2212
```

```
                        .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
                ;
016240 000401   WAITF:: BR      1$              ;NOP WHEN SUPER FIXED
016242                  BREAK                   ; DO A SUPVSR BREAK FIRST.
016242 104422           TRAP    C$BRK
016244 012746 003000    1$:     MOV     #3000,-(SP)     ;300 MSEC TIMER
016250 016500 000002    2$:     MOV     TSSR(R5),R0     ;READ THE TSSR REGISTER
016254 105700                   TSTB    R0              ;TEST FOR READY BIT SET
                ;
016256 100420                   BMI     3$              ; EXIT ON STOP FLAG.
016260                          DELAY   1               ; WAIT 100 USEC
016260 012727 000001            MOV     #1,(PC)+
016264 000000                   .WORD   0
016266 013727 002116            MOV     L$DLY,(PC)+
016272 000000                   .WORD   0
016274 005367 177772            DEC     -6(PC)
016300 001375                   BNE     .-4
016302 005367 177756            DEC     -22(PC)
016306 001367                   BNE     .-20
016310 005316                   DEC     (SP)            ;REDUCE DELAY COUNT
016312 001356                   BNE     2$              ;RETRY UNTIL TIMER EXPIRES
016314 000241                   CLC                     ; C = 0, CONTROLLER STILL RUNNING...
016316 000401                   BR      4$              ;...OR HUNG-UP AFTER 300 MSEC.
016320 000261           3$:     SEC                     ; C = 1, CONTROLLER IS STOPPED.
016322 005326           4$:     DEC     (SP)+           ;RESTORE STACK WITHOUT CHANGING CARRY BIT
016324 000207                   RTS     PC
```

```
                        .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
                ;
                ;-

016326          CHKTSSR:
016326  004737  016240          JSR     PC,WAITF                ;WAIT FOR READY
016332  103014                  BCC     20$                     ;BRANCH IF TIME OUT
016334  004737  016034          JSR     PC,CHKAMB               ;TSSR AMBIGOUS?
016340  103006                  BCC     10$                     ;BR IF YES
016342  032700  100000          BIT     #SC,R0                  ;SPECIAL CONDITION SET?
016346  001405                  BEQ     15$                     ;BR IF NO
016350  032700  074000          BIT     #<SCE!BIE!RMR!NXM>,R0   ;ANY ERROR BITS SET?
016354  001402                  BEQ     15$                     ;BR IF NO
016356  000241          10$:    CLC                             ;SET FAILURE
016360  000401                  BR      20$                     ;
016362  000261          15$:    SEC                             ;SET SUCCESS
016364  000207          20$:    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.
                                ;
016366  012737  016420  000004  XNXM:   MOV     #2$,@#4         ; SET BUSERR VECTOR.
016374  012737  000200  000006          MOV     #PRI04,@#6
016402  005003                         CLR     R3              ;FLAG.
016404  005711                  1$:    TST     (R1)            ;TEST THE ADDRESS(ES).
                                                                ;IF ANY TRAP, CONTINUE AT 2$.
016406  020102                         CMP     R1,R2           ;OTHERWISE, CONTINUE HERE.
016410  001407                         BEQ     3$              ;BR IF FINISHED (NO NEXM'S).
016412  062701  000002                 ADD     #2,R1           ;SET NEXT ADDRESS...
016416  000772                         BR      1$              ;...AND CONTINUE.

016420  005103                  2$:    COM     R3              ;GOT ONE, SET FLAG...
016422  012716  016430                 MOV     #3$,(SP)
016426  000002                         RTI                     ;...AND DISMISS INTERRUPT...
016430                          3$:    CLRVEC  #4              ;...AND GIVE BACK THE VECTOR.
016430  012700  000004                 MOV     #4,R0
016434  104436                         TRAP    C$CVEC
016436  005703                         TST     R3              ;DID WE CATCH ONE ??
016440  001401                         BEQ     .+4             ;NO, "C" = 0, SKIP NEXT.
016442  000261                         SEC                     ;YES, "C" = 1, (R1) = NEXM ADDR.
016444  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
                                ;
016446                          TSTLOOP::
016446  005737  002166                 TST     NOITS           ; ITERATIONS INHIBITED?
016452  001006                         BNE     1$              ; YES.
016454  005737  002202                 TST     QVP             ; NO.
016460  100403                         BMI     1$              ;LOOPS DISALLOWED IN QUICK PASS.
016462  005337  002214                 DEC     LOOPCNT         ; BUMP LOOP COUNTER.
016466  001002                         BNE     2$
016470  000241                  1$:    CLC                     ;LOOP DISALLOWED, OR DONE.
016472  000401                         BR      3$
016474  000261                  2$:    SEC                     ;LOOP ENABLED.
016476  000207                  3$:    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 RASIED TO LEVEL OF
                                ;       THE DEVICE UNDER TEST
                                ;
                                ;-
016500                          TSTSETUP::
016500  010046                          MOV     R0,-(SP)        ;SAVE THE TEST ID MESSAGE
016502  005037  003150                  CLR     SIFLAG          ; CLEAR "SOFT INIT" FLAG
016506  005037  016746                  CLR     ERRK            ; CLEAR LOCAL ERROR COUNTER.
016512  005037  005766                  CLR     EXTA            ; CLEAR ERROR EXTENSION FLAG.
016516  105037  016134                  CLRB    INTMASK         ; CLEAR INTERRUPT MASK (CHECK ERROR)
016522  013700  002200                  MOV     UNITN,R0        ; GET THE UNIT NUMBER,
016526  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
016530  005737  003110                  TST     NODEV           ; DID STARTUP FIND THE DEVICE?
016534  001430                          BEQ     4$              ; BR IF YES
016536  100010                          BPL     3$              ; BR IF NOT IDLE
016540  052760  160000  003172          BIS     #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
016546                                  ERRDF   1,NXR,NXRERR    ; NO DEVICE HERE -- PRINT IT
016546  104455                          TRAP    C$ERDF
016550  000001                          .WORD   1
016552  003734                          .WORD   NXR
016554  005732                          .WORD   NXRERR
016556  000407                          BR      2$
016560  052760  160001  003172  3$:     BIS     #160001,ERTABL(R0)      ; FLAG ERROR IN THE ERROR TABLE
016566                                  ERRDF   2,NOINIT        ; DEVICE NOT IDLE
016566  104455                          TRAP    C$ERDF
016570  000002                          .WORD   2
016572  004331                          .WORD   NOINIT
016574  000000                          .WORD   0
016576  012737  177777  003106  2$:     MOV     #-1,DUFLG       ; DROP THE UNIT
016604                                  DODU    UNITN
016604  013700  002200                  MOV     UNITN,R0
016610  104451                          TRAP    C$DODU
016612                                  DOCLN                   ; ABORT THE PASS
```

```
                016612  104444                          TRAP    C$DCLN
2377            016614  000423                          BR      5$
2378
2379            016616                          4$:     RFLAGS  R0              ; GET THE OPERATOR FLAGS.
                016616  104421                          TRAP    C$RFLA
2380            016620  032700  001000                  BIT     #PNT,R0         ; PRINT THE TEST NUMBERS?
2381            016624  001412                          BEQ     1$              ; BR IF NO
2382            016626  011600                          MOV     (SP),R0         ;GET THE ID MESSAGE
2383            016630                                  PRINTF  #TNAM,R0                ;DISPLAY THE TEST ID
                016630  010046                          MOV     R0,-(SP)
                016632  012746  016674                  MOV     #TNAM,-(SP)
                016636  012746  000002                  MOV     #2,-(SP)
                016642  010600                          MOV     SP,R0
                016644  104417                          TRAP    C$PNTF
                016646  062706  000006                  ADD     #6,SP
2384            016652  005237  002212          1$:     INC     TSTCNT          ; BUMP TEST COUNTER.
2385            016656                                  SETPRI  IPRI            ;PRIORITY THAT OF DEVICE
                016656  013700  002210                  MOV     IPRI,R0
                016662  104441                          TRAP    C$SPRI
2386            016664  005726                  5$:     TST     (SP)+           ;FIX UP THE STACK
2387            016666  013705  002204                  MOV     CSRADDR,R5      ; ADDRESS OF TSV REGISTERS ON UNIBUS
2388            016672  000207                          RTS     PC
2389            016674     045     123     045  TNAM:   .ASCIZ  '%S%T%A Test'
2390                                                    .EVEN
```

```
                                        .SBTTL  TSTEND  - PRINT ERRORS RECEIVED
                                ;
                                ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
                                ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
                                ;
016710                          TSTEND: RFLAGS  R0
016710  104421                         TRAP    C$RFLA
016712  030027  020000                 BIT     R0,#IER
016716  001412                         BEQ     1$                      ; BR IF "IER" NOT SET.
016720                                 PRINTF  #ESUM,ERRK              ; PRINT ERROR COUNT.
016720  013746  016746                 MOV     ERRK,-(SP)
016724  012746  016750                 MOV     #ESUM,-(SP)
016730  012746  000002                 MOV     #2,-(SP)
016734  010600                         MOV     SP,R0
016736  104417                         TRAP    C$PNTF
016740  062706  000006                 ADD     #6,SP
016744  000207                  1$:    RTS     PC

016746  000000                  ERRK:   0                               ; LOCAL ERROR COUNT.
016750     045     101     040  ESUM:   .ASCIZ  /%A %D%A ERRORS/
016767     105     122     122  EMAXDU: .ASCIZ  /ERROR LIMIT REACHED -- DROPPING UNIT/
                                        .EVEN

                                        .SBTTL  INCERK  - INCREMENT LOCAL ERROR COUNT
                                ;+
                                ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
                                ;-
017034  005237  016746          INCERK: INC     ERRK                    ; INCREMENT LOCAL ERROR COUNT
017040  010046                         MOV     R0,-(SP)                ; SAVE R0
017042  013700  002200                 MOV     UNITN,R0                ; GET UNIT NUMBER,
017046  006300                         ASL     R0                      ; ... AND MAKE IT A WORD OFFSET.
017050  062700  003172                 ADD     #ERTABL,R0              ; R0 GETS ADDRESS OF ERROR TABLE ENTRY.
017054  005210                         INC     (R0)                    ; INCREMENT THE DEVICE ERROR COUNT
017056  032710  007777                 BIT     #7777,(R0)              ; DID WE OVERFLOW THE FIELD?
017062  001001                         BNE     1$                      ; BR IF NO.
017064  005310                         DEC     (R0)                    ; YES -- BACK IT UP TO 7777.
017066  012600                  1$:    MOV     (SP)+,R0                ; RESTORE R0
017070  000207                         RTS     PC                      ; RETURN TO CALLER.

017072  010046                  CKEMAX: MOV     R0,-(SP)                ; SAVE R0
017074  013700  002200                 MOV     UNITN,R0                ; GET UNIT NUMBER
017100  006300                         ASL     R0                      ; ... AND MAKE IT A WORD OFFSET
017102  016000  003172                 MOV     ERTABL(R0),R0           ; GET ERROR TABLE ENTRY
017106  042700  170000                 BIC     #170000,R0              ; EXTRACT ERROR COUNT FIELD
017112  020037  002172                 CMP     R0,GERRMAX              ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
017116  103004                         BHIS    1$                      ; BR IF YES
017120  023737  016746  002170         CMP     ERRK,LERRMAX            ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
017126  103417                         BLO     2$                      ; BR IF NO
017130                          1$:    RFLAGS  R0                      ; GET OPERATOR FLAGS
017130  104421                         TRAP    C$RFLA
017132  032700  000040                 BIT     #IDU,R0                 ; IS DROPPING INHIBITED?
017136  001013                         BNE     2$                      ; BR IF YES.
017140  012737  177777  003106         MOV     #-1,DUFLG               ; NO -- DROP THE UNIT
017146                                 ERRDF   4,EMAXDU
017146  104455                         TRAP    C$ERDF
017150  000004                         .WORD   4
017152  016767                         .WORD   EMAXDU
```

```
        017154  000000                          .WORD   0
2438    017156                                  DODU    UNITN
        017156  013700  002200                  MOV     UNITN,R0
        017162  104451                          TRAP    C$DODU
2439    017164                                  DOCLN
        017164  104444                          TRAP    C$DCLN
2440    017166  012600                  2$:     MOV     (SP)+,R0        ; RESTORE R0
2441    017170  000207                          RTS     PC              ; RETURN TO CALLER
2442
```

```
                                        .SBTTL  CKDROP  - CHECK IF UNIT SHOULD BE DROPPED
                                ;+
                                ; CHECK IF UNIT SHOULD BE DROPPED
                                ;-
017172  010046                  CKDROP: MOV     R0,-(SP)
017174                                  FORCERROR       1$,NOTSSR
017204                                  RFLAGS  R0
017204  104421                          TRAP    C$RFLA
017206  032700  000040                  BIT     #IDU,R0
017212  001010                          BNE     1$
017214  011600                          MOV     (SP),R0
017216  012737  177777  003106          MOV     #-1,DUFLG
017224                                  DODU    UNITN
017224  013700  002200                  MOV     UNITN,R0
017230  104451                          TRAP    C$DODU
017232                                  DOCLN                   ;ABORT THE PASS
017232  104444                          TRAP    C$DCLN
017234  012600                  1$:     MOV     (SP)+,R0
017236  000207                          RTS     PC
```

```
                                        .SBTTL  CONFIG  - DETERMINE CONFIGURATION OF SYSTEM
                                ;
                                ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
                                ;
017240                          CONFIG:
017240  004737  015764                  JSR     PC,SOFINIT
017244  000207                          RTS     PC
```

```
                                        .SBTTL  KTON,KTOFF      - ENABLE/DISABLE MEMORY MANAGEMENT
                                ;
                                ; SUBROUTINE - ENABLE MEM MGT.
                                ;
017246  005737  003126        KTON:   TST     KTFLG           ; GOT KT?
017252  001403                        BEQ     1$              ; NO.
017254  012737  000001  177572        MOV     #1,SR0          ; YES. ENABLE KT11.
017262  000207                1$:     RTS     PC
                                ;
                                ; SUBROUTINE - DISABLE MEM MGT.
                                ;
017264  005737  003126        KTOFF:  TST     KTFLG           ; GOT KT11?
017270  001405                        BEQ     1$              ; NO.
017272  000240                        NOP
017274  000240                        NOP
017276  012737  000000  177572        MOV     #0,SR0          ; DISABLE KT.
017304  000207                1$:     RTS     PC
```

```
                                .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
                        ;-
017306                  SETMAP:
017306                          SAVREG                          ;SAVE R1-R4 UNTIL NEXT RETURN
017312  005737  003126          TST     KTFLG                   ;SYSTEM HAVE ABOVE 28K?
017316  001433                  BEQ     10$                     ;BR IF NO
017320  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
017352  042701  000177          BIC     #177,R1                 ;ALINE FOR LOWER 4K BOUNDARY
017356  020137  003126          CMP     R1,KTFLG                ;HIGHER THAN EXISTING MEMORY?
017362  103011                  BHIS    10$                     ;BR IF YES
017364  010137  172354          MOV     R1,@#KIPAR6             ;SETUP MAPPING REGISTER PAR6
017370  042702  160000          BIC     #160000,R2              ;SETUP DISPLACEMENT IN PAGE
017374  062702  140000          ADD     #140000,R2              ;ADD IN PAR6 BIAS
017400  010200                  MOV     R2,R0                   ;RETURN IN R0
017402  000261                  SEC                             ;SET SUCCESS
017404  000401                  BR      15$                     ;
017406  000241          10$:    CLC                             ;SET FAILURE
017410  000207          15$:    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
                ;-
                ;
017412          FILLMEM:
017412                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
017416  004737  017264          JSR     PC,KTOFF        ;DISABLE KT.
017422  010003                  MOV     R0,R3           ;COPY TEST PATTERN
017424  013701  003120          MOV     FREE,R1         ;GET FIRST FREE LOCATION
017430  013702  003122          MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
017434  010321          10$:    MOV     R3,(R1)+        ;STORE A BACKGROUND WORD
017436  005302                  DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
017440  003375                  BGT     10$             ;BR IF NO
017442  005737  003126          TST     KTFLG           ; GOT KT?
017446  001477                  BEQ     55$             ; NO. GET OUT.
017450  004737  017246          JSR     PC,KTON         ; YES. ENABLE KT.
017454  005000                  CLR     R0              ;HIGH ORDER ADDRESS START
017456  013701  003146          MOV     PST32W,R1       ;GET >28K START ADDRESS (IN 32W BLOCKS)
        000006                  .REPT   6
                                CLC                     ;CLEAR C BIT
                                ROL     R1              ;CONVERT BLOCKS TO WORDS
                                ROL     R0              ;MAKE IT DOUBLE PRECISION
                                .ENDR
017526  004737  017306          JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
017532  010320          30$:    MOV     R3,(R0)+        ;STORE TEST PATTERN IN >28K ADDRESS
017534  020027  160000          CMP     R0,#160000      ;END OF PAR6 MAPPING AREA?
017540  103774                  BLO     30$             ;BR IF NO
017542  162700  020000          SUB     #20000,R0       ;BACKUP INTO PAR6 MAPPING BEGIN
017546  062737  000200  172354  ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
017554  023737  172354  003126  CMP     @#KIPAR6,KTFLG  ;END OF MEMORY?
017562  001427                  BEQ     50$             ;BR IF YES
017564  005737  003140          TST     T23A            ;11/23A?
017570  001407                  BEQ     35$             ;NO KEEP GOING
017572  013704  177572          MOV     SR0,R4          ;GET SR0 CONTENTS
017576  042704  177761          BIC     #177761,R4      ;CLEAR ALL BUT PAGE NUMBER
017602  022704  000016          CMP     #16,R4          ;SEE IF PAGE 7
017606  001415                  BEQ     50$             ;EXIT IF THERE
017610  005737  003142  35$:    TST     T23B            ;11/23B?
017614  001410                  BEQ     45$             ;NO KEEP GOING
017616  023727  172354  007600  CMP     @#KIPAR6,#7600  ;REACHED 18 BITS?
017624  103001                  BHIS    40$             ;YES
017626  000403                  BR      45$             ;NO KEEP GOING
017630  012737  000020  172516  40$:    MOV     #20,SR3         ;SET 22 BIT RELOCATION
017636  000137  017532          45$:    JMP     30$             ;KEEP GOING ON ETC.
017642  004737  017264          50$:    JSR     PC,KTOFF        ; DISABLE KT.
017646  000207                  55$:    RTS     PC
```

```
2595
2596
```

```
                        .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
                ;-
017650          CMPMEM:
017650                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
017654  010003          MOV     R0,R3           ;COPY TEST PATTERN
017656  004737  017264  JSR     PC,KTOFF        ;DISABLE KT.
017662  013701  003120  MOV     FREE,R1         ;GET FIRST FREE LOCATION
017666  013702  003122  MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
017672  020311          10$:    CMP     R3,(R1) ;FREE SPACE LOCATION EQUAL TO EXPD?
017674  001411          BEQ     15$             ;BR IF YES
017676  010137  002236  MOV     R1,ERRLO        ;SAVE ADDRESS IN ERROR
017702  005037  002234  CLR     ERRHI           ;NO HIGH ADDRESS
017706  010337  002230  MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
017712  011137  002232  MOV     (R1),RECV       ;SAVE RECV FOR ERROR REPORT
017716  000474          BR      50$             ;
017720  005721          15$:    TST     (R1)+   ;POINT TO NEXT ADDRESS
017722  005302          DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
017724  003362          BGT     10$             ;BR IF NO
017726  005737  003126  TST     KTFLG           ; GOT KT?
017732  001472          BEQ     55$             ; NO. GET OUT.
017734  004737  017246  JSR     PC,KTON         ; YES. ENABLE KT.
017740  005000          CLR     R0              ;HIGH ORDER ADDRESS START
017742  013701  003146  MOV     PST32W,R1       ;GET >28K START ADDRESS (IN 32W BLOCKS)
                        .REPT   6
                        ROL     R1              ;CONVERT BLOCKS TO WORDS
                        ROL     R0              ;MAKE IT DOUBLE PRECISION
                        .ENDR
017776  042701  000177  BIC     #177,R1         ;ALINE 4K BOUNDARY
020002  010046          MOV     R0,-(SP)        ;SAVE HIGH ORDER
020004  010146          MOV     R1,-(SP)        ;SAVE LOW ORDER
020006  004737  017306  JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
020012  010004          MOV     R0,R4           ;COPY ADDRESS BIASED TO PAR6
020014  012601          MOV     (SP)+,R1        ;RESTORE LOW ORDER IN NON PAR6 FORMAT
020016  012600          MOV     (SP)+,R0        ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
020020  020314          30$:    CMP     R3,(R4) ;ABOVE 28K LOCATION EQUAL EXPD?
020022  001411          BEQ     32$             ;BR IF YES
020024  010037  002234  MOV     R0,ERRHI        ;SAVE HIGH ORDER IN ERROR
```

```
2655 020030  010137  002236                MOV   R1,ERRLO          ;SAVE LOW ORDER IN ERROR
2656 020034  010337  002230                MOV   R3,EXPD           ;SAVE EXPD FOR ERROR REPORT
2657 020040  011437  002232                MOV   (R4),RECV         ;SAVE RECV FOR ERROR REPORT
2658 020044  000421                        BR    50$               ;
2659 020046  062701  000002          32$:  ADD   #2,R1             ;UPDATE NON PAR6 ADDRESS
2660 020052  005500                        ADC   R0                ;MAKE IT DOUBLE PRECISION ADD
2661 020054  062704  000002                ADD   #2,R4             ;UPDATE PAR FORMAT ADDRESS
2662 020060  020427  160000                CMP   R4,#160000        ;END OF PAR6 MAPPING AREA?
2663 020064  103755                        BLO   30$               ;BR IF NO
2664 020066  162704  020000                SUB   #20000,R4         ;BACKUP INTO PAR6 MAPPING BEGIN
2665 020072  062737  000200  172354        ADD   #200,@#KIPAR6     ;POINT TO NEXT 4K BLOCK >28K.
2666 020100  023737  172354  003126        CMP   @#KIPAR6,KTFLG    ;END OF MEMORY?
2667 020106  101744                        BLOS  30$               ;BR IF NO
2668 020110  004737  017264          50$:  JSR   PC,KTOFF          ;TURN OFF MEMORY MAPPING
2669 020114  000241                        CLC                     ;SET FAILURE
2670 020116  000403                        BR    60$               ;
2671 020120  004737  017264          55$:  JSR   PC,KTOFF          ;TURN OFF MEMORY MAPPING
2672 020124  000261                        SEC                     ;SET SUCCESS
2673 020126  000207                  60$:  RTS   PC
2674
```

```
                                .SBTTL  REGSAV  - SAVE R1-R5 ON STACK
                        ;+
                        ;
                        ;ROUTINE TO
                        ;SAVE R1 THROUGH R5 ON THE STACK
                        ;
                        ;CALLING SEQUENCE:
                        ;
                        ;       JSR     R5,REGSAV
                        ;
                        ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
                        ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
                        ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
                        ;REGISTERS.
                        ;
                        ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
                        ;CALLED VIA A JSR PC INSTRUCTION
                        ;
                        ;-
020130                  REGSAV:
020130  010446                  MOV     R4,-(SP)
020132  010346                  MOV     R3,-(SP)
020134  010246                  MOV     R2,-(SP)
020136  010146                  MOV     R1,-(SP)
020140  010546                  MOV     R5,-(SP)
020142  016605  000012          MOV     10.(SP),R5
020146  004736                  JSR     PC,@(SP)+
020150  012601                  MOV     (SP)+,R1
020152  012602                  MOV     (SP)+,R2
020154  012603                  MOV     (SP)+,R3
020156  012604                  MOV     (SP)+,R4
020160  012605                  MOV     (SP)+,R5
020162  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
                ;
                ;-
                GETPAT::
                        SAVREG                          ;SAVE THE GENERAL REGISTERS
020164          1$:     GMANID  DATASC,PATDAT,0,377,0,377,NO
020170  104443          TRAP    C$GMAN
020172  000406          BR      10000$
020174  020220          .WORD   PATDAT
020176  000022          .WORD   T$CODE
020200  020222          .WORD   DATASC
020202  000377          .WORD   377
020204  000000          .WORD   T$LOLIM
020206  000377          .WORD   T$HILIM
020210                  10000$:
020210                          BNCOMPLETE      1$      ;RETRY IF ERROR
020210  103367          BCC     1$
020212  013700  020220  MOV     PATDAT,R0               ;DATA PATTERN FROM OPERATOR
020216  000207          RTS     PC                      ;RETURN TO CALLER

                ;+
                ;LOCAL DATA AREA
                ;-

020220  000000          PATDAT: .WORD   0               ;TEMPORARY STORAGE FOR DATA
020222  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
                ;
                ;-

                GETSEL::
                        SAVREG                          ;SAVE GENERAL REGISTERS
                        MOV     R0,R2                   ;SAVE THE MENU ADDRESS
                1$:     MOV     R2,R3                   ;START OF MENU STRING
                2$:     TST     (R3)                    ;END OF ASCII ?
                        BEQ     3$                      ;BRANCH IF ALL LINES DISPLAYED
                        PRINTF  #SELASC,(R3)+           ;DISPLAY THE MENU
020262  012346          MOV     (R3)+,-(SP)
020264  012746  020432  MOV     #SELASC,-(SP)
020270  012746  000002  MOV     #2,-(SP)
020274  010600          MOV     SP,R0
020276  104417          TRAP    C$PNTF
020300  062706  000006  ADD     #6,SP
020304  000764          BR      2$
020306                  GMANID  MENASC,MENRES,D,-1,0,-1,NO
020306  104443          TRAP    C$GMAN
020310  000406          BR      10001$
020312  020466          .WORD   MENRES
020314  000042          .WORD   T$CODE
020316  020437          .WORD   MENASC
020320  177777          .WORD   -1
020322  000000          .WORD   T$LOLIM
020324  177777          .WORD   T$HILIM
020326                  10001$:
020326                  BNCOMPLETE      1%              ;RETRY IF ERROR
020326  103352          BCC     1$
020330  013700  020466  MOV     MENRES,R0               ;GET THE OPERATOR'S REPLY
020334  020001          CMP     R0,R1                   ;COMPARE TO MAXIMUM ALLOWED
020336  101411          BLOS    5$                      ;BRANCH IF OK
020340                  PRINTF  #MENERR                 ;DISPLAY ERROR MESSAGE
020340  012746  020364  MOV     #MENERR,-(SP)
020344  012746  000001  MOV     #1,-(SP)
020350  010600          MOV     SP,R0
020352  104417          TRAP    C$PNTF
020354  062706  000004  ADD     #4,SP
020360  000735          BR      1$                      ;RETRY
020362  000207          5$:     RTS     PC              ;RETURN TO CALLER
020364    045   116   045  MENERR: .ASCIZ  '%N%A *** Menu Selection Too Large ***'
020432    045   116   045  SELASC: .ASCIZ  '%N%T'
020437    105   156   164  MENASC: .ASCIZ  'Enter Menu Selection: '
```

```
 2782
 2783 020466  000000                    MENRES:  .EVEN
                                                .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.
                ;
                ;-

020470          CHKMAN::
                        SAVREG                  ;SAVE THE REGISTERS
                        MANUAL                  ;SEE IF MANUAL INTERVENTION OK
020474  104450          TRAP    C$MANI
                        BCOMPLETE 1$            ;BRANCH IF ALLOWED
020476  103411          BCS     1$
020500                  PRINTF  #NOMAN          ;PRINT THE WARNING MESSAGE
020500  012746  020524          MOV     #NOMAN,-(SP)
020504  012746  000001          MOV     #1,-(SP)
020510  010600                  MOV     SP,R0
020512  104417                  TRAP    C$PNTF
020514  062706  000004          ADD     #4,SP
020520  000241                  CLC                     ;CLEAR CARRY FOR ERROR
020522  000207          1$:     RTS     PC              ;RETURN

020524      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.
                ;
020620          ENVIRN: MEMORY  R0
020620 104431           TRAP    C$MEM
020622 010037 003120    MOV     R0,FREE         ; GET 1ST FREE ADDRESS...
020626 062737 000002 003120     ADD     #2,FREE
020634 011037 003122    MOV     (R0),FRESIZ     ;...AND WORD COUNT.
020640 162737 000004 003122     SUB     #4,FRESIZ
020646 013702 002012    MOV     L$UNIT,R2       ; GET NUMBER OF UNITS
020652 162737 000007 003122 10$:        SUB     #7,FRESIZ       ; TAKE AWAY 7 WORDS PER UNIT
020660 005302           DEC     R2
020662 001373           BNE     10$
020664 013700 003120    MOV     FREE,R0         ;GET FIRST FREE ADDRESS
020670 063700 003122    ADD     FRESIZ,R0       ;POINT TO LAST FREE ADDRESS
020674 162700 000002    SUB     #2,R0           ;BACKUP 1 WORD
020700 010037 003124    MOV     R0,FREEHI       ;STORE LAST FREE ADDRESS
020704 000240           NOP                     ;*****************************************
020706 012701 177520    MOV     #BDVPCR,R1      ;GET BDV11 PCR ADDRESS
020712 010102           MOV     R1,R2           ;COPY TO R2
020714 062702 000002    ADD     #2,R2           ;SET THE RANGE
020720 004737 016366    JSR     PC,XNXM         ;SEE IF WE HAVE ONE
020724 103001           BCC     15$             ;OK TO SET FLAGS
020726 000445           BR      40$             ;RETURN WITH FLAGS CLEAR
020730 013701 177520 15$:       MOV     BDVPCR,R1       ;SAVE PCR CONTENTS
020734 062701 000001    ADD     #1,R1           ;ADD ONE TO IT
020740 012702 177520    MOV     #BDVPCR,R2      ;GET BDV11 PCR ADDRESS
020744 005212           INC     (R2)            ;TRY TO WRITE TO IT
020746 013703 177520    MOV     BDVPCR,R3       ;GET RESULTS
020752 020103           CMP     R1,R3           ;DID IT CHANGE?
020754 001017           BNE     20$             ;NO, MUST BE 11/23B
020756 005237 003140    INC     T23A            ;SET THE FLAG
020762 042737 170000 002120     BIC     #170000,L$HIME  ;SUPERVISOR COULD BE WRONG
020770 000240           NOP                     ;BR 40$ FOR RELEASE
020772                  PRINTF  #M8186          ;TELL THE SYSTEM TYPE
020772 012746 005550    MOV     #M8186,-(SP)
020776 012746 000001    MOV     #1,-(SP)
021002 010600           MOV     SP,R0
021004 104417           TRAP    C$PNTF
021006 062706 000004    ADD     #4,SP
021012 000413           BR      40$             ;RETURN
021014 005237 003142 20$:       INC     T23B            ;SET THE FLAG
021020 000240           NOP                     ;BR 40$ FOR RELEASE
021022                  PRINTF  #M8189          ;TELL THE SYSTEM TYPE
021022 012746 005641    MOV     #M8189,-(SP)
021026 012746 000001    MOV     #1,-(SP)
021032 010600           MOV     SP,R0
021034 104417           TRAP    C$PNTF
021036 062706 000004    ADD     #4,SP
021042 000207 40$:      RTS     PC              ;RETURN
```

```
                                .SBTTL  KTINIT  - SETUP KT11 MEMORY MANAGEMENT REGISTERS
                        ;+
                        ;
                        ;ROUTINE TO INIT KT-11
                        ;
                        ;-

021044                  KTINIT:
021044  005037  003126          CLR     KTFLG           ; INIT >28K MEMORY FLAG
021050  005037  003130          CLR     KTENABLE        ; INIT TEST >28K FLAG
021054  023727  002120  001577  CMP     L$HIME,#1577    ; GOT ENOUGH MEMORY (>28K)?
021062  101444                  BLOS    9$              ; NO.
021064  013700  000004          MOV     @#ERRVEC,R0     ; SAVE OLD ERR VEC PTR.
021070  012737  021162  000004  MOV     #2$,@#ERRVEC    ; SET ERR VEC PTR.
021076  005737  177572          TST     @#SR0           ; GOT KT11?
021102  000240                  NOP                     ;  (TRAP IF NO).
021104  013737  002120  003126  MOV     L$HIME,KTFLG    ; YES. SET KT FLAG.
021112  042737  000177  003126  BIC     #177,KTFLG      ;
021120  010037  000004          MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
021124  005000                  CLR     R0              ; R0 = AR DATA.
021126  012701  172340          MOV     #KIPAR0,R1      ; R1 = KI REGS PTR.
021132  012761  077406  177740  1$:     MOV     #77406,-40(R1)  ; SET DESCRIPTOR REG.
021140  010021                  MOV     R0,(R1)+        ; SET KIPAR REG.
021142  062700  000200          ADD     #200,R0         ; BUMP AR DATA BY "4K".
021146  020027  002000          CMP     R0,#2000        ; AT "I/O"?
021152  001367                  BNE     1$              ; NO.
021154  012741  177600          MOV     #177600,-(R1)   ; YES. SET KTPAR7 FOR I/O.
021160  000405                  BR      9$

021162  012716  021170  2$:     MOV     #6$,(SP)        ; SET UP RETURN
021166  000002                  RTI                     ; RTI TO NEXT LOCATION

021170  010037  000004  6$:     MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.

021174  000207          9$:     RTS     PC
```

```
2897                                    ;+
2898                                    ;       SUBROUTINE TO SET EXTENDED FEATURES SWITCH
2899                                    ;
2900                                    ;       Requires that SOFINIT and WRTCHR have been done previous to call.
2901                                    ;
2902                                    ;
2903                                    ;INPUTS:
2904                                    ;       R5      CURRENT UNIT NUMBER
2905                                    ;OUTPUTS:
2906                                    ;       The Extended Features Switch is set.
2907                                    ;
2908                                    ;-
2909
2910 021176                            INVERT::
2911
2912 021176  005737  002224                    TST     EXTFEA                  ; IS SWITCH SET?
2913 021202  001020                            BNE     1$                      ; YES,EXIT STAGE RIGHT!(or the next one outa town!)
2914 021204  012737  100206  021250            MOV     #100206,CMDPKT          ; WRT SUB-SYS MEM CMD
2915 021212  012737  021260  021252            MOV     #WSMBK,CMDPKT+2         ; MSG BUF ADDR
2916 021220  012737  000006  021256            MOV     #6,CMDPKT+6             ; BYTE COUNT
2917 021226  012737  100010  021260            MOV     #100010,WSMBK           ; INVERT THE SWITCH
2918 021234  012704  021250                    MOV     #CMDPKT,R4              ; SET CMDPKT INTO R4
2919 021240  004737  010652                    JSR     PC,WRTCHR               ; DO IT
2920 021244  000207                    1$:     RTS     PC                      ; RETURN
2921
2922
2923                                    ;       COMMAND PACKET.
2924
2925         021250                     .       =       <.+3>&177774    ;MUST BE ON MOD 4 BOUNDRY.
2926
2927 021250  000000                    CMDPKT:: 0                      ;1ST WORD IS TSO5 COMMAND.
2928 021252  000000                             0                      ;2ND WORD IS THE BUFFER LOW ADDRESS.
2929 021254  000000                             0                      ;3RD WORD IS THE BUFFER HIGH ADDRESS.
2930 021256  000000                             0                      ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
2931
2932
2933                                    ;       WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
2934
2935 021260  000000                    WSMBK::  0                      ;1ST WORD:: SEL 0
2936 021262  000000                             0                      ;2ND WORD:: SEL 2
2937 021264  000000                             0                      ;3RD WORD:: SEL 4
2938                                            .EVEN
2939
```

```
                                        ;+
                                        ;       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.
                                        ;-

021266                                  MEMCK::

021266                                          SAVREG                          ;SAVE THE REGISTERS
021272  005037  003132                          CLR     NXMFLG                  ;CLEAR THE FLAG
021276  005037  003134                          CLR     NXMLO                   ;CLEAR THE TEST ADDRESS LO
021302  005037  003136                          CLR     NXMHI                   ;CLEAR THE TEST ADDRESS HI
021306  005737  003142                          TST     T23B                    ;IS IT A 11/23B?
021312  001407                                  BEQ     1$                      ;NO
021314  023727  002120  007777                  CMP     L$HIME,#7777            ; GREATER THAN 128K
021322  103406                                  BLO     2$                      ; NO
021324  004737  021442                          JSR     PC,NXMTST               ;SETUP THE ADDRESS
021330  000427                                  BR      13$                     ;SET THE FLAG AND EXIT
021332  005737  003140          1$:             TST     T23A                    ;IS IT A 11/23A?
021336  001413                                  BEQ     4$                      ;NO
021340  023727  002120  005777  2$:             CMP     L$HIME,#5777            ;GREATER THAN 96K
021346  101023                                  BHI     14$                     ;YES,23A/23B WITH 128K MEMORY
021350  023727  002120  003777                  CMP     L$HIME,#3777            ;GREATER THAN 64K BUT LESS THAN 92K?
021356  103403                                  BLO     4$                      ;NO, CHECK 24K
021360  004737  021442                          JSR     PC,NXMTST               ;SETUP THE ADDRESS
021364  000411                                  BR      13$                     ;SET THE FLAG AND EXIT
021366  023727  002120  001577  4$:             CMP     L$HIME,#1577            ;GREATER THAN 24K BUT LESS THAN 64K?
021374  103410                                  BLO     14$                     ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
021376  004737  021442                          JSR     PC,NXMTST               ;SETUP THE ADDRESS
021402  062737  000077  003136                  ADD     #77,NXMHI               ;FOOL THE 11/02 & 11/03
021410  005237  003132          13$:            INC     NXMFLG                  ;SET THE FLAG
021414  000411                                  BR      15$                     ;EXIT
021416  000410                  14$:            BR      15$                     ;NOP FOR PRINTOUT
021420                                          PRINTF  #NOMEM                  ;TELL THEM & EXIT ***NO PRINT*****
021420  012746  005454                          MOV     #NOMEM,-(SP)
021424  012746  000001                          MOV     #1,-(SP)
021430  010600                                  MOV     SP,R0
021432  104417                                  TRAP    C$PNTF
021434  062706  000004                          ADD     #4,SP
021440  000207                  15$:            RTS     PC                      ;RETURN

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

021442  013701  002120          NXMTST: MOV     L$HIME,R1                       ;GET TOP OF MEMORY
021446  062701  000200                  ADD     #200,R1                         ;MAKE IT I/O BLOCK OR OTHER NXM
021452  042701  000177                  BIC     #177,R1
021456  010102                          MOV     R1,R2                           ;RESAVE RESULTS
```

```
                000006                          .REPT   6
                                                ASL     R1              ;PUT IN PLACE FOR XFER
                                                .ENDR
021474  010137  003134                          MOV     R1,NXMLO        ;SAVE TEST ADDRESS LOW
                000012                          .REPT   10.
                                                ASR     R2              ;PUT IN PLACE FOR XFER
                                                .ENDR
021524  042702  177700                          BIC     #177700,R2      ;DON'T WANT ILA!
021530  010237  003136                          MOV     R2,NXMHI        ;SAVE TEST ADDRESS HIGH
021534  000207                                  RTS     PC              ;RETURN


021536                                          ENDMOD
```

```
                                                J 9
                                .TITLE  TSV4 - MISCELLANEOUS SECTIONS

                                BGNMOD  TSV4
021536                  TSV4::
021536
```

```
                                        .SBTTL  PROTECTION TABLE
021536                                  BGNPROT
021536                          L$PROT::
021536  177777  177777  177777        .WORD   -1, -1, -1, -1          ;NO DEVICE PROTECTION REQUIRED.
021546                                  ENDPROT
```

```
                        .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
                ;-
021546                  BGNINIT
021546          L$INIT::
021546 005037 002224    40$:    CLR     EXTFEA
021552 005037 003132            CLR     NXMFLG
021556 012737 006354 002176     MOV     #EPRT1,EPRTSW           ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
021564 005037 003150            CLR     SIFLAG                  ;CLEAR "SOFT INIT" FLAG
021570 005037 003130            CLR     KTENABLE                ;CLEAR TEST ABOVE 28K FLAG
021574 005037 002300            CLR     RAMSIZ                  ;CLEAR RAM SIZE FOR RAMERR ROUTINE
021600                          READEF  #EF.CONTINUE
021600 012700 000036            MOV     #EF.CONTINUE,R0
021604 104447                   TRAP    C$REFG
021606                          BNCOMPLETE 1$
021606 103023                   BCC     1$
021610 023737 002200 002012     CMP     UNITN,L$UNIT            ;UNIT IN RANGE?
021616 103070                   BHIS    4$                      ;BR IF NO.
021620 005737 003106            TST     DUFLG                   ;DROPPED UNIT?
021624 100472                   BMI     NXTU                    ;BR IF YES
021626 013701 002200            MOV     UNITN,R1
021632 006301                   ASL     R1
021634 005761 003172            TST     ERTABL(R1)
021640 001516                   BEQ     SETU
021642 032761 040000 003172     BIT     #BIT14,ERTABL(R1)       ;DROPPED?
021650 001060                   BNE     NXTU
021652                          EXIT    INIT                    ;DO NOTHING IF "CONTINUE".
021652 104432                   TRAP    C$EXIT
021654 000416                   .WORD   L10030-.
021656                  1$:     READEF  #EF.NEW
021656 012700 000035            MOV     #EF.NEW,R0
021662 104447                   TRAP    C$REFG
021664                          BNCOMPLETE NXTU                 ;TAKE NEXT UNIT IF NOT NEW PASS.
021664 103052                   BCC     NXTU
021666                          READEF  #EF.START
021666 012700 000040            MOV     #EF.START,R0
021672 104447                   TRAP    C$REFG
021674                          BCOMPLETE 2$
021674 103404                   BCS     2$
021676                          READEF  #EF.RESTART
021676 012700 000037            MOV     #EF.RESTART,R0
021702 104447                   TRAP    C$REFG
021704                          BNCOMPLETE 31$
021704 103031                   BCC     31$
021706                  2$:                                     ;1ST PASS, BUS-INIT...
021706                          BRESET                          ;BUS RESET.
021706 104433                   TRAP    C$RESET
```

```
                                        CLR     TSTCNT                  ;NUMBER OF TESTS RUN IN PASS
                                        CLR     FATFLG                  ;CLEAR FATAL ERROR COUNT
                                        CLR     T23A                    ;CLEAR 11/23A FLAG
                                        CLR     T23B                    ;CLEAR 11/23B FLAG
                                ;       MOV     #340,-(SP)
                                ;       MOV     #20$,-(SP)              ;RETURN TO DEBUGGER
                                ;       JMP     0.ODT           ;;@@ENTER THE DEBUGGER
021730  005037  003374                  CLR     SKIPT           ;CLEAR THE SUBTEST "SKIPPER"
                                20$:
021734  012737  177777  002202          MOV     #-1,QVP                 ;...QUICK VERIFY...
021742  004737  020620                  JSR     PC,ENVIRN               ;SET ENVIRONMENT.
021746  004737  021044                  JSR     PC,KTINIT               ;INITIALIZE KT MEMORY MANAGEMENT
021752  012700  003172                  MOV     #ERTABL,R0
021756  005020                  30$:    CLR     (R0)+                   ;CLEAR THE ERROR TABLE
021760  020027  003372                  CMP     R0,#ERTABE
021764  103774                          BLO     30$
021766  000404                          BR      4$
021770  005037  002202          31$:    CLR     QVP
021774  000137  022044                  JMP     PASRPT                  ;GO REPORT THE STATUS

022000                          4$:
022000  012737  177777  002200  NEWPAS: MOV     #-1,UNITN               ;INIT UNIT NUMBER...
022006  005037  002216                  CLR     DEVCNT                  ;CLEAR COUNT OF DEVICES RUNNING
022012                          NXTU:   BREAK
022012  104422                          TRAP    C$BRK
022014  005237  002200                  INC     UNITN                   ;...AND SET NEXT UNIT NUMBER.
022020  023737  002200  002012          CMP     UNITN,L$UNIT
022026  103423                          BLO     SETU
022030  012737  177777  003106          MOV     #-1,DUFLG
022036  000401                          BR      11$
022040                                  DOCLN
022040  104444                          TRAP    C$DCLN                  ;ABORT, NO MORE UNITS.
022042  000240                  11$:    NOP
022044                          PASRPT:
022044  023727  002012  000001          CMP     L$UNIT,#1               ;HOW MANY UNITS SELECTED?
022052  101752                          BLOS    NEWPAS                  ;BR IF ONLY 1
022054  005737  002216                  TST     DEVCNT                  ;ARE ANY STILL RUNNING?
022060  001747                          BEQ     NEWPAS                  ;BR IF NO
022062                                  RFLAGS  R0
022062  104421                          TRAP    C$RFLA
022064  032700  000100                  BIT     #ISR,R0                 ;SHOULD WE PRINT STATISTICS
022070  001343                          BNE     NEWPAS                  ;BR IF NO

022072                                  DORPT
022072  104424                          TRAP    C$DRPT
022074  000741                          BR      NEWPAS
022076                          10$:

022076                          SETU:   GPHARD  UNITN,R0                ;GET UNIT N P-TABLE POINTER.
022076  013700  002200                  MOV     UNITN,R0
022102  104442                          TRAP    C$GPHRD
022104                                  BNCOMPLETE NXTU                 ;BR IF UNIT NOT AVAILABLE.
022104  103342                          BCC     NXTU
022106  005037  003106                  CLR     DUFLG                   ;CLEAR "DROPPED" FLAG.
022112  005237  002216                  INC     DEVCNT
022116  012001                          MOV     (R0)+,R1                ;GET 1ST REGISTER ADDRESS.
022120  010137  002204                  MOV     R1,CSRADDR              ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
```

```
                                                ;GET VECTOR ADDRESS.
                        MOV     (R0)+,R1
                        ;MOV    (R0),R2         ;GET INTERRUPT PRIORITY
                        ;MOV    R2,IPRI         ;SET INTERRUPT PRIORITY.
022126  010137  002206  MOV     R1,IVEC         ;SET INTERRUPT VECTOR POINTER...
022132  012721  016206  MOV     #INTR,(R1)+     ;...VECTOR...
022136  013721  002210  MOV     IPRI,(R1)+      ;...AND PRIORITY.

022142                  1$:
                        ;       TST     QVP             ;1ST PASS ??
                        ;       BEQ     5$              ;NO, SKIP THE PASS 1 STUFF.

                        ;
                        ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
                        ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
                        ;
022142  013701  002200          MOV     UNITN,R1
022146  006301                  ASL     R1
022150  052761  100000  003172  BIS     #BIT15,ERTABL(R1)       ;SAY DEVICE RUNNING
022156  005037  005766          CLR     EXTA                    ;CLEAR ERROR EXTENSION FLAG.
022162  023727  002012  000001  CMP     L$UNIT,#1               ;ARE WE TESTING MULTIPLE UNITS?
022170  101416                  BLOS    10$                     ;BR IF NO.
022172                          RFLAGS  R0                      ;YES -- GET OPERATOR FLAGS.
022172  104421                  TRAP    C$RFLA
022174  032700  001000          BIT     #PNT,R0                 ;SHOULD WE PRINT UNIT #?
022200  001412                  BEQ     10$                     ;BR IF NOT.
022202                          PRINTF  #PUNIT,UNITN            ;PRINT THE UNIT #
022202  013746  002200          MOV     UNITN,-(SP)
022206  012746  022274          MOV     #PUNIT,-(SP)
022212  012746  000002          MOV     #2,-(SP)
022216  010600                  MOV     SP,R0
022220  104417                  TRAP    C$PNTF
022222  062706  000006          ADD     #6,SP
022226                  10$:
022226  005037  003110          CLR     NODEV
022232  013701  002204          MOV     CSRADDR,R1              ;ADDRESS OF FIRST REGISTER
022236  010102                  MOV     R1,R2                   ;START OF REGISTERS
022240  062702  000002          ADD     #TSSR,R2                ;ADDRESS OF TSSR REGISTER
022244  004737  016366          JSR     PC,XNXM                 ;TEST BOTH CONTROLLER REGISTERS...
022250  103005                  BCC     2$                      ;...AND BR IF ALL OK.
022252  010137  003110          MOV     R1,NODEV                ;FLAG DEVICE AS NON-EXISTENT
022256  012737  177777  003106  MOV     #-1,DUFLG               ;DROP THIS UNIT.
022264                  2$:
                        ;
                        ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
                        ;
022264                  5$:     SETPRI  #PRI00          ;ENABLE INTERRUPTS.
022264  012700  000000          MOV     #PRI00,R0
022270  104441                  TRAP    C$SPRI
022272                          ENDINIT
022272                  L10030:
022272  104411                  TRAP    C$INIT

022274      045     116     045 PUNIT:  .ASCIZ  /%N%N%A***** TESTING UNIT %D2%A *****/
                                .EVEN
```

```
                        .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.
                ;--
022342                  BGNAU
022342          L$AU::
022342 010001           MOV     R0,R1                   ; GET UNIT TO BE ADDED (R0)
022344 006301           ASL     R1                      ; MAKE IT A WORD INDEX
022346 052761 100000 003172     BIS     #100000,ERTABL(R1)      ; SET THE "ACTIVE" BIT
022354 042761 040000 003172     BIC     #40000,ERTABL(R1)       ; CLEAR THE "DROPPED" BIT
022362                  PRINTF  #1$,R0
022362 010046           MOV     R0,-(SP)
022364 012746 022410    MOV     #1$,-(SP)
022370 012746 000002    MOV     #2,-(SP)
022374 010600           MOV     SP,R0
022376 104417           TRAP    C$PNTF
022400 062706 000006    ADD     #6,SP
022404                  EXIT    AU
022404 000167           .WORD   J$JMP
022406 000026           .WORD   L10031-2-.
022410   045   116   045  1$:  .ASCIZ  /%N%A  UNIT %D%A ADDED/
                        .EVEN

022436                  ENDAU           ; UNUSED.
022436          L10031:
022436 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.

022440                  BGNDU
022440          L$DU::
022440 012737 177777 003106     MOV     #-1,DUFLG
022446 010001           MOV     R0,R1
022450 006301           ASL     R1
022452 052761 140000 003172     BIS     #140000,ERTABL(R1)      ; SAY DROPPED
022460 000240 000240 000240     240,240,240                     ; ?????????
022466                  PRINTF  #1$,R0
022466 010046           MOV     R0,-(SP)
022470 012746 022514    MOV     #1$,-(SP)
022474 012746 000002    MOV     #2,-(SP)
022500 010600           MOV     SP,R0
022502 104417           TRAP    C$PNTF
022504 062706 000006    ADD     #6,SP
022510                  EXIT    DU
022510 000167           .WORD   J$JMP
022512 000030           .WORD   L10032-2-.
```

```
                                1$:     .ASCIZ  /%N%A  UNIT %D%A DROPPED/
                                        .EVEN
                                        ENDDU
                                L10032:
                                        TRAP    C$DU
                                ;++
                                ; AUTO-DROP CODE SECTION.
                                ;--
                                        BGNAUTO
                                L$AUTO::
                                        MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
                                        MOV     #360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
                                10$:    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
                                        BCS     20$                     ;LEAVE WHEN SSR IS SET
                                        DELAY   250.                    ;WAIT FOR .25 SECONDS
                                        MOV     #250.,(PC)+
                                        .WORD   0
                                        MOV     L$DLY,(PC)+
                                        .WORD   0
                                        DEC     -6(PC)
                                        BNE     .-4
                                        DEC     -22(PC)
                                        BNE     .-20
                                        DEC     R3                      ;BUMP COUNTER DOWN
                                        BNE     10$                     ;KEEP GOING
                                        JSR     PC,CKDROP               ;TRY AND DROP UNIT
                                20$:
                                        ENDAUTO                 ; UNUSED.
                                L10033:
                                        TRAP    C$AUTO
```

```
                                        .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.
                                ;--
022626                                  BGNCLN
022626                          L$CLEAN::
022626  013705  002204                  MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
022632  005737  003106                  TST     DUFLG                   ;"DROPPED" FLAG IS SET ON...
022636  100405                          BMI     1$                      ;...AND GROSS CONTROLLER FAULT...
                                                                        ;...DON'T TRY TO XCT CLEANUP CODE.

022640  012765  000000  000002          MOV     #0,TSSR(R5)             ;DO SOFT INIT
022646  004737  016240                  JSR     PC,WAITF
022652                          1$:
022652                          2$:     ENDCLN
022652                          L10034:
022652  104412                          TRAP    C$CLEAN
                                ;++
                                ; THE REPORT CODING SECTION CONTAINS THE
                                ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;--
022654                                  BGNRPT
022654                          L$RPT::
022654                                  PRINTS  #DEVSUM
022654  012746  023116                  MOV     #DEVSUM,-(SP)
022660  012746  000001                  MOV     #1,-(SP)
022664  010600                          MOV     SP,R0
022666  104416                          TRAP    C$PNTS
022670  062706  000004                  ADD     #4,SP
022674  010246                          MOV     R2,-(SP)
022676  010346                          MOV     R3,-(SP)
022700  010446                          MOV     R4,-(SP)
022702  012704  003172                  MOV     #ERTABL,R4              ; GET START OF ERROR TABLE.
022706  005003                          CLR     R3                      ; CLEAR UNIT NUMBER
022710  011402                  1$:     MOV     (R4),R2                 ; GET ERROR TABLE ENTRY & TEST IT.
022712  001467                          BEQ     4$                      ; ZERO IF UNIT NOT RUN
022714  100066                          BPL     4$
022716  032702  040000                  BIT     #BIT14,R2               ; WAS UNIT DROPPED?
022722  001015                          BNE     2$                      ; BR IF YES
022724  042702  170000                  BIC     #^C7777,R2              ; GET ERROR COUNT FIELD
022730                                  PRINTS  #DEVONL,R3,R2           ; PRINT
022730  010246                          MOV     R2,-(SP)
022732  010346                          MOV     R3,-(SP)
022734  012746  023153                  MOV     #DEVONL,-(SP)
022740  012746  000003                  MOV     #3,-(SP)
022744  010600                          MOV     SP,R0
022746  104416                          TRAP    C$PNTS
022750  062706  000010                  ADD     #10,SP
022754  000446                          BR      4$
022756  020227  160000          2$:     CMP     R2,#160000              ; WAS UNIT NON-EXISTENT?
022762  001012                          BNE     3$                      ; BR IF NO
022764                                  PRINTS  #DEVNXR,R3
022764  010346                          MOV     R3,-(SP)
022766  012746  023223                  MOV     #DEVNXR,-(SP)
```

```
                022772  012746  000002          MOV     #2,-(SP)
                022776  010600                  MOV     SP,R0
                023000  104416                  TRAP    C$PNTS
                023002  062706  000006          ADD     #6,SP
254             023006  000431                  BR      4$
255             023010  020227  160001  3$:     CMP     R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
256             023014  001012                  BNE     30$             ; BR IF NO.
257             023016                          PRINTS  #DEVNRD,R3
                023016  010346                  MOV     R3,-(SP)
                023020  012746  023305          MOV     #DEVNRD,-(SP)
                023024  012746  000002          MOV     #2,-(SP)
                023030  010600                  MOV     SP,R0
                023032  104416                  TRAP    C$PNTS
                023034  062706  000006          ADD     #6,SP
258             023040  000414                  BR      4$
259             023042  042702  170000  30$:    BIC     #^C7777,R2
260             023046                          PRINTS  #DEVDRO,R3,R2
                023046  010246                  MOV     R2,-(SP)
                023050  010346                  MOV     R3,-(SP)
                023052  012746  023366          MOV     #DEVDRO,-(SP)
                023056  012746  000003          MOV     #3,-(SP)
                023062  010600                  MOV     SP,R0
                023064  104416                  TRAP    C$PNTS
                023066  062706  000010          ADD     #10,SP
261             023072  062704  000002  4$:     ADD     #2,R4
262             023076  005203                  INC     R3
263             023100  020427  003372          CMP     R4,#ERTABE
264             023104  103701                  BLO     1$
265             023106  012604                  MOV     (SP)+,R4
266             023110  012603                  MOV     (SP)+,R3
267             023112  012602                  MOV     (SP)+,R2
268             023114                          ENDRPT          ; UNUSED.
                023114                  L10035:
                023114  104425                  TRAP    C$RPT
269
270
271             023116      045     116     045 DEVSUM: .ASCIZ  /%N%ADEVICE STATUS SUMMARY:%N/
272             023153      045     101     040 DEVONL: .ASCIZ  /%A  UNIT %D3%A    ONLINE,  ERRORS = %D%N/
273             023223      045     101     040 DEVNXR: .ASCIZ  /%A  UNIT %D3%A    DROPPED, NON-EXISTENT REGISTER%N/
274             023305      045     101     040 DEVNRD: .ASCIZ  /%A  UNIT %D3%A    DROPPED, NOT READY AT STARTUP%N/
275             023366      045     101     040 DEVDRO: .ASCIZ  /%A  UNIT %D3%A    DROPPED, ERRORS = %D%N/
276                                                 .EVEN
277
278             023436                          ENDMOD
279
280
```

```
                                .TITLE  TSV5A - HARDWARE TESTS
                                BGNMOD  TSV5
023436                  TSV5::
023436
```

```
                                          .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.
                                  ;

 023436                                   BGNTST
 023436                                                                     T1::
 023436  012700  023634                   MOV     #TST1ID,R0               ;ASCII MESSAGE TO IDENTIFY TEST
 023442  004737  016500                   JSR     PC,TSTSETUP              ;DO INITIAL TEST SETUP
 023446  012737  000024  002214           MOV     #20.,LOOPCNT             ;PERFORM 20 ITERATIONS
 023454                          T1LOOP:
 023454  005003                           CLR     R3                       ;USE R3 AS FATAL ERROR FLAG

 023456                                   BGNSUB                           ;//////////// BEGIN SUBTEST ////////////
 023456                                                                          T1.1:
 023456  104402                                                                          TRAP    C$BSUB

 023460                                   BRESET                           ;ISSUE A BUS RESET
 023460  104433                                                                          TRAP    C$RESET
 023462  004737  016240                   JSR     PC,WAITF                 ;WAIT FOR READY
 023466  016501  000002                   MOV     TSSR(R5),R1              ;GET THE CONTENTS OF TSSR
 023472  010102                           MOV     R1,R2                    ;CONTENTS OF TSSR
 023474  042702  176277                   BIC     #^C<HIADDR!OFL>,R2       ;THESE BITS MAY BE SET
```

```
     83 023500  052702  002200                BIS     #SSR!NBA,R2             ;READY AND NEW DATA SHOULD BE SET
     84 023504  020102                        CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
     85 023506  001405                        BEQ     10$                     ;BRANCH IF COMPARE
     89 023510                                ERRDF   ERRNO,SFHERR,SFFMSG     ;REPORT A FATAL ERROR
        023510  104455                                                              TRAP    C$ERDF
        023512  000145                                                              .WORD   101
        023514  003701                                                              .WORD   SFHERR
        023516  012072                                                              .WORD   SFFMSG
     90 023520  005203                        INC     R3                      ;SET THE FATAL ERROR FLAG
     91 023522                          10$:
     92 023522                                ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        023522                                                                      L10037:
        023522  104403                                                              TRAP    C$ESUB
     93
```

```
                                        TST     R3              ;DID WE HAVE FATAL ERROR ?
                                        BEQ     20$             ;BRANCH IF NOT
                                        JSR     PC,CKDROP       ;GO DROP THIS UNIT, IF ALLOWED
                                20$:    CLR     R3              ;RESET FATAL ERROR FLAG

                                        BGNSUB                  ;////////////// BEGIN SUBTEST //////////////
                                                                        T1.2:
                                                                                TRAP    C$BSUB

                                        CLR     TSSR(R5)        ;WRITE TO ISSUE A SOFT RESET
                                        JSR     PC,WAITF        ;WAIT FOR READY TO SET
                                        MOV     TSSR(R5),R1     ;GET REGISTER TSSR DATA
                                        MOV     R1,R2           ;CONTENTS OF TSSR
                                        BIC     #^C<HIADDR!OFL>,R2 ;THESE BITS MAY BE SET
                                        BIS     #SSR!NBA,R2     ;READY AND NEW DATA SHOULD BE SET
                                        CMP     R1,R2           ;COMPARE EXPECTED TO RECEIVED
                                        BEQ     10$             ;BRANCH IF COMPARE
                                        ERRDF   ERRNO,SFIERR,SFFMSG ;REPORT A FATAL ERROR
                                                                                TRAP    C$ERDF
                                                                                .WORD   102
                                                                                .WORD   SFIERR
                                                                                .WORD   SFFMSG
                                        INC     R3              ;SET THE ERROR FLAG
                                10$:
                                        ENDSUB                  ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
                                                                        L10040:
                                                                                TRAP    C$ESUB


                                        TST     R3              ;FATAL ERROR DETECTED ?
                                        BEQ     20$             ;BRANCH IF NOT
                                        JSR     PC,CKDROP       ;SEE IF TIME TO DROP UNIT
                                20$:    JSR     PC,TSTLOOP      ;SHOULD WE DO ITERATIONS ?
                                        BCC     40$             ;BRANCH IF NOT
                                        JMP     T1LOOP          ;LOOP UNTIL COUNT EXPIRED
                                40$:    EXIT    TST             ;ALL DONE THIS TEST
                                                                                TRAP    C$EXIT
                                                                                .WORD   L10036-.

                                ;+
                                ;LOCAL TEXT MESSAGES FOR TEST
                                ;-

                        111     156     151 TST1ID: .ASCIZ  'Initialization'
                                            .EVEN
                                            ENDTST
                                                                        L10036:
                                                                                TRAP    C$ETST
```

```
                        .SBTTL  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).
                        ;
                                BGNTST
                                                                                T2::
                        T2LOOP: MOV     #TST2ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
                                JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
                                MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
                                CLR     R4                      ;STARTING DATA PATTERN
                                MOV     #-1,R3                  ;DO INIT ON FIRST TIME THROUGH
                        5$:     TST     R3                      ;DO WE NEED SOFT INIT
                                BEQ     10$                     ;BRANCH IF NOT
                                CLR     R3                      ;DON'T NEED INIT NEXT TIME THRU
                                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$ERDF
                                                                        .WORD   201
                                                                        .WORD   SFIERR
                                                                        .WORD   SFIMSG
                                INC     R3                      ;FORCE SOFT INIT ON NEXT PASS
                        10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG

                                BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
```

```
                023736  104404                                                          TRAP    C$BSEG
197
198 023740  110465  000001          MOVB    R4,TSDBH(R5)        ;SET MAINT MODE + WRITE DATA
199 023744  004737  016240          JSR     PC,WAITF            ;WAIT FOR SSR TO SET
200 023750  103411                  BCS     15$                 ;BR IF CARRY SET (GOOD RETURN)
201 023752  010001                  MOV     R0,R1               ;SAVE CONTENTS OF TSSR
202 023754  010402                  MOV     R4,R2               ;DATA THAT WAS WRITTEN
206 023756                          ERRDF   ERRNO,T2SSR,EXPREC  ;DEVICE FATAL SSR FAILED TO SET
    023756  104455                                                                  TRAP    C$ERDF
    023760  000312                                                                 .WORD   202
    023762  024252                                                                 .WORD   T2SSR
    023764  015464                                                                 .WORD   EXPREC
207 023766  005203                  INC     R3                  ;FORCE SOFT INIT ON NEXT PASS
208 023770  005237  002220          INC     FATFLG              ;SET FATAL ERROR FLAG
209 023774                  15$:    CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
    023774  104406                                                                  TRAP    C$CLP1
210 023776  005737  002220          TST     FATFLG              ;WAS FATAL ERROR RECEIVED ?
211 024002  001402                  BEQ     20$                 ;BRANCH IF NOT
212 024004  004737  017172          JSR     PC,CKDROP           ;SEE IF TIME TO DROP UNIT
213 024010  010402          20$:    MOV     R4,R2               ;DATA PATTERN WRITTEN
214 024012  042702  177774          BIC     #^C<BIT0!BIT1>,R2   ;CLEAR ALL BUT LOW 2 BITS
215 024016  000302                  SWAB    R2                  ;BITS 8 AND 9 HAVE LOW DATA BITS
216 024020  052702  002200          BIS     #SSR!NBA,R2         ;THESE BITS MUST BE SET ALSO
217 024024  016501  000002          MOV     TSSR(R5),R1         ;GET THE CONTENTS OF TSSR
218 024030  032701  000100          BIT     #OFL,R1             ;IS OFF-LINE BIT SET ?
219 024034  001402                  BEQ     25$                 ;BRANCH IF NOT OFF-LINE
220 024036  052702  000100          BIS     #OFL,R2             ;SET OFF-LINE IN EXPECTED DATA
221 024042  020201          25$:    CMP     R2,R1               ;DOES EXPECTED MATCH RECEIVED ?
222 024044  001405                  BEQ     30$                 ;OKAY IF MATCH
226 024046                          ERRHRD  ERRNO,T2TSSR,EXPREC ;TSSR WASN'T CORRECT
    024046  104456                                                                 TRAP    C$ERHRD
    024050  000313                                                                 .WORD   203
    024052  024205                                                                 .WORD   T2TSSR
    024054  015464                                                                 .WORD   EXPREC
227 024056  005203                  INC     R3                  ;FORCE SOFT INIT ON NEXT PASS
228 024060                  30$:    CKLOOP                      ;LOOP ON ERROR ?
    024060  104406                                                                 TRAP    C$CLP1
229 024062  016501  000000          MOV     TSBA(R5),R1         ;GET TSBA REGISTER CONTENTS
230 024066  005002                  CLR     R2                  ;
231 024070  150402                  BISB    R4,R2               ;DATA PATTERN WRITTEN
232 024072  000302                  SWAB    R2                  ;MOVE INTO TOP BYTE
233 024074  150402                  BISB    R4,R2               ;BOTH HALVES SHOULD BE SAME
234 024076  020102                  CMP     R1,R2               ;COMPARE EXPECTED TO RECEIVED
235 024100  001405                  BEQ     35$                 ;ERROR IF NOT EQUAL
239 024102                          ERRHRD  ERRNO,T2TSBA,EXPREC ;PRINT THE ERROR & EXPD/RECV
    024102  104456                                                                 TRAP    C$ERHRD
    024104  000314                                                                 .WORD   204
    024106  024140                                                                 .WORD   T2TSBA
    024110  015464                                                                 .WORD   EXPREC
240 024112  005203                  INC     R3                  ;FORCE SOFT INIT ON NEXT PASS
241
242 024114                  35$:    ENDSEG                      ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
    024114                                                                      10000$:
    024114  104405                                                                 TRAP    C$ESEG
243
244 024116  105204                  INCB    R4                  ;NEXT DATA PATTERN TO TEST
245 024120  001270                  BNE     5$                  ;BRANCH TILL BACK TO ZERO
```

```
                                                JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
                                                BCC     40$                     ;BRANCH IF NOT
                                                JMP     T2LOOP                  ;LOOP UNTIL COUNT EXPIRED
                                        40$:    EXIT    TST                     ;ALL DONE THIS TEST
024134  104432                                                                          TRAP    C$EXIT
024136  000214                                                                          .WORD   L10041-.

                                        ;+
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-

024140      124     123     102     T2TSBA: .ASCIZ  'TSBA Incorrect After TSDB High Write'
024205      124     123     123     T2TSSR: .ASCIZ  'TSSR Incorrect After TSDB High Write'
024252      116     157     040     T2SSR:  .ASCIZ  'No Sub-System Ready After TSDB High Write'
024324      127     162     141     TST2ID: .ASCIZ  'Wrap Data - High Byte'
                                                .EVEN
024352                                          ENDTST
024352                                                                  L10041:
024352  104401                                                                  TRAP    C$ETST
```

```
                                .SBTTL  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).
                        ;
024354                          BGNTST
024354                                                                          T3::
024354  012700  025017            MOV     #TST3ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
024360  004737  016500            JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
024364  012737  000024  002214    MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
024372  005004            T3LOOP: CLR     R4                      ;STARTING DATA PATTERN
024374  012703  000001            MOV     #1,R3                   ;FORCE SOFT INIT ON FIRST PASS
024400  005703            5$:     TST     R3                      ;SET IF INIT IS REQUIRED
024402  001412                    BEQ     10$                     ;BRANCH IF NO INIT NEEDED
024404  005003                    CLR     R3                      ;ASSUME NO INIT NEEDED ON NEXT LOOP
024406  004737  015764            JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
024412  103406                    BCS     10$                     ;BR IF SOFT INIT = OK
024414  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
024416                            ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
024416  104455                                                                    TRAP    C$ERDF
024420  000455                                                                    .WORD   301
024422  003646                                                                    .WORD   SFIERR
024424  012024                                                                    .WORD   SFIMSG
024426  005203                    INC     R3                      ;FORCE INIT ON NEXT PASS
024430  005037  002220    10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG

024434                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
024434  104404                                                                  TRAP    C$BSEG

024436  110465  000000            MOVB    R4,TSDB(R5)             ;SET MAINT MODE + WRITE DATA
024442  004737  016240            JSR     PC,WAITF                ;WAIT FOR SSR TO SET
024446  103411                    BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
024450  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
024452  010402                    MOV     R4,R2                   ;DATA THAT WAS WRITTEN
024454                            ERRDF   ERRNO,T3SSR,EXPREC      ;DEVICE FATAL SSR FAILED TO SET
```

```
                024454  104455                                                          TRAP    C$ERDF
                024456  000456                                                          .WORD   302
                024460  024746                                                          .WORD   T3SSR
                024462  015464                                                          .WORD   EXPREC
     325        024464  005203                  INC     R3              ;FORCE INIT ON NEXT PASS
     326        024466  005237  002220          INC     FATFLG          ;SET THE FATAL ERROR FLAG
     327        024472                  15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
                024472  104406                                                          TRAP    C$CLP1
     328        024474  005737  002220          TST     FATFLG          ;DID WE GET FATAL ERROR ?
     329        024500  001402                  BEQ     20$             ;BRANCH IF NOT
     330        024502  004737  017172          JSR     PC,CKDROP       ;DROP UNIT, IF DROP ALLOWED
     331        024506  010402          20$:    MOV     R4,R2           ;DATA PATTERN WRITTEN
     332        024510  042702  177774          BIC     #^C<BIT0!BIT1>,R2 ;CLEAR ALL BUT LOW 2 BITS
     333        024514  000302                  SWAB    R2              ;BITS 8 AND 9 HAVE LOW DATA BITS
     334        024516  052702  002200          BIS     #SSR!NBA,R2     ;THESE BITS MUST BE SET ALSO
     335        024522  016501  000002          MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
     336        024526  032701  000100          BIT     #OFL,R1         ;IS DRIVE OFF-LINE
     337        024532  001402                  BEQ     25$             ;BRANCH IF NOT OFFLINE
     338        024534  052702  000100          BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED
     339        024540  020201          25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
     340        024542  001405                  BEQ     30$             ;OKAY IF MATCH
     344        024544                  ERRHRD  ERRNO,T3TSSR,EXPREC     ;TSSR WASN'T CORRECT
                024544  104456                                                          TRAP    C$ERHRD
                024546  000457                                                          .WORD   303
                024550  024702                                                          .WORD   T3TSSR
                024552  015464                                                          .WORD   EXPREC
     345        024554  005203                  INC     R3              ;FORCE INIT ON NEXT PASS
     346        024556          30$:    CKLOOP                          ;LOOP ON ERROR ?
                024556  104406                                                          TRAP    C$CLP1
     347        024560  016501  000000          MOV     TSBA(R5),R1     ;GET TSBA REGISTER CONTENTS
     348        024564  005002                  CLR     R2              ;
     349        024566  150402                  BISB    R4,R2           ;DATA PATTERN WRITTEN
     350        024570  000302                  SWAB    R2              ;MOVE INTO TOP BYTE
     351        024572  150402                  BISB    R4,R2           ;BOTH HALVES SHOULD BE SAME
     352        024574  020102                  CMP     R1,R2           ;COMPARE EXPECTED TO RECEIVED
     353        024576  001405                  BEQ     35$             ;BR IF ERROR NOT EQUAL
     357        024600                  ERRHRD  ERRNO,T3TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
                024600  104456                                                          TRAP    C$ERHRD
                024602  000460                                                          .WORD   304
                024604  024636                                                          .WORD   T3TSBA
                024606  015464                                                          .WORD   EXPREC
     358        024610  005203                  INC     R3              ;FORCE INIT ON NEXT PASS
     359
     360        024612          35$:    ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                024612                                                                  10000$:
                024612  104405                                                          TRAP    C$ESEG
     361
     362        024614  105204                  INCB    R4              ;NEXT DATA PATTERN TO TEST
     363        024616  001270                  BNE     5$              ;ALL DONE WHEN BACK TO ZERO
     364        024620  004737  016446          JSR     PC,TSTLOOP      ;SHOULD WE DO ITERATIONS ?
     365        024624  103002                  BCC     40$             ;BRANCH IF NOT
     366        024626  000137  024372          JMP     T3LOOP          ;LOOP UNTIL COUNT EXPIRED
     367        024632          40$:    EXIT    TST                     ;ALL DONE THIS TEST
                024632  104432                                                          TRAP    C$EXIT
                024634  000210                                                          .WORD   L10042-.
     368
     369                          ;+
```

```
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-
                                        
024636      124     123     102  T3TSBA: .ASCIZ  'TSBA Incorrect After TSDB Low Write'
024702      124     123     123  T3TSSR: .ASCIZ  'TSSR Incorrect After TSDB Low Write'
024746      116     157     040  T3SSR:  .ASCIZ  'No Sub-System Ready After TSDB Low Write'
025017      127     162     141  TST3ID: .ASCIZ  'Wrap Data - Low Byte'
                                        .EVEN
                                        ENDTST
025044
025044                                                                  L10042:
025044  104401                                                                  TRAP    C$ETST
```

```
                                       .SBTTL  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).  THESE
                                 ;      TESTS ARE PERFORMED BY THREE SUBTESTS, DESCRIBED BELOW.  A
                                 ;      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 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.
                                 ;
025046                                 BGNTST
025046                                                                  T4::

025046                                 BGNSUB                           ;//////////// BEGIN SUBTEST ////////////
025046                                                                  T4.1:
025046  104402                                                                  TRAP    C$BSUB

025050  012700  026356                 MOV     #TST4ID,R0               ;ASCII MESSAGE TO IDENTIFY TEST
025054  004737  016500                 JSR     PC,TSTSETUP              ;DO INITIAL TEST SETUP
025060  012737  000005  002214         MOV     #5,LOOPCNT               ;PERFORM 5 ITERATIONS
```

```
                                T4LOOP:
                                        JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
                                        BCS     20$                     ;BR IF INIT WAS OK
                                        MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
                                        ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
                                                                                TRAP    C$ERDF
                                                                                .WORD   401
                                                                                .WORD   SFIERR
                                                                                .WORD   SFIMSG
                                20$:    CLR     R4                      ;SET RAM ADDRESS AT ZERO
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        CLRB    TSDB(R5)                ;SET INTO MAINTENANCE MODE
                                25$:
                                        BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
                                                                                TRAP    C$BSEG

                                        MOVB    R4,R2                   ;EXPECTED DATA FROM WRAP-AROUND
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        MOV     R4,TSDB(R5)             ;LOAD ADDRESS INTO TSDB
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        MOVB    R2,TSDB(R5)             ;LOADS DATA INTO RAM LOCATION
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        MOV     TSBA(R5),R1             ;READS WRAP DATA
                                        CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ
                                        BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
                                        ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
                                                                                TRAP    C$ERHRD
                                                                                .WORD   402
                                                                                .WORD   TSBAM2
                                                                                .WORD   EXPREC
                                30$:
                                        ENDSEG                          ;<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<
                                                                                10000$:
                                                                                TRAP    C$ESEG

                                        INC     R4                      ;NEXT ADDRESS
                                        CMP     R4,#10000               ;END OF RAM MEMORY CHECK
                                        BNE     25$                     ;LOOP TILL ALL RAM WRITTEN
                                        ESCAPE  SUB                     ;DON'T CONTINUE IF ERROR ON WRITE
                                                                                TRAP    C$ESCAPE
                                                                                .WORD   L10044-.

                                        CLR     R2                      ;CLEAR OUT R2 HIGH BITS
                                        DEC     R4                      ;SET BACK TO 7777
                                40$:    MOVB    R4,R2                   ;GET DATA PATTERN BACK IN SHAPE
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        MOV     R4,TSDB(R5)             ;LOAD UP RAM ADDRESS POINTER
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        MOV     TSBA(R5),R1             ;READ RAM CONTENTS BACK
                                        CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
                                        BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
                                        ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
                                                                                TRAP    C$ERHRD
                                                                                .WORD   403
                                                                                .WORD   TSBAM2
                                                                                .WORD   EXPREC
                                45$:    CKLOOP                          ;SCOPE LOOP
```

```
                025250  104406                                                          TRAP    C$CLP1
    489         025252  116501  000001          MOVB    TSBAH(R5),R1        ;HIGH BYTE READ OF TSBA
    490         025256  010402                  MOV     R4,R2               ;DATA PATTERN WRITTEN
    491         025260  000302                  SWAB    R2                  ;HIGH TO LOW
    492         025262  060402                  ADD     R4,R2               ;TOTAL OF BYTES IN LOW BYTE
    493         025264  120102                  CMPB    R1,R2               ;SUM OF BYTES WRITTEN TO TSDB = TSBAH
    494         025266  001404                  BEQ     50$                 ;BR IF OK, THEY SHOULD BE
    498         025270                          ERRHRD  ERRNO,M2901.EXPREC  ;2901 PROBLEM ADDER
                025270  104456                                                          TRAP    C$ERHRD
                025272  000624                                                          .WORD   404
                025274  026124                                                          .WORD   M2901
                025276  015434                                                          .WORD   EXPREC
    499         025300                  50$:    CKLOOP                      ;SCOPE LOOP
                025300  104406                                                          TRAP    C$CLP1
    500         025302  005304                  DEC     R4                  ;DROP DATA COUNTER (PATTERN)
    501         025304  002342                  BGE     40$                 ;NOT AT LOC. ZERO YET
    502
    503         025306                          ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                025306                                                                  L10044:
                025306  104403                                                          TRAP    C$ESUB
    504
```

```
                                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
                                                                                                        T4.2:
                                                                                                TRAP    C$BSUB
                                        ;       TEST 4, SUBTEST 2
                                        ;
                                        ;       THIS SUBTEST WRITES RAM WITH ALL ZEROS
                                        ;       THEN WALKS AN ALL ONES WORD DOWN THROUGH MEMORY
                                        ;
025312  004737  015764                  JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
025316  103405                          BCS     20$                     ;BR IF INIT WAS OK
025320  010001                          MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
025322                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
025322  104455                                                                  TRAP    C$ERDF
025324  000625                                                                  .WORD   405
025326  003646                                                                  .WORD   SFIERR
025330  012024                                                                  .WORD   SFIMSG
025332  005002                  20$:    CLR     R2                      ;TEST DATA = 0
025334  005004                          CLR     R4                      ;STARTING RAM ADDRESS = 0
025336  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025342  105065  000000                  CLRB    TSDB(R5)                ;SET INTO MAINTENANCE MODE
025346                          25$:
025346                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
025346  104404                                                                  TRAP    C$BSEG

025350  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025354  010465  000000                  MOV     R4,TSDB(R5)             ;LOAD ADDRESS INTO TSDB
025360  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025364  110265  000000                  MOVB    R2,TSDB(R5)             ;LOADS DATA INTO RAM LOCATION
025370  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025374  016501  000000                  MOV     TSBA(R5),R1             ;READS WRAP DATA
025400  120102                          CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ ?
025402  001404                          BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
025404                                  ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
025404  104456                                                                  TRAP    C$ERHRD
025406  000626                                                                  .WORD   406
025410  026214                                                                  .WORD   TSBAM2
025412  015404                                                                  .WORD   EXPREC
025414                          30$:
025414                                  ENDSEG                          ;<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<
025414                                                                                  10000$:
025414  104405                                                                  TRAP    C$ESEG

025416  005204                          INC     R4                      ;NEXT ADDRESS
025420  020427  010000                  CMP     R4,#10000               ;END OF RAM MEMORY CHECK
025424  001350                          BNE     25$                     ;BR, MORE RAM TO GO

025426  005304                  35$:    DEC     R4                      ;SET BACK TO 7777
025430  005002                          CLR     R2                      ;SET TO ALL ZEROS
025432  004737  016326          40$:    JSR     PC,CHKTSSR              ;WAIT FOR READY (SSR) TO SET
025436  010465  000000                  MOV     R4,TSDB(R5)             ;LOAD UP THE ADDRESS FOR RAM
025442  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR READY (SSR) TO SET
025446  016501  000000                  MOV     TSBA(R5),R1             ;READ THE RAM CONTENTS BACK
025452  005002                          CLR     R2                      ;LOOKING FOR 000000 (EXPECTED)
025454  120102                          CMPB    R1,R2                   ;BOTH SHOULD BE 00000000 BINARY
025456  001404                          BEQ     43$                     ;BR, IF DATA IS GOOD
```

```
559 025460                          ERRHRD  ERRNO,TSBAM3,EXPREC     ;CHARACTERISTICS DATA NOT CORRECT
    025460  104456                                                              TRAP    C$ERHRD
    025462  000627                                                              .WORD   407
    025464  026276                                                              .WORD   TSBAM3
    025466  015464                                                              .WORD   EXPREC
560 025470  012702  000377      43$:    MOV     #000377,R2              ;SET ALL ONES WORD
561 025474  010465  000000              MOV     R4,TSDB(R5)             ;LOAD UP RAM ADDRESS POINTER
562 025500  004737  016326              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
563 025504  110265  000000              MOVB    R2,TSDB(R5)             ;WRITE DATA INTO RAM
564 025510  004737  016326              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
565 025514  016501  000000              MOV     TSBA(R5),R1             ;READ RAM CONTENTS BACK
566 025520  120102                      CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
567 025522  001404                      BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
571 025524                          ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
    025524  104456                                                              TRAP    C$ERHRD
    025526  000630                                                              .WORD   408
    025530  026214                                                              .WORD   TSBAM2
    025532  015464                                                              .WORD   EXPREC
572 025534                  45$:    CKLOOP                          ;SCOPE LOOP
    025534  104406                                                              TRAP    C$CLP1
573 025536  004737  016326              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
574 025542  010465  000000              MOV     R4,TSDB(R5)             ;WORD WRITE TO SET UP ADDRESS
575 025546  004737  016326              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
576 025552  116501  000001              MOVB    TSBAH(R5),R1            ;HIGH BYTE READ OF TSBA
577 025556  010403                      MOV     R4,R3                   ;DATA PATTERN WRITTEN
578 025560  000303                      SWAB    R3                      ;HIGH TO LOW
579 025562  060403                      ADD     R4,R3                   ;TOTAL OF BYTES IN LOW BYTE
580 025564  120103                      CMPB    R1,R3                   ;SUM OF BYTES WRITTEN TO TSDB = TSBAH
581 025566  001404                      BEQ     50$                     ;BR IF OK, THEY SHOULD BE
585 025570                          ERRHRD  ERRNO,M2901,EXPREC      ;2901 PROBLEM ADDER
    025570  104456                                                              TRAP    C$ERHRD
    025572  000631                                                              .WORD   409
    025574  026124                                                              .WORD   M2901
    025576  015464                                                              .WORD   EXPREC
586 025600                  50$:    CKLOOP                          ;SCOPE LOOP
    025600  104406                                                              TRAP    C$CLP1
587 025602  005304                      DEC     R4                      ;DROP RAM ADDRESS POINTER
588 025604  002312                      BGE     40$                     ;NOT AT LOC. ZERO YET
589
590 025606                          ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
    025606                                                                  L10045:
    025606  104403                                                              TRAP    C$ESUB
591
```

```
                                        BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
                                                                                              T4.3:
025610  104402                                                                                TRAP    C$BSUB

                        ;       TEST 4, SUBTEST 3
                        ;
                        ;       THIS SUBTEST WRITES RAM WITH ALL ONES
                        ;       THEN WALKS A ZERO WORD DOWN THROUGH MEMORY
                        ;
025612  004737  015764          JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
025616  103405                  BCS     20$                     ;BR IF INIT WAS OK
025620  010001                  MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
025622                          ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
025622  104455                                                                        TRAP    C$ERDF
025624  000632                                                                        .WORD   410
025626  003646                                                                        .WORD   SFIERR
025630  012024                                                                        .WORD   SFIMSG
025632  012702  177777  20$:     MOV     #177777,R2              ;SET DATA AT ALL ONES
025636  005004                  CLR     R4                      ;SET RAM ADDRESS AT ZERO
025640  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025644  105065  000000          CLRB    TSDB(R5)                ;SET INTO MAINTENANCE MODE
025650                  25$:
025650                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
025650  104404                                                                        TRAP    C$BSEG

025652  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025656  010465  000000          MOV     R4,TSDB(R5)             ;LOAD ADDRESS INTO TSDB
025662  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025666  110265  000000          MOVB    R2,TSDB(R5)             ;LOADS DATA INTO RAM LOCATION
025672  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025676  016501  000000          MOV     TSBA(R5),R1             ;READS WRAP DATA
025702  120102                  CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ ?
025704  001404                  BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
025706                          ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
025706  104456                                                                        TRAP    C$ERHRD
025710  000633                                                                        .WORD   411
025712  026214                                                                        .WORD   TSBAM2
025714  015464                                                                        .WORD   EXPREC
025716          30$:
025716                          ENDSEG                          ;<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<
                                                                                        10000$:
025716  104405                                                                        TRAP    C$ESEG

025720  005204                  INC     R4                      ;NEXT ADDRESS
025722  020427  010000          CMP     R4,#10000               ;END OF RAM MEMORY CHECK
025726  001350                  BNE     25$                     ;BR, MORE RAM TO GO

025730                          ESCAPE  SUB                     ;NO CHECK IF WRITTEN INCORRECTLY
025730  104410                                                                        TRAP    C$ESCAPE
025732  000152                                                                        .WORD   L10046-.
025734  005304          35$:    DEC     R4                      ;SET BACK TO 7777
025736  004737  016326  40$:    JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025742  012702  000377          MOV     #000377,R2              ;SET UP EXPECTED DATA REGISTER
025746  005001                  CLR     R1                      ;CLEAN OUT REGISTER
025750  010465  000000          MOV     R4,TSDB(R5)             ;SELECT ADDRESS IN RAM
025754  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY (SSR)
```

```
641 025760  016501  000000          MOV     TSBA(R5),R1             ;PICK UP RAM CONTENTS
642 025764  120102                  CMPB    R1,R2                   ;IS MEMORY STILL ALL ONES
643 025766  001404                  BEQ     43$                     ;BR, IF OK (ALL ONES)
647 025770                          ERRHRD  ERRNO,TSBAM3,EXPREC     ;MEMORY CHANGED AFTER ALL ONES WRITE
    025770  104456                                                          TRAP    C$ERHRD
    025772  000634                                                          .WORD   412
    025774  026276                                                          .WORD   TSBAM3
    025776  015464                                                          .WORD   EXPREC
648 026000  005002          43$:    CLR     R2                      ;SET UP NEW EXPECTED
649 026002  010465  000000          MOV     R4,TSDB(R5)             ;LOAD UP RAM ADDRESS POINTER
650 026006  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
651 026012  110265  000000          MOVB    R2,TSDB(R5)             ;WRITE DATA INTO RAM
652 026016  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
653 026022  016501  000000          MOV     TSBA(R5),R1             ;READ RAM CONTENTS BACK
654 026026  120102                  CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
655 026030  001404                  BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
659 026032                          ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
    026032  104456                                                          TRAP    C$ERHRD
    026034  000635                                                          .WORD   413
    026036  026214                                                          .WORD   TSBAM2
    026040  015464                                                          .WORD   EXPREC
660 026042          45$:    CKLOOP                                  ;SCOPE LOOP
    026042  104406                                                          TRAP    C$CLP1
661 026044  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
662 026050  116501  000001          MOVB    TSBAH(R5),R1            ;HIGH BYTE READ OF TSBA
663 026054  010203                  MOV     R2,R3                   ;DATA PATTERN WRITTEN
664 026056  000303                  SWAB    R3                      ;HIGH TO LOW
665 026060  060203                  ADD     R2,R3                   ;TOTAL OF BYTES IN LOW BYTE
666 026062  120103                  CMPB    R1,R3                   ;SUM OF BYTES WRITTEN TO TSDB = TSBAH
667 026064  001404                  BEQ     50$                     ;BR IF OK, THEY SHOULD BE
671 026066                          ERRHRD  ERRNO,M2901,EXPREC      ;2901 PROBLEM ADDER
    026066  104456                                                          TRAP    C$ERHRD
    026070  000636                                                          .WORD   414
    026072  026124                                                          .WORD   M2901
    026074  015464                                                          .WORD   EXPREC
672 026076          50$:    CKLOOP                                  ;SCOPE LOOP
    026076  104406                                                          TRAP    C$CLP1
673 026100  005304                  DEC     R4                      ;DROP RAM ADDRESS POINTER
674 026102  001315                  BNE     40$                     ;NOT AT LOC. ZERO YET
675
676 026104                          ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
    026104                                                                  L10046:
    026104  104403                                                          TRAP    C$ESUB
677
678 026106  004737  016446          JSR     PC,TSTLOOP              ;DO WE NEED TO ITERATE TEST ?
679 026112  103002                  BCC     63$                     ;BRANCH IF NOT
680 026114  000137  025066          JMP     T4LOOP                  ;EXECUTE AGAIN
681 026120          63$:    EXIT    TST                             ;ALL DONE THIS TEST
    026120  104432                                                          TRAP    C$EXIT
    026122  000256                                                          .WORD   L10043-.
682                         ;+
683                         ;LOCAL TEXT MESSAGES FOR TEST
684                         ;-
685
686 026124  040  124  123  M2901:  .ASCIZ  ' TSBA High Byte Not Sum of Last TSDB Write (2901 Error)'
687 026214  040  127  162  TSBAM2: .ASCIZ  ' Write to TSDB Not Equal to Read of TSBA Low Byte'
688 026276  127  162  151  TSBAM3: .ASCIZ  'Write To RAM Location Modified Another Location'
```

```
    689 026356     122     101     115  TST4ID: .ASCIZ  'RAM Verification'
    690                                 .EVEN
    691 026400                          ENDTST
        026400                                                    L10043:
        026400  104401                                                    TRAP    C$ETST
    692
```

```
                                                 .SBTTL  TEST  5: SECOND INITIALIZATION TEST

                                        ;        THIS TEST VERIFIES THE SAME ELEMENTS AS DID INITIALIZATION TEST
                                        ;        #1 AND ALSO CHECKS THAT CERTAIN PARTS OF RAM IS CLEARED TO ZERO
                                        ;        AND THAT 2901 REGISTERS 10 AND 11 ARE ALSO CLEARED TO ZERO.
                                        ;        THIS IS A CONFIDENCE CHECK OF A PART OF THE SELF-TEST SEQUENCE
                                        ;        (I.E., THAT IT IS REALLY BEING EXECUTED).  FOR EACH OF TWO
                                        ;        SUBTESTS (ONE FOR INITIALIZING VIA A BUS INIT, THE OTHER FOR
                                        ;        INITIALIZING BY WRITING INTO THE TSSR), THE FOLLOWING SEQUENCE
                                        ;        IS PERFORMED:
                                        ;
                                        ;
                                        ;                1.  EACH RAM LOCATION AND 2901 REGISTERS 10 AND 11 ARE SET
                                        ;                    TO ALL 1'S BY USING WRITES INTO THE TSDB REGISTER (LOW
                                        ;                    BYTE AND MAINTENANCE MODE WORD WRITES).
                                        ;
                                        ;                2.  THE CONTROLLER IS INITIALIZED AND THE VARIOUS CHECKS ON
                                        ;                    THE TSSR DESCRIBED IN INITIALIZATION TEST #1 ARE
                                        ;                    PERFORMED.
                                        ;
                                        ;                3.  #1'S (377 OCTAL) ARE WRITTEN INTO THE LOW BYTE OF TSDB,
                                        ;                    WHICH SHOULD CAUSE RAM LOCATION 0 TO BE WRITTEN TO ALL
                                        ;                    1'S SINCE 2901 REGISTERS 10 AND 11, SPECIFYING THE RAM
                                        ;                    ADDRESS, SHOULD BE 0.  RAM LOCATION 0 IS VERIFIED BY
                                        ;                    WRITING A WORD OF ZEROS INTO THE TSDB.  THE RESULTING
                                        ;                    LOW BYTE OF TSBA SHOULD CONTAIN ALL 1'S.
                                        ;
                                        ;                4.  THE ENTIRE RAM IS SCANNED.  LOCATION 0 SHOULD CONTAIN
                                        ;                    ALL 1'S AND THE REMAINING LOCATIONS, EXCEPT FOR THE
                                        ;                    MESSAGE BUFFER IMAGE AREA, SHOULD CONTAIN 0.
                                        ;                    DISCREPANCIES ARE REPORTED.  AN ERROR AT THIS POINT IS
                                        ;                    MOST LIKELY DUE TO A ROM, PIPELINE OR SEQUENCER PROBLEM
                                        ;                    OR A TIMING PROBLEM.
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
 026402                                          BGNTST
 026402                                                                          T5::
 026402  012700  027354                          MOV     #TST5ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
 026406  004737  016500                          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
 026412  012737  000024  002214                  MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
 026420                                  T5LOOP:
 026420  005037  002220                          CLR     FATFLG                  ;CLEAR THE FATAL ERROR FLAG

 026424                                          BGNSUB                          ;//////////// BEGIN SUBTEST ///////////
 026424                                                                                  T5.1:
 026424  104402                                                                                  TRAP    C$BSUB

 026426  004737  015764                          JSR     PC,SOFINIT              ;DO A SOFT TO START
 026432  103404                                  BCS     10$                     ;BRANCH IF O.K.
 026434                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;REPORT ERROR AND DROP DRIVE
 026434  104455                                                                                  TRAP    C$ERDF
 026436  000765                                                                                  .WORD   501
 026440  003646                                                                                  .WORD   SFIERR
 026442  012024                                                                                  .WORD   SFIMSG
 026444  012702  177777                  10$:    MOV     #-1,R2                  ;ALL ONE DATA PATTERN
```

```
                                        CLR     R4                      ;STARTING RAM ADDRESS
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                15$:    CLRB    TSDB(R5)                ;SET MAINTENANCE MODE
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        MOV     R4,TSDB(R5)             ;SET THE NEXT RAM ADDRESS
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        MOVB    R2,TSDB(R5)             ;LOAD TEST DATA
                                        INC     R4                      ;NEXT ADDRESS TO TEST
                                        CMP     R4,#7777                ;COMPARE TO LAST ADDRESS
                                        BLE     15$                     ;BRANCH TILL ALL DATA WRITTEN
                                        BRESET                          ;ISSUE A BUS RESET
                                                                                TRAP    C$RESET
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
                                        MOV     R1,R2                   ;CONTENTS OF TSSR
                                        BIC     #^C<HIADDR!OFL>,R2      ;THESE BITS MAY BE SET
                                        BIS     #SSR!NBA,R2             ;READY AND NEW DATA SHOULD BE SET
                                        CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
                                        BEQ     20$                     ;BRANCH IF COMPARE
                                        ERRDF   ERRNO,SFHERR,SFFMSG     ;REPORT A FATAL ERROR
                                                                                TRAP    C$ERDF
                                                                                .WORD   502
                                                                                .WORD   SFHERR
                                                                                .WORD   SFFMSG
                                        INC     FATFLG                  ;SET FATAL ERROR FLAG
                                20$:    CKLOOP                          ;LOOP ON ERROR IF FLAG SET
                                                                                TRAP    C$CLP1
                                        ESCAPE  SUB                     ;EXIT IF FATAL ERROR DETECTED
                                                                                TRAP    C$ESCAPE
                                                                                .WORD   L10050-.
                                        JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
                                        CLRB    TSDB(R5)                ;PUT BACK INTO MAINTENANCE MODE
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        CLR     TSDB(R5)                ;SET ADDRESS BACK TO 0000
                                        MOV     #377,R2                 ;
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        MOVB    R2,TSDB(R5)             ;SHOULD POINT TO RAM 0
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        CLR     TSDB(R5)                ;SELECT LOCATION 0
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        MOVB    TSBA(R5),R1             ;READ RAM LOCATION SPECIFIED
                                        CMPB    R1,R2                   ;LOCATION SHOULD BE 377 OCTAL
                                        BEQ     25$                     ;BR IF OK
                                        ERRDF   ERRNO,T5ADDR,EXPREC     ;WASN'T POINTING TO CORRECT LOC.
                                                                                TRAP    C$ERDF
                                                                                .WORD   502
                                                                                .WORD   T5ADDR
                                                                                .WORD   EXPREC
                                        INC     FATFLG                  ;SET THE FATAL ERROR FLAG
                                25$:    CKLOOP                          ;SCOPE LOOP
                                                                                TRAP    C$CLP1
                                        ESCAPE  SUB                     ;NO MORE CHECKS IF FATAL ERROR
                                                                                TRAP    C$ESCAPE
                                                                                .WORD   L10050-.
                                        MOV     #310,R4                 ;START WITH LOC 310
                                        CLR     R2                      ;MEMORY EXPECTED SHOULD BE 000000
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
```

```
796 026700  010465  000000  30$:    MOV     R4,TSDB(R5)             ;SELECT LOCATION SPECIFIED
797 026704  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
798 026710  116501  000000          MOVB    TSBA(R5),R1             ;READ LOC CONTENTS
799 026714  120102                  CMPB    R1,R2                   ;CHECK MEMORY FOR 000000
800 026716  001406                  BEQ     40$                     ;BRANCH IF DATA OKAY
801 026720                          ERRDF   ERRNO,T5MEM,SFFMSG      ;MEMORY NOT ZERO AFTER INIT.
    026720  104455                                                          TRAP    C$ERDF
    026722  000766                                                          .WORD   502
    026724  027404                                                          .WORD   T5MEM
    026726  012072                                                          .WORD   SFFMSG
802 026730  005237  002220          INC     FATFLG                  ;SET THE FATAL ERROR FLAG
803 026734          40$:    CKLOOP
    026734  104406                                                          TRAP    C$CLP1
804 026736                          ESCAPE  SUB                     ;EXIT ON FATAL ERROR
    026736  104410                                                          TRAP    C$ESCAPE
    026740  000012                                                          .WORD   L10050-.
805 026742  005204                  INC     R4                      ;LOOK AT NEXT RAM LOC.
806 026744  020427  000400          CMP     R4,#400                 ;AT TOP OF RAM ADDRESS SPACE
807 026750  001353                  BNE     30$                     ;BRANCH TILL ALL MEMORY TESTED
808
809 026752                          ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
    026752                                                                  L10050:
    026752  104403                                                          TRAP    C$ESUB
810
811 026754  005737  002220          TST     FATFLG                  ;IS FATAL ERROR FLAG SET ?
812 026760  001404                  BEQ     50$                     ;BRANCH IF NOT
813 026762  004737  017172          JSR     PC,CKDROP               ;NO LOOP, TRY TO DROP DEVICE
814 026766  005037  002220          CLR     FATFLG                  ;CLEAR THE FATAL ERROR FLAG
815 026772          50$:
```

```
817 026772                              BGNSUB                  ;////////////// BEGIN SUBTEST //////////////
    026772                                                                          T5.2:
    026772  104402                                                                     TRAP    C$BSUB
818
819 026774  004737  015764              JSR     PC,SOFINIT      ;DO A SOFT TO START
820 027000  103404                      BCS     10$             ;BRANCH IF O.K.
824 027002                              ERRDF   ERRNO,SFIERR,SFIMSG ;REPORT ERROR AND DROP DRIVE
    027002  104455                                                                  TRAP    C$ERDF
    027004  000767                                                                  .WORD   503
    027006  003646                                                                  .WORD   SFIERR
    027010  012024                                                                  .WORD   SFIMSG
825 027012  012702  177777      10$:    MOV     #-1,R2          ;ALL ONE DATA PATTERN
826 027016  005004                      CLR     R4              ;STARTING RAM ADDRESS
827 027020  004737  016326              JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
828 027024  105065  000000      15$:    CLRB    TSDB(R5)        ;SET MAINTENANCE MODE
829 027030  004737  016326              JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
830 027034  010465  000000              MOV     R4,TSDB(R5)     ;SET THE NEXT RAM ADDRESS
831 027040  004737  016326              JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
832 027044  110265  000000              MOVB    R2,TSDB(R5)     ;LOAD TEST DATA
833 027050  005204                      INC     R4              ;NEXT ADDRESS TO TEST
834 027052  020427  007777              CMP     R4,#7777        ;COMPARE TO LAST ADDRESS
835 027056  003762                      BLE     15$             ;BRANCH TILL ALL DATA WRITTEN
836 027060  005065  000002              CLR     TSSR(R5)        ;ISSUE A SOFT RESET
837 027064  004737  016326              JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
838 027070  016501  000002              MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
839 027074  010102                      MOV     R1,R2           ;CONTENTS OF TSSR
840 027076  042702  176277              BIC     #^C<HIADDR!OFL>,R2 ;THESE BITS MAY BE SET
841 027102  052702  002200              BIS     #SSR!NBA,R2     ;READY AND NEW DATA SHOULD BE SET
842 027106  020102                      CMP     R1,R2           ;COMPARE EXPECTED TO RECEIVED
843 027110  001406                      BEQ     20$             ;BRANCH IF COMPARE
847 027112                              ERRDF   ERRNO,SFHERR,SFFMSG ;REPORT A FATAL ERROR
    027112  104455                                                                  TRAP    C$ERDF
    027114  000770                                                                  .WORD   504
    027116  003701                                                                  .WORD   SFHERR
    027120  012072                                                                  .WORD   SFFMSG
848 027122  005237  002220      20$:    INC     FATFLG          ;SET FATAL ERROR FLAG
849 027126                              CKLOOP                  ;LOOP ON ERROR IF FLAG SET
    027126  104406                                                                  TRAP    C$CLP1
850 027130                              ESCAPE  SUB             ;EXIT IF FATAL ERROR DETECTED
    027130  104410                                                                  TRAP    C$ESCAPE
    027132  000170                                                                  .WORD   L10051-.
851 027134  004737  016326              JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
852 027140  105065  000000              CLRB    TSDB(R5)        ;PUT BACK INTO MAINTENANCE MODE
853 027144  004737  016326              JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
854 027150  005065  000000              CLR     TSDB(R5)        ;SET ADDRESS BACK TO 0000
855 027154  012702  000377              MOV     #377,R2         ;
856 027160  004737  016326              JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
857 027164  110265  000000              MOVB    R2,TSDB(R5)     ;SHOULD POINT TO RAM 0
858 027170  004737  016326              JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
859 027174  005065  000000              CLR     TSDB(R5)        ;SELECT LOCATION 0
860 027200  004737  016326              JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
861 027204  116501  000000              MOVB    TSBA(R5),R1     ;READ RAM LOCATION SPECIFIED
862 027210  120102                      CMPB    R1,R2           ;LOCATION SHOULD BE 377 OCTAL
863 027212  001406                      BEQ     25$             ;BR IF OK
864 027214                              ERRDF   ERRNO,T5ADDR,EXPREC ;WASN'T POINTING TO CORRECT LOC.
    027214  104455                                                                  TRAP    C$ERDF
    027216  000770                                                                  .WORD   504
```

```
                027220  027442                                                  .WORD   T5ADDR
                027222  015464                                                  .WORD   EXPREC
865             027224  005237  002220          INC     FATFLG          ;SET THE FATAL ERROR FLAG
866             027230                  25$:    CKLOOP                  ;SCOPE LOOP
                027230  104406                                                  TRAP    C$CLP1
867             027232                          ESCAPE  SUB             ;NO MORE CHECKS IF FATAL ERROR
                027232  104410                                                  TRAP    C$ESCAPE
                027234  000066                                                  .WORD   L10051-.
868             027236  012704  000310          MOV     #310,R4         ;START WITH LOC 310
869             027242  005002                  CLR     R2              ;MEMORY EXPECTED SHOULD BE 000000
870             027244  004737  016326          JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
871             027250  010465  000000  30$:    MOV     R4,TSDB(R5)     ;SELECT LOCATION SPECIFIED
872             027254  004737  016326          JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
873             027260  116501  000000          MOVB    TSBA(R5),R1     ;READ LOC CONTENTS
874             027264  120102                  CMPB    R1,R2           ;CHECK MEMORY FOR 000000
875             027266  001406                  BEQ     40$             ;BRANCH IF DATA OKAY
876             027270                          ERRDF   ERRNO,T5MEM,SFFMSG      ;MEMORY NOT ZERO AFTER INIT.
                027270  104455                                                  TRAP    C$ERDF
                027272  000770                                                  .WORD   504
                027274  027404                                                  .WORD   T5MEM
                027276  012072                                                  .WORD   SFFMSG
877             027300  005237  002220          INC     FATFLG          ;SET THE FATAL ERROR FLAG
878             027304                  40$:    CKLOOP
                027304  104406                                                  TRAP    C$CLP1
879             027306                          ESCAPE  SUB             ;EXIT ON FATAL ERROR
                027306  104410                                                  TRAP    C$ESCAPE
                027310  000012                                                  .WORD   L10051-.
880             027312  005204                  INC     R4              ;LOOK AT NEXT RAM LOC.
881             027314  020427  000400          CMP     R4,#400         ;AT TOP OF RAM ADDRESS SPACE
882             027320  001353                  BNE     30$             ;BRANCH TILL ALL MEMORY TESTED
883
884             027322                          ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                027322                                                          L1005i:
                027322  104403                                                  TRAP    C$ESUB
885
886             027324  005737  002220          TST     FATFLG          ;IS FATAL ERROR FLAG SET ?
887             027330  001402                  BEQ     50$             ;BRANCH IF NOT
888             027332  004737  017172          JSR     PC,CKDROP       ;NO LOOP, TRY TO DROP DEVICE
889             027336  004737  016446  50$:    JSR     PC,TSTLOOP      ;SHOULD WE DO ITERATIONS ?
890             027342  103002                  BCC     60$             ;BRANCH IF NOT
891             027344  000137  026420          JMP     T5LOOP          ;LOOP UNTIL COUNT EXPIRED
892             027350                  60$:    EXIT    TST             ;ALL DONE THIS TEST
                027350  104432                                                  TRAP    C$EXIT
                027352  000132                                                  .WORD   L10047-.
893
894                                     ;+
895                                     ;LOCAL TEXT MESSAGES FOR TEST
896                                     ;-
897
898             027354      105     170     164 TST5ID: .ASCIZ  'Extended Initialization'
899             027404      111     156     143 T5MEM:  .ASCIZ  'Incorrect RAM Data After Init'
900             027442      111     156     143 T5ADDR: .ASCIZ  'Incorrect RAM Address After Init'
901                                             .EVEN
902             027504                          ENDTST
                027504                                                  L10047:
                027504  104401                                                  TRAP    C$ETST
903
```

```
904
```

```
                        .SBTTL  TEST  6: COMMAND REJECT

                ;       THIS TEST VERIFIES THAT ALL COMMANDS OTHER THAN WRITE
                ;       CHARACTERISTICS ARE REJECTED DUE TO THE NEED BUFFER ADDRESS
                ;       (NBA) BIT BEING SET IN TSSR, AND THAT THE TSBA AND TSSR
                ;       REGISTERS ARE LEFT IN THE PROPER STATE AFTER EACH COMMAND IS
                ;       REJECTED.  THIS TEST CHECKS MICROPROCESSOR SEQUENCING, BASIC
                ;       COMMAND DECODING AND DATI DMA HANDLING.  THIS TEST CONTAINS TWO
                ;       SUBTESTS:  SUBTEST 1 SEQUENCES THROUGH ALL COMMAND WORDS (OTHER
                ;       THAN WRITE CHARACTERISTICS) WITH THE INTERRUPT ENABLE (IE) BIT
                ;       CLEAR AND VERIFIES THAT AN INTERRUPT IS NOT GENERATED BY THE
                ;       REJECTED COMMAND;  SUBTEST 2 PERFORMS SIMILARLY TO SUBTEST 1 BUT
                ;       SETS THE IE BIT IN EACH COMMAND WORD AND VERIFIES THAT AN
                ;       INTERRUPT IS GENERATED WHEN THE COMMAND IS REJECTED.  SUBTEST 1
                ;       SETS UP THE INTERRUPT SERVICE ROUTINE TO FLAG UNEXPECTED
                ;       INTERRUPTS.  THE COMMAND WORD IN THE COMMAND BUFFER IS
                ;       INITIALIZED TO 100000 (OCTAL) AND THE REMAINING THREE WORDS IN
                ;       THE COMMAND BUFFER ARE SET TO KNOWN UNIQUE PATTERNS.  THEN THE
                ;       FOLLOWING SEQUENCE IS PERFORMED:
                ;
                ;           1.  INITIALIZE THE CONTROLLER BY WRITING INTO THE TSSR;
                ;               PROPER INITIAL CONDITIONS ARE VERIFIED.
                ;
                ;           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.
                                ;
                                        BGNTST
                                                                                T6::
                                        MOV     #TST6ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
                                        JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
                                        MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
                                T6LOOP:
                                        BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
                                                                                T6.1:
                                                                                TRAP    C$BSUB

                                        SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERPUPTS
                                                                                MOV     #PRI00,R0
                                                                                TRAP    C$SPRI
                                        MOV     #T6PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
                                        MOV     #TSTBLK,R3              ;BLOCK OF TEST DATA
                                5$:     MOV     (R3)+,(R4)              ;INSERT THE NEXT TEST DATA WORD
                                        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$ERDF
                                                                                .WORD   601
                                                                                .WORD   SFIERR
                                                                                .WORD   SFIMSG
                                10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
                                        CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        BIC     #BIT7,(R4)              ;DISABLE INTERRUPTS
                                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
                                        JSR     PC,WAITF                ;WAIT FOR SSR TO SET
                                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        ERRDF   ERRNO,T6SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                                                                TRAP    C$ERDF
                                                                                .WORD   602
                                                                                .WORD   T6SSR
                                                                                .WORD   PKTSSR
                                        INC     FATFLG                  ;SET FATAL ERROR FLAG
                                15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                TRAP    C$CLP1
                                        ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
                                                                                TRAP    C$ESCAPE
                                                                                .WORD   L10053-.
                                        TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
                                        BEQ     22$                     ;BRANCH IF NOT
                                        ERRHRD  ERRNO,T6INT,PKTSSR
                                                                                TRAP    C$ERHRD
                                                                                .WORD   603
                                                                                .WORD   T6INT
                                                                                .WORD   PKTSSR
                                22$:    MOV     #SC!NBA!SSR!TSREJ,R2    ;EXPECTED CONTENTS OF TSSR
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
```

```
1012 027674 016501 000002          MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1013 027700 032701 000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1014 027704 001402                 BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1015 027706 052702 000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1016 027712 020201           25$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1017 027714 001404                 BEQ     30$                     ;OKAY IF MATCH
1021 027716                        ERRHRD  ERRNO,T6NBA,PKTSSR      ;NBA NOT SET TO REJECT
     027716 104456                                                          TRAP    C$ERHRD
     027720 001134                                                          .WORD   604
     027722 030430                                                          .WORD   T6NBA
     027724 012036                                                          .WORD   PKTSSR
1022 027726                  30$:  CKLOOP                          ;LOOP ON ERROR ?
     027726 104406                                                          TRAP    C$CLP1
1023 027730 004737 016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1024 027734 016501 000000          MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
1025 027740 010402                 MOV     R4,R2                   ;START OF THE PACKET
1026 027742 062702 000010          ADD     #10,R2                  ;EXPECT TSDA TO PACKET + 10
1027 027746 020102                 CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1028 027750 001404                 BEQ     35$                     ;ERROR IF NOT EQUAL
1032 027752                        ERRHRD  ERRNO,T6TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     027752 104456                                                          TRAP    C$ERHRD
     027754 001135                                                          .WORD   605
     027756 030671                                                          .WORD   T6TSBA
     027760 015464                                                          .WORD   EXPREC
1033
1034
1035 027762 004737 011104    35$:  JSR     PC,CKRAM                ;SEE IF DATA IN RAM IS CORRECT
1036 027766 103404                 BCS     40$                     ;BRANCH IF PACKET IN RAM IS CORRECT
1040 027770                        ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
     027770 104456                                                          TRAP    C$ERHRD
     027772 001136                                                          .WORD   606
     027774 004741                                                          .WORD   PKTRAM
     027776 015500                                                          .WORD   RAMERR
1041 030000                  40$:  ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     030000                                                                 10000$:
     030000 104405                                                          TRAP    C$ESEG
1042 030002 011300                 MOV     (R3),R0                 ;PACKET COMMAND WORD
1043 030004 042700 177740          BIC     #177740,R0              ;GET BITS 0-4
1044 030010 020027 000004          CMP     R0,#4                   ;DON'T TEST WRITE CHARACTERISTICS
1045 030014 001002                 BNE     45$                     ;BRANCH IF OK
1046 030016 062703 000002          ADD     #2,R3                   ;GET NEXT WORD FROM DATA TABLE
1047 030022 020327 003056    45$:  CMP     R3,#TBLEND              ;REACHED END OF TABLE ?
1048 030026 103002                 BHIS    50$                     ;BRANCH IF END OF TABLE
1049 030030 000137 027544          JMP     5$                      ;CONTINUE TEST WITH NEW DATA
1050
1051 030034                  50$:  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     030034                                                                 L10053:
     030034 104403                                                          TRAP    C$ESUB
1052
1053 030036 005737 002220          TST     FATFLG                  ;ANY FATAL ERRORS ?
1054 030042 001402                 BEQ     60$                     ;BRANCH IF NOT
1055 030044 004737 017172          JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
```

```
1057 030050                            60$:    BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     030050                                                                                    T6.2:
     030050  104402                                                                                   TRAP    C$BSUB
1058
1059 030052                                    SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     030052  012700  000000                                                                           MOV     #PRI00,R0
     030056  104441                                                                                   TRAP    C$SPRI
1060 030060  012704  030420                    MOV     #T6PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1061 030064  012703  002750                    MOV     #TSTBLK,R3              ;START OF TEST DATA
1062 030070  012314                    5$:     MOV     (R3)+,(R4)              ;PLACE NEXT DATA WORD IN PACKET
1063 030072                                    BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     030072  104404                                                                                   TRAP    C$BSEG
1064 030074  004737  015764                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1065 030100  103405                            BCS     10$                     ;BR IF SOFT INIT = OK
1069 030102  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1070 030104                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     030104  104455                                                                                   TRAP    C$ERDF
     030106  001137                                                                                   .WORD   607
     030110  003646                                                                                   .WORD   SFIERR
     030112  012024                                                                                   .WORD   SFIMSG
1071 030114  005037  002220            10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1072 030120  005037  002222                    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1073 030124  004737  016326                    JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1074 030130  052714  000200                    BIS     #BIT7,(R4)              ;ENABLE INTERRUPTS
1075 030134  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
1076 030140  004737  016240                    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
1077 030144  103407                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1078 030146  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1082 030150                                    ERRDF   ERRNO,T6SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     030150  104455                                                                                   TRAP    C$ERDF
     030152  001140                                                                                   .WORD   608
     030154  030455                                                                                   .WORD   T6SSR
     030156  012036                                                                                   .WORD   PKTSSR
1083 030160  005237  002220                    INC     FATFLG                  ;SET FATAL ERROR FLAG
1084 030164                            15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     030164  104406                                                                                   TRAP    C$CLP1
1085 030166                                    ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
     030166  104410                                                                                   TRAP    C$ESCAPE
     030170  000170                                                                                   .WORD   L10054-.
1086 030172  005737  002222                    TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1087 030176  001004                            BNE     22$                     ;BRANCH IF YES
1091 030200                                    ERRHRD  ERRNO,T6NINT,PKTSSR     ;REPORT ERROR IF NO INTERRUPT
     030200  104456                                                                                   TRAP    C$ERHRD
     030202  001141                                                                                   .WORD   609
     030204  030611                                                                                   .WORD   T6NINT
     030206  012036                                                                                   .WORD   PKTSSR
1092 030210  012702  102206            22$:    MOV     #SC!NBA!SSR!TSREJ,R2    ;EXPECTED CONTENTS OF TSSR
1093 030214  004737  016326                    JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1094 030220  016501  000002                    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1095 030224  032701  000100                    BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1096 030230  001402                            BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1097 030232  052702  000100                    BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1098 030236  020201                    25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1099 030240  001404                            BEQ     30$                     ;OKAY IF MATCH
1103 030242                                    ERRHRD  ERRNO,T6NBA,PKTSSR      ;NBA NOT SET TO REJECT
     030242  104456                                                                                   TRAP    C$ERHRD
     030244  001142                                                                                   .WORD   610
```

```
                030246  030430                                                          .WORD   T6NBA
                030250  012036                                                          .WORD   PKTSSR
1104            030252                  30$:    CKLOOP                  ;LOOP ON ERROR ?
                030252  104406                                                          TRAP    C$CLP1
1105            030254  004737  016326          JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
1106            030260  016501  000000          MOV     TSBA(R5),R1     ;GET TSBA REGISTER CONTENTS
1107            030264  010402                  MOV     R4,R2           ;START OF THE PACKET
1108            030266  062702  000010          ADD     #10,R2          ;EXPECT TSDA TO PACKET + 10
1109            030272  020102                  CMP     R1,R2           ;COMPARE EXPECTED TO RECEIVED
1110            030274  001404                  BEQ     35$             ;ERROR IF NOT EQUAL
1114            030276                          ERRHRD  ERRNO,T6TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
                030276  104456                                                          TRAP    C$ERHRD
                030300  001143                                                          .WORD   611
                030302  030671                                                          .WORD   T6TSBA
                030304  015464                                                          .WORD   EXPREC
1115
1116
1117            030306  004737  011104  35$:    JSR     PC,CKRAM        ;SEE IF DATA IN RAM IS CORRECT
1118            030312  103404                  BCS     40$             ;BRANCH IF PACKET IN RAM IS CORRECT
1122            030314                          ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
                030314  104456                                                          TRAP    C$ERHRD
                030316  001144                                                          .WORD   612
                030320  004741                                                          .WORD   PKTRAM
                030322  015500                                                          .WORD   RAMERR
1123            030324                  40$:    ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                030324                                                                  10000$:
                030324  104405                                                          TRAP    C$ESEG
1124            030326  011300                  MOV     (R3),R0         ;NEXT PACKET COMMAND WORD
1125            030330  042700  177740          BIC     #177740,R0      ;GET BITS 0-4
1126            030334  020027  000004          CMP     R0,#4           ;DON'T TEST WRITE CHARACTERISTICS
1127            030340  001002                  BNE     45$             ;BRANCH IF NOT WRITE CHARACTERISTICS
1128            030342  062703  000002          ADD     #2,R3           ;BY-PASS WRITE CHARACTERISTCS
1129            030346  020327  003056  45$:    CMP     R3,#TBLEND      ;HAVE WE COMPLETED DATA TABLE ?
1130            030352  103002                  BHIS    50$             ;BRANCH IF ALL TESTED
1131            030354  000137  030070          JMP     5$              ;TEST WITH NEXT DATA
1132
1133            030360                  50$:    ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                030360                                                                  L10054:
                030360  104403                                                          TRAP    C$ESUB
1134            030362  005737  002220          TST     FATFLG          ;ANY FATAL ERRORS ?
1135            030366  001402                  BEQ     60$             ;BRANCH IF NOT
1136            030370  004737  017172          JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
1137            030374  004737  016446  60$:    JSR     PC,TSTLOOP      ;SHOULD WE DO ITERATIONS ?
1138            030400  103002                  BCC     62$             ;BRANCH IF NOT
1139            030402  000137  027524          JMP     T6LOOP          ;LOOP UNTIL COUNT EXPIRED
1140            030406                  62$:    EXIT    TST             ;ALL DONE THIS TEST
                030406  104432                                                          TRAP    C$EXIT
                030410  000352                                                          .WORD   L10052-.
1141
1142                                    ;+
1143                                    ;LOCAL STORAGE FOR THIS TEST
1144                                    ;-
1145
1147                    030420                  .=<.+10>&177770
1149            030420                  T6PACKET:
1150            030420  000000                  .WORD   0               ;COMMAND PACKET FOR TEST
1151            030422  052525                  .WORD   052525          ;WILL CONTAIN VARIABLE COMMANDS
```

```
 1152 030424  125252                        .WORD   125252
 1153 030426  052525                        .WORD   052525
 1154
 1155
 1156                                ;+
 1157                                ;LOCAL TEXT MESSAGES FOR TEST
 1158                                ;-
 1159
 1160 030430      103     157     155  T6NBA:   .ASCIZ  'Command Not Rejected'
 1161 030455      103     157     156  T6SSR:   .ASCIZ  'Contents of TSSR Incorrect After Write Packet'
 1162 030533      125     156     145  T6INT:   .ASCIZ  'Unexpected Interrupt Received On Write Packet'
 1163 030611      105     170     160  T6NINT:  .ASCIZ  'Expected Interrupt Not Received On Write Packet'
 1164 030671      111     156     143  T6TSBA:  .ASCIZ  'Incorrect TSBA Address After Packet Write'
 1165 030743      103     157     155  TST6ID:  .ASCIZ  'Command Reject'
 1166                                        .EVEN
 1167 030762                                ENDTST
      030762                                                          L10052:
      030762  104401                                                          TRAP    C$ETST
```

```
1169
1170                                    .SBTTL  TEST  7: WRITE CHARACTERISTICS
1171
1172
1173                            ;       THIS TEST VERIFIES BASIC OPERATION OF THE WRITE CHARACTERISTICS
1174                            ;       COMMAND.  IT VERIFIES THAT THE COMMAND BLOCK AND CHARACTERISTICS
1175                            ;       DATA BLOCK ARE FETCHED PROPERLY FROM CPU MEMORY, THE NEED BUFFER
1176                            ;       ADDRESS (NBA) BIT IN TSSR IS HANDLED PROPERLY, AND THAT A PROPER
1177                            ;       MESSAGE PACKET IS STORED, WHERE APPROPRIATE.  THIS TEST DOES NOT
1178                            ;       CHECK THAT THE VARIOUS FUNCTIONS ENABLED BY CHARACTERISTIC MODE
1179                            ;       DATA BITS OPERATE PROPERLY;  THE FUNCTIONING OF THESE BITS IS
1180                            ;       VERIFIED IN SUBSEQUENT TESTS.  ALL COMMANDS EXECUTED IN THIS
1181                            ;       TEST HAVE THE INTERRUPT ENABLE (IE) BIT CLEARED TO ZERO, SO NO
1182                            ;       INTERRUPTS SHOULD BE GENERATED.  HOWEVER, THE PROGRAM RUNS AT
1183                            ;       PROCESSOR PRIORITY 0, WITH THE INTERRUPT SERVICE ROUTINE SET UP
1184                            ;       TO FLAG UNEXPECTED INTERRUPTS.  IF AN INTERRUPT OCCURS, A
1185                            ;       PROBLEM EXISTS IN EITHER THE LSI-11 BUS INTERFACE SECTION OR IN
1186                            ;       THE ROM OR PIPELINE.
1187
1188                            ;       THIS TEST CHECKS VARIOUS MICROPROGRAM SEQUENCES, COMMAND
1189                            ;       DECODING, DMA LOGIC, AND BASIC PACKET PROTOCOL HANDLING.  THIS
1190                            ;       IS THE FIRST TEST IN WHICH DATO DMA CYCLES (FOR STORING THE
1191                            ;       MESSAGE PACKET) ARE PERFORMED.  ANY ERRORS IN THE BODY OF THE
1192                            ;       TEST (I.E. ERRORS OTHER THAN INITIALIZATION ERRORS RELATED TO
1193                            ;       THE TRANSPORT BUS) DEFINITELY INDICATE A BAD M7196 MODULE.
1194
1195
1196
1197 030764                                    BGNTST
      030764                                                                    T7::
1202 030764  012700  034253                    MOV     #TST7ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
1203 030770  004737  016500                    JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
1204 030774  012737  000024  002214            MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
1205 031002                             T7LOOP:
1206 031002                                    BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
      031002                                                                    T7.1:
      031002  104402                                                                    TRAP    C$BSUB
1207 031004  004737  034302                    JSR     PC,T7REST               ;SET PACKET TO START-UP VALUES
1208
1209 031010                                    SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      031010  012700  000000                                                            MOV     #PRI00,R0
      031014  104441                                                                    TRAP    C$SPRI
1210 031016  012703  002762                    MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
1211 031022  012704  033120                    MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1212 031026  012764  000010  000006            MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
1213 031034                             5$:
1214 031034                                    BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      031034  104404                                                                    TRAP    C$BSEG
1215
1216 031036  004737  015764                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1217 031042  103405                            BCS     10$                     ;BR IF SOFT INIT = OK
1221 031044  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1222 031046                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      031046  104455                                                                    TRAP    C$ERDF
      031050  001275                                                                    .WORD   701
      031052  003646                                                                    .WORD   SFIERR
      031054  012024                                                                    .WORD   SFIMSG
```

```
1223 031056  005037  002220          10$:   CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1224 031062  005037  002222                 CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1225 031066  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
1226 031072  004737  016326                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1227 031076  103407                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1228 031100  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1232 031102                                ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     031102  104455                                                                TRAP    C$ERDF
     031104  001276                                                                .WORD   702
     031106  033721                                                                .WORD   T7SSR
     031110  012036                                                                .WORD   PKTSSR
1233 031112  005237  002220                 INC     FATFLG                  ;SET FATAL ERROR FLAG
1234 031116                        15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     031116  104406                                                                TRAP    C$CLP1
1235 031120                                ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     031120  104410                                                                TRAP    C$ESCAPE
     031122  000152                                                                .WORD   10000$-.
1236 031124  005737  002222                 TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1237 031130  001404                        BEQ     22$                     ;BRANCH IF NOT
1241 031132                                ERRHRD  ERRNO,T7INT,PKTSSR
     031132  104456                                                                TRAP    C$ERHRD
     031134  001277                                                                .WORD   703
     031136  034101                                                                .WORD   T7INT
     031140  012036                                                                .WORD   PKTSSR
1242 031142  016501  000002          22$:   MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1243 031146  012702  000200                 MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
1244 031152  032701  000100                 BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1245 031156  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1246 031160  052702  000100                 BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1247 031164  020201                25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1248 031166  001404                        BEQ     30$                     ;OKAY IF MATCH
1252 031170                                ERRHRD  ERRNO,T7NBA,PKTSSR      ;NBA NOT ZERO
     031170  104456                                                                TRAP    C$ERHRD
     031172  001300                                                                .WORD   704
     031174  033260                                                                .WORD   T7NBA
     031176  012036                                                                .WORD   PKTSSR
1253 031200                        30$:    CKLOOP                          ;LOOP ON ERROR ?
     031200  104406                                                                TRAP    C$CLP1
1254 031202  004737  016326                 JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1255 031206  016501  000000                 MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
1256 031212  012702  033146                 MOV     #T7BFR,R2               ;START OF THE DATA BUFFER
1257 031216  032762  000200  000012         BIT     #BIT7,XST2(R2)          ;IS EXTENDED FEATURES BIT SET ?
1258 031224  001404                        BEQ     32$                     ;BRANCH IF EXTENDED FEATURES OFF
1259 031226  005237  002224                 INC     EXTFEA                  ;SET EXTENDED FEATURES FOR SUBTEST 6
1260 031232  062702  000002                 ADD     #2,R2                   ;EXTRA WORD IF SPECIAL FEATURES
1261 031236  062702  000016          32$:   ADD     #14.,R2                 ;EXPECTED CONTENTS OF TSDA
1262 031242  020102                        CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1263 031244  001404                        BEQ     35$                     ;ERROR IF NOT EQUAL
1267 031246                                ERRHRD  ERRNO,T7TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     031246  104456                                                                TRAP    C$ERHRD
     031250  001301                                                                .WORD   705
     031252  034170                                                                .WORD   T7TSBA
     031254  015464                                                                .WORD   EXPREC
1268
1269
1270 031256  004737  011104          35$:   JSR     PC,CKRAM                ;SEE IF DATA IN RAM IS CORRECT
1271 031262  103404                        BCS     40$                     ;BRANCH IF PACKET IN RAM IS CORRECT
```

```
1275 031264                                ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
     031264  104456                                                            TRAP    C$ERHRD
     031266  001302                                                            .WORD   706
     031270  004741                                                            .WORD   PKTRAM
     031272  015500                                                            .WORD   RAMERR
1276
1277 031274                           40$:  ENDSEG                            ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     031274                                                                                10000$:
     031274  104405                                                                    TRAP    C$ESEG
1278
1279 031276  012364  000006                 MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
1280 031302  020327  003056                 CMP     R3,#TBLEND              ;HAS ALL DATA BEEN TESTED ?
1281 031306  103002                         BHIS    55$                     ;BRANCH IF ALL DATA DONE
1282 031310  000137  031034                 JMP     5$                      ;BRANCH TILL BACK TO ZERO
1283
1284 031314                           55$:  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     031314                                                                                L10056:
     031314  104403                                                                    TRAP    C$ESUB
1285
1286 031316  005737  002220                 TST     FATFLG                  ;ANY FATAL ERRORS ?
1287 031322  001402                         BEQ     60$                     ;BRANCH IF NOT
1288 031324  004737  017172                 JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
1289 031330                           60$:
```

```
                                        ;+
                                        ;
                                        ;TEST 7, SUBTEST 2
                                        ;
                                        ;CHECK THAT UNUSED BITS BEING SET CAUSES
                                        ;WRITE CHARACTERISTICS COMMAND TO BE REJECTED
                                        ;
                                        ;-

                        BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
                                                                T7.2:
                                                                TRAP    C$BSUB

                        SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
                                                                MOV     #PRI00,R0
                                                                TRAP    C$SPRI
                        MOV     #T72DATA,R3             ;START OF TEST DATA FOR SUBTEST
5$:                     MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
                        JSR     PC,T7REST               ;RESTORE PACKET TO STARTING VALUES

                        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$ERDF
                                                                .WORD   707
                                                                .WORD   SFIERR
                                                                .WORD   SFIMSG
10$:                    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
                        MOV     R4,R0                   ;START OF THE COMMAND PACKET
                        ADD     (R3),R0                 ;OFFSET TO THE DATA WORD TO TEST
                        BIS     2(R3),(R0)              ;SET THE DATA BITS TO BE TESTED
                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
                        JSR     PC,WAITF                ;WAIT FOR SSR TO SET
                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                        ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                                                TRAP    C$ERDF
                                                                .WORD   708
                                                                .WORD   T7SSR
                                                                .WORD   PKTSSR
15$:                    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                TRAP    C$CLP1
                        ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
                                                                TRAP    C$ESCAPE
                                                                .WORD   10000$-.
                        TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
                        BEQ     22$                     ;BRANCH IF NOT
                        ERRHRD  ERRNO,T7INT,PKTSSR
                                                                TRAP    C$ERHRD
                                                                .WORD   709
                                                                .WORD   T7INT
                                                                .WORD   PKTSSR
22$:                    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
```

```
1337 031466  012702  102206          MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
1338 031472  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1339 031476  001402                  BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1340 031500  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1341 031504  020201              25$: CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1342 031506  001414                  BEQ     30$                     ;OKAY IF MATCH
1343 031510  010100                  MOV     R1,R0                   ;DATA FROM TSSR
1344 031512                          XOR     R2,R0                   ;FIND BITS IN ERROR
1345 031522  020027  002000          CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
1346 031526  001404                  BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
1350 031530                          ERRHRD  ERRNO,T72REJ,PKTSSR     ;COMMAND NOT REJECTED
     031530  104456                                                          TRAP    C$ERHRD
     031532  001306                                                          .WORD   710
     031534  033333                                                          .WORD   T72REJ
     031536  012036                                                          .WORD   PKTSSR
1351 031540                      30$: CKLOOP                          ;LOOP ON ERROR ?
     031540  104406                                                          TRAP    C$CLP1
1352 031542  032701  002000          BIT     #NBA,R1                 ;IS NBA BIT SET ?
1353 031546  001004                  BNE     35$                     ;OKAY IF NBA SET
1357 031550                          ERRHRD  ERRNO,T72NBA,PKTSSR     ;NBA NOT SET
     031550  104456                                                          TRAP    C$ERHRD
     031552  001307                                                          .WORD   711
     031554  033202                                                          .WORD   T72NBA
     031556  012036                                                          .WORD   PKTSSR
1358 031560                      35$:
1359 031560                          ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     031560                                                                10000$:
     031560  104405                                                          TRAP    C$ESEG
1360
1361 031562  062703  000004          ADD     #4,R3                   ;POINT TO NEXT DATA PAIR
1362 031566  020327  033202          CMP     R3,#T72DONE             ;COMPARE TO END OF TEST DATA
1363 031572  103002                  BHIS    57$                     ;BRANCH IF ALL DATA TESTED
1364 031574  000137  031344          JMP     5$                      ;BRANCH TILL BACK TO ZERO
1365
1366 031600                      57$: ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     031600                                                                L10057:
     031600  104403                                                          TRAP    C$ESUB
```

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

                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
                                                                                        T7.3:
031602  104402                                                                          TRAP    C$BSUB

                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
031604  012700  000000                                                                  MOV     #PRI00,R0
031610  104441                                                                          TRAP    C$SPRI
031612  012703  000001        MOV     #1,R3                   ;STARTING BYTE COUNT
031616  012704  033120  5$:    MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
031622  004737  034302        JSR     PC,T7REST               ;RESTORE PACKET TO STARTING VALUES

                                BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
031626  104404                                                                          TRAP    C$BSEG
031630  004737  015764        JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
031634  103405                BCS     10$                     ;BR IF SOFT INIT = OK
031636  010001                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
031640  104455                                                                          TRAP    C$ERDF
031642  001310                                                                          .WORD   712
031644  003646                                                                          .WORD   SFIERR
031646  012024                                                                          .WORD   SFIMSG
031650  005037  002222  10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
031654  010364  000006        MOV     R3,PKBCNT(R4)           ;INSERT THE BYTE COUNT FOR TEST
031660  010465  000000        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
031664  004737  016240        JSR     PC,WAITF                ;WAIT FOR SSR TO SET
031670  103405                BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
031672  010001                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
031674  104455                                                                          TRAP    C$ERDF
031676  001311                                                                          .WORD   713
031700  033721                                                                          .WORD   T7SSR
031702  012036                                                                          .WORD   PKTSSR
031704  104406  15$:    CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
                                                                                        TRAP    C$CLP1
                                ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
031706  104410                                                                          TRAP    C$ESCAPE
031710  000116                                                                          .WORD   10000$-.
031712  005737  002222        TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
031716  001404                BEQ     22$                     ;BRANCH IF NOT
                                ERRHRD  ERRNO,T7INT,PKTSSR
031720  104456                                                                          TRAP    C$ERHRD
031722  001312                                                                          .WORD   714
031724  034101                                                                          .WORD   T7INT
031726  012036                                                                          .WORD   PKTSSR
031730  016501  000002  22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
031734  012702  102206        MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
031740  032701  000100        BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
```

```
1414 031744  001402                    BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1415 031746  052702  000100            BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1416 031752  020201              25$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1417 031754  001414                    BEQ     30$                     ;OKAY IF MATCH
1418 031756  010100                    MOV     R1,R0                   ;DATA FROM TSSR
1419 031760                            XOR     R2,R0                   ;FIND BITS IN ERROR
1420 031770  020027  002000            CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
1421 031774  001404                    BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
1425 031776                            ERRHRD  ERRNO,T73REJ,PKTSSR     ;COMMAND NOT REJECTED
     031776  104456                                                            TRAP    C$ERHRD
     032000  001313                                                            .WORD   715
     032002  033432                                                            .WORD   T73REJ
     032004  012036                                                            .WORD   PKTSSR
1426 032006                      30$:  CKLOOP                          ;LOOP ON ERROR ?
     032006  104406                                                            TRAP    C$CLP1
1427 032010  032701  002000            BIT     #NBA,R1                 ;IS NBA BIT SET ?
1428 032014  001004                    BNE     35$                     ;OKAY IF NBA SET
1432 032016                            ERRHRD  ERRNO,T72NBA,PKTSSR     ;NBA NOT SET
     032016  104456                                                            TRAP    C$ERHRD
     032020  001314                                                            .WORD   716
     032022  033202                                                            .WORD   T72NBA
     032024  012036                                                            .WORD   PKTSSR
1433 032026                      35$:
1434 032026                            ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     032026                                                                 10000$:
     032026  104405                                                            TRAP    C$ESEG
1435
1436 032030  005203                    INC     R3                      ;NEXT BYTE COUNT
1437 032032  020327  000006            CMP     R3,#6                   ;TESTED ALL INVALID ?
1438 032036  002002                    BGE     57$                     ;BRANCH IF TEST DONE
1439 032040  000137  031616            JMP     5$                      ;BRANCH TILL BACK TO ZERO
1440
1441 032044                      57$:  ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     032044                                                                 L10060:
     032044  104403                                                            TRAP    C$ESUB
```

```
                                        ;+
                                        ;
                                        ;TEST 7, SUBTEST 4
                                        ;
                                        ;SUBTEST TO VERIFY THAT A WRITE CHARACTERISTICS COMMAND IS
                                        ;REJECTED IF AN ILLEGAL DATA BLOCK ADDRESS IS ISSUED.
                                        ;
                                        ;-

032046                                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
032046                                                                                      T7.4:
032046  104402                                                                              TRAP    C$BSUB

032050                                          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
032050  012700  000000                                                                      MOV     #PRI00,R0
032054  104441                                                                              TRAP    C$SPRI
032056  012703  033166                5$:     MOV     #T72DATA,R3             ;START OF TEST DATA FOR SUBTEST
032062  012704  033120                        MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
032066  004737  034302                        JSR     PC,T7REST               ;RESTORE PACKET TO STARTING VALUES

032072  004737  015764                        JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
032076  103405                              BCS     10$                     ;BR IF SOFT INIT = OK
032100  010001                              MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
032102                                      ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
032102  104455                                                                              TRAP    C$ERDF
032104  001315                                                                              .WORD   717
032106  003646                                                                              .WORD   SFIERR
032110  012024                                                                              .WORD   SFIMSG
032112  005037  002222                10$:     CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
032116  052737  000001  033130              BIS     #1,T7DATA               ;MAKE ADDRESS ODD
032124  010465  000000                      MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
032130  004737  016240                      JSR     PC,WAITF                ;WAIT FOR SSR TO SET
032134  103405                              BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
032136  010001                              MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
032140                                      ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
032140  104455                                                                              TRAP    C$ERDF
032142  001316                                                                              .WORD   718
032144  033721                                                                              .WORD   T7SSR
032146  012036                                                                              .WORD   PKTSSR
032150                                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
032150  104406                                                                              TRAP    C$CLP1
032152                                          ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
032152  104410                                                                              TRAP    C$ESCAPE
032154  000116                                                                              .WORD   L10061-.
032156  005737  002222                        TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
032162  001404                              BEQ     22$                     ;BRANCH IF NOT
032164                                      ERRHRD  ERRNO,T7INT,PKTSSR
032164  104456                                                                              TRAP    C$ERHRD
032166  001317                                                                              .WORD   719
032170  034101                                                                              .WORD   T7INT
032172  012036                                                                              .WORD   PKTSSR
032174  016501  000002                22$:     MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
032200  012702  102206                      MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
032204  032701  000100                      BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
032210  001402                              BEQ     25$                     ;BRANCH IF NOT OFF-LINE
032212  052702  000100                      BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
```

```
1490 032216  020201                25$:   CMP     R2,R1                ;DOES EXPECTED MATCH RECEIVED ?
1491 032220  001414                       BEQ     30$                  ;OKAY IF MATCH
1492 032222  010100                       MOV     R1,R0                ;DATA FROM TSSR
1493 032224                               XOR     R2,R0                ;FIND BITS IN ERROR
1494 032234  020027  002000               CMP     R0,#NBA              ;IS NBA ONLY BIT IN ERROR ?
1495 032240  001404                       BEQ     30$                  ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
1499 032242                               ERRHRD  ERRNO,T74REJ,PKTSSR  ;COMMAND NOT REJECTED
     032242  104456                                                            TRAP    C$ERHRD
     032244  001320                                                            .WORD   720
     032246  033525                                                            .WORD   T74REJ
     032250  012036                                                            .WORD   PKTSSR
1500 032252                        30$:   CKLOOP                       ;LOOP ON ERROR ?
     032252  104406                                                            TRAP    C$CLP1
1501 032254  032701  002000               BIT     #NBA,R1              ;IS NBA BIT SET ?
1502 032260  001004                       BNE     35$                  ;OKAY IF NBA SET
1506 032262                               ERRHRD  ERRNO,T72NBA,PKTSSR  ;NBA NOT SET
     032262  104456                                                            TRAP    C$ERHRD
     032264  001321                                                            .WORD   721
     032266  033202                                                            .WORD   T72NBA
     032270  012036                                                            .WORD   PKTSSR
1507
1508 032272                        35$:   ENDSUB                       ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     032272                                                                  L10061:
     032272  104403                                                            TRAP    C$ESUB
1509
```

```
                                        ;+
                                        ;
                                        ;TEST 7, SUBTEST 5
                                        ;
                                        ;CHECK THAT WRITE CHARACTERISTICS COMMAND IS REJECTED IF THE
                                        ;MESSAGE BUFFER LENGTH SPECIFIES AN INVALID COUNT (LESS THAN 14)
                                        ;
                                        ;-

                        BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
                                                                        T7.5:
                                                                        TRAP    C$BSUB
                        SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
                                                                        MOV     #PRI00,R0
                                                                        TRAP    C$SPRI
                        MOV     #1,R3                   ;STARTING BUFFER LENGTH
5$:                     MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
                        JSR     PC,T7REST               ;RESTORE PACKET TO STARTING VALUES

                        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$ERDF
                                                                        .WORD   722
                                                                        .WORD   SFIERR
                                                                        .WORD   SFIMSG
10$:                    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
                        MOV     R3,T7DATA+4             ;INSERT THE BAD MESSAGE LENGTH
                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
                        JSR     PC,WAITF                ;WAIT FOR SSR TO SET
                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                        ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                                                        TRAP    C$ERDF
                                                                        .WORD   723
                                                                        .WORD   T7SSR
                                                                        .WORD   PKTSSR
15$:                    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                        TRAP    C$CLP1
                        ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
                                                                        TRAP    C$ESCAPE
                                                                        .WORD   10000$-.
                        TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
                        BEQ     22$                     ;BRANCH IF NOT
                        ERRHRD  ERRNO,T7INT,PKTSSR
                                                                        TRAP    C$ERHRD
                                                                        .WORD   724
                                                                        .WORD   T7INT
                                                                        .WORD   PKTSSR
22$:                    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
                        MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
                        BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
```

```
1557 032436  001402                    BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1558 032440  052702  000100            BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1559 032444  020201              25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1560 032446  001414                    BEQ     30$                     ;OKAY IF MATCH
1561 032450  010100                    MOV     R1,R0                   ;DATA FROM TSSR
1562 032452                             XOR     R2,R0                   ;FIND BITS IN ERROR
1563 032462  020027  002000            CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
1564 032466  001404                    BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
1568 032470                             ERRHRD  ERRNO,T75REJ,PKTSSR     ;COMMAND NOT REJECTED
     032470  104456                                                                    TRAP    C$ERHRD
     032472  001325                                                                    .WORD   725
     032474  033623                                                                    .WORD   T75REJ
     032476  012036                                                                    .WORD   PKTSSR
1569 032500              30$:   CKLOOP                                  ;LOOP ON ERROR ?
     032500  104406                                                                    TRAP    C$CLP1
1570 032502  032701  002000            BIT     #NBA,R1                 ;IS NBA BIT SET ?
1571 032506  001004                    BNE     35$                     ;OKAY IF NBA SET
1575 032510                             ERRHRD  ERRNO,T72NBA,PKTSSR     ;NBA NOT SET
     032510  104456                                                                    TRAP    C$ERHRD
     032512  001326                                                                    .WORD   726
     032514  033202                                                                    .WORD   T72NBA
     032516  012036                                                                    .WORD   PKTSSR
1576 032520              35$:
1577 032520                             ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     032520                                                                         10000$:
     032520  104405                                                                    TRAP    C$ESEG
1578
1579 032522  005203                    INC     R3                      ;NEXT BUFFER LENGTH
1580 032524  020327  000016            CMP     R3,#14.                 ;HAVE ALL BAD VALUES BEEN TESTED ?
1581 032530  002002                    BGE     57$                     ;BRANCH IF ALL TESTED
1582 032532  000137  032310            JMP     5$                      ;BRANCH TILL BACK TO ZERO
1583
1584 032536              57$:   ENDSUB                                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     032536                                                                         L10062:
     032536  104403                                                                    TRAP    C$ESUB
```

```
                                        ;+
                                        ;
                                        ;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.
                                        ;
                                        ;-
032540                                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
032540                                                                                  T7.6:
032540  104402                                                                          TRAP    C$BSUB
032542  005737  002224                        TST     EXTFEA                  ;IS EXTENDED FEATURES SOFT. SW SET?
032546  001002                                BNE     4$                      ;BR, IF SOFTWARE SWITCH IS SET (ON)
032550                                        EXIT    TST                     ;NO EXTENDED FEATURES EXIT THIS TEST
032550  104432                                                                          TRAP    C$EXIT
032552  001576                                                                          .WORD   L10055-.
032554  004737  034302          4$:           JSR     PC,T7REST               ;SET PACKET TO START-UP VALUES
                                        ;
032560                                        SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
032560  012700  000000                                                                  MOV     #PRI00,R0
032564  104441                                                                          TRAP    C$SPRI
032566  012703  002764                        MOV     #TSTBLK+12.,R3          ;START OF TEST DATA
032572  012704  033120                        MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
032576  012737  000200  033140                MOV     #200,T7SP               ;SPECIAL BIT SET FOR EXTFEA RAM RD
032604  012764  000012  000006                MOV     #10.,PKBCNT(R4)         ;START WITH EXTENDED FEATURES VALUE
032612                          5$:
032612                                        BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
032612  104404                                                                          TRAP    C$BSEG
                                        ;
032614  004737  015764                        JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
032620  103405                                BCS     10$                     ;BR IF SOFT INIT = OK
032622  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
032624                                        ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
032624  104455                                                                          TRAP    C$ERDF
032626  001327                                                                          .WORD   727
032630  003646                                                                          .WORD   SFIERR
032632  012024                                                                          .WORD   SFIMSG
032634  005037  002220          10$:          CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
032640  005037  002222                        CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
032644  010465  000000                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
032650  004737  016326                        JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
032654  103407                                BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
032656  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
032660                                        ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
032660  104455                                                                          TRAP    C$ERDF
032662  001330                                                                          .WORD   728
032664  033721                                                                          .WORD   T7SSR
032666  012036                                                                          .WORD   PKTSSR
032670  005237  002220                        INC     FATFLG                  ;SET FATAL ERROR FLAG
032674          15$:          CKLOOP                                          ;LOOP ON ERROR, IF FLAG SET
032674  104406                                                                          TRAP    C$CLP1
032676                                        ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
032676  104410                                                                          TRAP    C$ESCAPE
```

```
                032700  000156
                                                                                .WORD   10000$-.
1632            032702  005737  002222          TST     INTRECV         ;DID AN~INTERRUPT OCCUR ?
1633            032706  001404                  BEQ     22$             ;BRANCH IF NOT
1637            032710                          ERRHRD  ERRNO,T7INT,PKTSSR
                032710  104456                                          TRAP    C$ERHRD
                032712  001331                                          .WORD   729
                032714  034101                                          .WORD   T7INT
                032716  012036                                          .WORD   PKTSSR
1638            032720  016501  000002  22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
1639            032724  012702  000200          MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
1640            032730  032701  000100          BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
1641            032734  001402                  BEQ     25$             ;BRANCH IF NOT OFF-LINE
1642            032736  052702  000100          BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
1643            032742  020201          25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
1644            032744  001404                  BEQ     30$             ;OKAY IF MATCH
1648            032746                          ERRHRD  ERRNO,T7NBA,PKTSSR      ;NBA NOT ZERO
                032746  104456                                          TRAP    C$ERHRD
                032750  001332                                          .WORD   730
                032752  033260                                          .WORD   T7NBA
                032754  012036                                          .WORD   PKTSSR
1649            032756                  30$:    CKLOOP                  ;LOOP ON ERROR ?
                032756  104406                                          TRAP    C$CLP1
1650            032760  004737  016326          JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
1651            032764  016501  000000          MOV     TSBA(R5),R1     ;GET TSBA REGISTER CONTENTS
1652            032770  012702  033146          MOV     #T7BFR,R2       ;START OF THE DATA BUFFER
1653            032774  032762  000200  000012  BIT     #BIT7,XST2(R2)  ;IS EXTENDED FEATURES BIT SET ?
1654            033002  001402                  BEQ     32$             ;BRANCH IF EXTENDED FEATURES OFF
1655            033004  062702  000002          ADD     #2,R2           ;EXTRA WORD IF SPECIAL FEATURES
1656            033010  062702  000016  32$:    ADD     #14.,R2         ;EXPECTED CONTENTS OF TSDA
1657            033014  020102                  CMP     R1,R2           ;COMPARE EXPECTED TO RECEIVED
1658            033016  001404                  BEQ     35$             ;ERROR IF NOT EQUAL
1662            033020                          ERRHRD  ERRNO,T7TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
                033020  104456                                          TRAP    C$ERHRD
                033022  001333                                          .WORD   731
                033024  034170                                          .WORD   T7TSBA
                033026  015464                                          .WORD   EXPREC
1663
1664
1665            033030  012704  033130  35$:    MOV     #T7DATA,R4      ;SET POINTER FOR CHECKER
1666            033034  004737  011214          JSR     PC,CKRAM2       ;SEE IF DATA IN RAM IS CORRECT
1667            033040  103404                  BCS     40$             ;BRANCH IF PACKET IN RAM IS CORRECT
1671            033042                          ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
                033042  104456                                          TRAP    C$ERHRD
                033044  001334                                          .WORD   732
                033046  004741                                          .WORD   PKTRAM
                033050  015500                                          .WORD   RAMERR
1672
1673            033052  012704  033120  40$:    MOV     #T7PACKET,R4    ;RESET PACKET POINTER
1674            033056                          ENDSEG                  ;<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<
                033056                                                  10000$:
                033056  104405                                          TRAP    C$ESEG
1675
1676            033060  012364  000006          MOV     (R3)+,PKTCNT(R4)        ;SET THE TEST WORD
1677            033064  020327  003056          CMP     R3,#TBLEND      .       ;HAS ALL DATA BEEN TESTED ?
1678            033070  103002                  BHIS    55$             ;BRANCH IF ALL DATA DONE
1679            033072  000137  032612          JMP     5$              ;BRANCH TILL BACK TO ZERO
1680
```

```
1681 033076                          55$:    ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     033076                                                                                L10063:
     033076  104403                                                                              TRAP    C$ESUB
1682
1683 033100  005737  002220                  TST     FATFLG                  ;ANY FATAL ERRORS ?
1684 033104  001402                          BEQ     60$                     ;BRANCH IF NOT
1685 033106  004737  017172                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
1686 033112                          60$:
1687 033112                                  EXIT    TST                     ;ALL DONE THIS TEST
     033112  104432                                                                              TRAP    C$EXIT
     033114  001234                                                                              .WORD   L10055-.
1688
1689                                 ;+
1690                                 ;LOCAL STORAGE FOR THIS TEST
1691                                 ;-
1692
1694         033120                          .=<.+10>&177770
1696 033120                          T7PACKET:
1697 033120  100004                          .WORD   100004                  ;COMMAND PACKET FOR TEST
1698 033122  033130                          .WORD   T7DATA                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
1699 033124  000000                          .WORD   0                       ;ADDRESS OF CHARACTERISTICS BLOCK
1700 033126  000010                          .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
1701
1702 033130                          T7DATA:                                 ;CHARACTERISTICS DATA BLOCK
1703 033130  033146                          .WORD   T7BFR                   ;ADDRESS OF MESSAGE BUFFER
1704 033132  000000                          .WORD   0
1705 033134  000016                          .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
1706 033136  000000                          .WORD   0
1707 033140  000000                  T7SP:   .WORD   0                       ;EXTFEA EXTRA WORD
1708
1709 033142  000000  000000                  .WORD   0,0                     ;SPACE
1710 033146                          T7BFR:  .BLKW   8.                      ;MESSAGE BUFFER
1711
1712                                 ;+
1713                                 ;
1714                                 ;TEST DATA FOR SUBTEST TWO
1715                                 ;
1716                                 ;DATA HAS FORMAT:
1717                                 ;
1718                                 ;       1ST WORD        OFFSET TO TEST WORD IN PACKET
1719                                 ;       2ND WORD        BITS TO SET FOR TEST
1720                                 ;
1721                                 ;-
1722
1723 033166                          T72DATA:
1724 033166  000000  037140                  .WORD   0,BIT5!BIT6!BIT9!BIT10!BIT11!BIT12!BIT13
1725 033172  000002  000001                  .WORD   2,BIT0
1726 033176  000004  100100                  .WORD   4,BIT6!BIT15
1727         033202                  T72DONE=.
1728
1729
1730                                 ;+
1731                                 ;LOCAL TEXT MESSAGES FOR TEST
1732                                 ;-
1733
1734 033202     116     102     101  T72NBA: .ASCIZ  'NBA Not Set On Rejected WRITE CHARACTERISTICS'
1735 033260     127     122     111  T7NBA:  .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
```

```
1736 033333     127     122     111  T72REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Non-Zero Unused Fields'
1737 033432     127     122     111  T73REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Data Count'
1738 033525     127     122     111  T74REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Block Address'
1739 033623     127     122     111  T75REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Buffer Length'
1740 033721     103     157     156  T7SSR:  .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
1741 034010     105     170     160  T7NINT: .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
1742 034101     125     156     145  T7INT:  .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
1743 034170     111     156     143  T7TSBA: .ASCIZ  'Incorrect TSBA Address After WRITE CHARACTERISTICS'
1744 034253     127     162     151  TST7ID: .ASCIZ  'Write Characteristics'
1745                                  .EVEN
1746
```

```
1748
1749                                    ;+
1750                                    ;
1751                                    ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
1752                                    ;
1753                                    ;-
1754
1755 034302                            T7REST:
1756 034302                                    SAVREG                          ;SAVE THE REGISTERS
1757 034306  012701  033120                     MOV     #T7PACKET,R1            ;START OF THE PACKET
1758 034312  012721  100004                     MOV     #100004,(R1)+           ;WRITE CHARACTERISTICS WITH ACK
1759 034316  012721  033130                     MOV     #T7DATA,(R1)+           ;ADDRESS OF CHAR DATA BLOCK
1760 034322  005021                             CLR     (R1)+                   ;EXTENDED ADDRESS
1761 034324  012721  000010                     MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
1762 034330  012721  033146                     MOV     #T7BFR,(R1)+            ;ADDRESS OF MESSAGE BUFFER
1763 034334  005021                             CLR     (R1)+
1764 034336  012721  000020                     MOV     #16.,(R1)+              ;LENGTH OF MESSAGE BUFFER
1765 034342  005021                             CLR     (R1)+
1766 034344  005011                             CLR     (R1)
1767 034346  000207                             RTS     PC                      ;RETURN
1768 034350                                     ENDTST
     034350                                                                     L10055:
     034350  104401                                                                     TRAP    C$ETST
1769
```

```
1771                                    .SBTTL  TEST  8: VOLUME CHECK
1772
1773
1774                            ;       THIS TEST VERIFIES THAT THE VOLUME CHECK (VCK) BIT, A FLAG HELD
1775                            ;       WITHIN THE M7196 AND APPEARING IN XSTO, IS SET BY INITIALIZE AND
1776                            ;       CLEARED BY EXECUTING A WRITE CHARACTERISTICS COMMAND WITH THE
1777                            ;       CVC BIT SET.  IT IS ALSO VERIFIED THAT A WRITE CHARACTERISTICS
1778                            ;       COMMAND WITH THE CVC BIT CLEAR DOES NOT AFFECT THE STATE OF THE
1779                            ;       VOLUME CHECK BIT.  THE ACTUAL FUNCTION OF VOLUME CHECK, THAT OF
1780                            ;       PREVENTING OR ALLOWING A TAPE MOTION COMMAND DEPENDING UPON
1781                            ;       WHETHER VOLUME CHECK IS SET OR CLEAR, IS NOT CHECKED BY THIS
1782                            ;       TEST;  THIS FUNCTIONALITY IS CHECKED IN THE INDIVIDUAL TESTS OF
1783                            ;       TAPE MOTION COMMANDS.
1784                            ;
1785                            ;       THE TEST PROCEEDS AS FOLLOWS:
1786                            ;
1787                            ;            1.  THE CONTROLLER IS INITIALIZED BY WRITING INTO THE TSSR.
1788                            ;
1789                            ;            2.  A WRITE CHARACTERISTICS COMMAND IS ISSUED (WITH CVC=0)
1790                            ;                AND XSTO IN THE RETURNED MESSAGE BUFFER IS EXAMINED;
1791                            ;                THE VCK BIT SHOULD BE CLEAR (0).
1792                            ;
1793                            ;            3.  THE PREVIOUS STEP IS REPEATED TO VERIFY THAT VCK DOES
1794                            ;                NOT CHANGE (REMAINS AT 0).
1795                            ;
1796                            ;            4.  A WRITE CHARACTERISTICS COMMAND IS ISSUED WITH CVC=1
1797                            ;                AND THE VCK BIT IN XSTO IN THE MESSAGE BUFFER IS
1798                            ;                EXAMINED;  THE VCK BIT SHOULD BE CLEAR (0).
1799                            ;
1800                            ;            5.  A WRITE CHARACTERISTICS COMMAND IS ISSUED WITH CVC=0
1801                            ;                AND THE VCK BIT IN XSTO IN THE MESSAGE BUFFER IS
1802                            ;                EXAMINED;  THE VCK BIT SHOULD REMAIN CLEAR (0).
1803                            ;
1804
1805 034352                                BGNTST
      034352                                                                    T8::
1810 034352  012700  035237                MOV     #TST8ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
1811 034356  004737  016500                JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
1812 034362  012737  000024  002214        MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
1813 034370                        T8LOOP:
1814
1815 034370  012704  034760                MOV     #T8PACKET,R4            ;PACKET FOR WRITE CHARACTERISTICS
1816 034374  004737  015764        5$:     JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1817 034400  103405                        BCS     10$                     ;BR IF SOFT INIT = OK
1821 034402  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1822 034404                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      034404  104455                                                                    TRAP    C$ERDF
      034406  001441                                                                    .WORD   801
      034410  003646                                                                    .WORD   SFIERR
      034412  012024                                                                    .WORD   SFIMSG
1823 034414  042714  040000        10$:    BIC     #BIT14,(R4)             ;CLEAR THE CVC BIT
1824 034420  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
1825 034424  004737  016326                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1826 034430  103405                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1827 034432  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1831 034434                                ERRDF   ERRNO,T8SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      034434  104455                                                                    TRAP    C$ERDF
```

```
                                                                                .WORD   802
                                                                                .WORD   T8SSR
                                                                                .WORD   PKTSSR
1832 034444                     15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
       034444 104406                                                            TRAP    C$CLP1
1833 034446                             ESCAPE  TST             ;EXIT IF FATAL ERROR
       034446 104410                                                            TRAP    C$ESCAPE
       034450 000604                                                            .WORD   L10064-.
1834 034452 012702 035002               MOV     #T8BFR,R2       ;ADDRESS OF THE MESSAGE BUFFER
1835 034456 032762 000020 000006        BIT     #XSOVCK,XSTO(R2) ;IS VOLUME CHECK SET IN XSTO ?
1836 034464 001406                      BEQ     20$             ;OKAY IF VOLUME CHECK IS CLEAR
1840 034466 016501 000002               MOV     TSSR(R5),R1     ;CONTENTS OF TSSR FOR ERROR REPORT
1841 034472                             ERRHRD  ERRNO,T8NVCK,PKTMES ;VOLUME CHECK NOT CLEAR
       034472 104456                                                            TRAP    C$ERHRD
       034474 001443                                                            .WORD   803
       034476 035057                                                            .WORD   T8NVCK
       034500 012100                                                            .WORD   PKTMES
1842 034502                     20$:    CKLOOP                  ;LOOP ON ERROR ?
       034502 104406                                                            TRAP    C$CLP1
1843 034504 010465 000000               MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS FOR WRITE CHAR
1844 034510 004737 016326               JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
1845 034514 103405                      BCS     25$             ;BR IF CARRY SET (GOOD RETURN)
1846 034516 010001                      MOV     R0,R1           ;SAVE CONTENTS OF TSSR
1850 034520                             ERRDF   ERRNO,T8SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
       034520 104455                                                            TRAP    C$ERDF
       034522 001444                                                            .WORD   804
       034524 035150                                                            .WORD   T8SSR
       034526 012036                                                            .WORD   PKTSSR
1851 034530                     25$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
       034530 104406                                                            TRAP    C$CLP1
1852 034532                             ESCAPE  TST             ;EXIT IF FATAL ERROR
       034532 104410                                                            TRAP    C$ESCAPE
       034534 000520                                                            .WORD   L10064-.
1853 034536 032762 000020 000006        BIT     #XSOVCK,XSTO(R2) ;IS VOLUME CHECK SET IN XSTO ?
'854 034544 001406                      BEQ     30$             ;OKAY IF VOLUME CHECK IS SET
1858 034546 016501 000002               MOV     TSSR(R5),R1     ;CONTENTS OF TSSR FOR ERROR REPORT
1859 034552                             ERRHRD  ERRNO,T8NVCK,PKTMES ;VOLUME CHECK NOT SET
       034552 104456                                                            TRAP    C$ERHRD
       034554 001445                                                            .WORD   805
       034556 035057                                                            .WORD   T8NVCK
       034560 012100                                                            .WORD   PKTMES
1860 034562                     30$:    CKLOOP                  ;LOOP ON ERROR ?
       034562 104406                                                            TRAP    C$CLP1
1861 034564 052714 040000               BIS     #BIT14,(R4)     ;SET THE CVC BIT
1862 034570 010465 000000               MOV     R4,TSDB(R5)     ;SET THE FACKET ADDRESS FOR WRITE CHAR
1863 034574 004737 016326               JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
1864 034600 103405                      BCS     35$             ;BR IF CARRY SET (GOOD RETURN)
1865 034602 010001                      MOV     R0,R1           ;SAVE CONTENTS OF TSSR
1869 034604                             ERRDF   ERRNO,T8SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
       034604 104455                                                            TRAP    C$ERDF
       034606 001446                                                            .WORD   806
       034610 035150                                                            .WORD   T8SSR
       034612 012036                                                            .WORD   PKTSSR
1870 034614                     35$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
       034614 104406                                                            TRAP    C$CLP1
1871 034616                             ESCAPE  TST             ;EXIT IF FATAL ERROR
       034616 104410                                                            TRAP    C$ESCAPE
```

```
                034620  000434                                                          .WORD   L10064-.
1872            034622  032762  000020  000006          BIT     #XSOVCK,XSTO(R2)        ;IS VOLUME CHECK CLEAR IN XSTO ?
1873            034630  001406                          BEQ     40$                     ;OKAY IF VOLUME CHECK IS CLEARED
1877            034632  016501  000002                  MOV     TSSR(R5),R1             ;CONTENTS OF TSSR FOR ERROR REPORT
1878            034636                                  ERRHRD  ERRNO,T8VCK,PKTMES      ;VOLUME CHECK NOT CLEARED
                034636  104456                                                          TRAP    C$ERHRD
                034640  001447                                                          .WORD   807
                034642  035022                                                          .WORD   T8VCK
                034644  012100                                                          .WORD   PKTMES
1879            034646                          40$:    CKLOOP                          ;LOOP ON ERROR ?
                034646  104406                                                          TRAP    C$CLP1
1880            034650  042714  040000                  BIC     #BIT14,(R4)             ;CLEAR THE CVC BIT
1881            034654  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
1882            034660  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1883            034664  103405                          BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
1884            034666  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1888            034670                                  ERRDF   ERRNO,T8SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                034670  104455                                                          TRAP    C$ERDF
                034672  001450                                                          .WORD   808
                034674  035150                                                          .WORD   T8SSR
                034676  012036                                                          .WORD   PKTSSR
1889            034700                          45$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                034700  104406                                                          TRAP    C$CLP1
1890            034702                                  ESCAPE  TST                     ;EXIT IF FATAL ERROR
                034702  104410                                                          TRAP    C$ESCAPE
                034704  000350                                                          .WORD   L10064-.
1891            034706  032762  000020  000006          BIT     #XSOVCK,XSTO(R2)        ;IS VOLUME CHECK CLEAR IN XSTO ?
1892            034714  001406                          BEQ     50$                     ;OKAY IF VOLUME CHECK IS CLEARED
1896            034716  016501  000002                  MOV     TSSR(R5),R1             ;CONTENTS OF TSSR FOR ERROR REPORT
1897            034722                                  ERRHRD  ERRNO,T8VCK,PKTMES      ;VOLUME CHECK NOT CLEARED
                034722  104456                                                          TRAP    C$ERHRD
                034724  001451                                                          .WORD   809
                034726  035022                                                          .WORD   T8VCK
                034730  012100                                                          .WORD   PKTMES
1898            034732                          50$:    CKLOOP                          ;LOOP ON ERROR ?
                034732  104406                                                          TRAP    C$CLP1
1899            034734  004737  016446          60$:    JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
1900            034740  103002                          BCC     62$                     ;BRANCH IF NOT
1901            034742  000137  034370                  JMP     T8LOOP                  ;LOOP UNTIL COUNT EXPIRED
1902            034746                          62$:    EXIT    TST                     ;ALL DONE THIS TEST
                034746  104432                                                          TRAP    C$EXIT
                034750  000304                                                          .WORD   L10064-.
1903
1904                                            ;+
1905                                            ;LOCAL STORAGE FOR THIS TEST
1906                                            ;-
1907
1909                    034760                          .=<.+10>&177770
1911            034760                          T8PACKET:                               ;COMMAND PACKET FOR TEST
1912            034760  100004                          .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND
1913            034762  034770                          .WORD   T8DATA                  ;ADDRESS OF CHARACTERISTICS BLOCK
1914            034764  000000                          .WORD   0
1915            034766  000010                          .WORD   10                      ;STARTING VALUE OF COUNTER
1916
1917            034770                          T8DATA:                                 ;CHARACTERISTICS DATA BLOCK
1918            034770  035002                          .WORD   T8BFR                   ;ADDRESS OF MESSAGE BUFFER
1919            034772  000000                          .WORD   0
```

```
 1920 034774  000020                          .WORD   16.                     ;LENGTH OF MESSAGE BUFFER
 1921 034776  000000  000000                  .WORD   0,0
 1922
 1923 035002                          T8BFR:  .BLKW   8.                      ;MESSAGE BUFFER
 1924
 1925
 1926                                 ;+
 1927                                 ;LOCAL TEXT MESSAGES FOR TEST
 1928                                 ;-
 1929
 1930 035022      126     157     154 T8VCK:  .ASCIZ  'Volume Check Bit Not Cleared'
 1931 035057      126     157     154 T8NVCK: .ASCIZ  'Volume Check Bit (VCK) Not Clear After Initialize (XSTO)'
 1932 035150      103     157     156 T8SSR:  .ASCIZ  'Contents of TSSR Incorrect After Write Characteristics'
 1933 035237      126     157     154 TST8ID: .ASCIZ  'Volume Check'
 1934                                         .EVEN
 1935 035254                                 ENDTST
      035254                                                          L10064:
      035254  104401                                                        TRAP    C$ETST
```

```
                        .SBTTL  TEST  9: COMPLETION INTERRUPT


                ;       THIS TEST VERIFIES THAT AN INTERRUPT IS GENERATED AT THE
                ;       COMPLETION OF THE WRITE CHARACTERISTICS COMMAND IF THE INTERRUPT
                ;       ENABLE (IE) BIT IN THE COMMAND HEADER WORD IS SET.  THIS TEST
                ;       CHECKS THE FUNCTIONING OF THE INTERRUPT LOGIC AND BASIC
                ;       PROCESSING OF THE IE BIT.
                ;
                ;       THE SEQUENCES OF TEST 7 ARE REPEATED, EXCEPT THAT THE INTERRUPT
                ;       SERVICE ROUTINE IS SET UP TO EXPECT INTERRUPTS AND EACH WRITE
                ;       CHARACTERISTICS COMMAND IS ISSUED WITH THE IE BIT SET (1).  IT
                ;       IS VERIFIED, WHERE APPROPRIATE, THAT THE IE STATUS BIT IN XST0
                ;       OF ANY MESSAGE PACKET IS SET AND THAT A COMPLETION INTERRUPT IS
                ;       GENERATED.  FINALLY, A SEQUENCE OF TWO COMMANDS ARE ISSUED, THE
                ;       FIRST WITH IE=1 AND THE SECOND WITH IE=0.  IT IS VERIFIED THAT
                ;       NO INTERRUPT IS GENERATED AFTER THE SECOND COMMAND AND THAT THE
                ;       IE BIT IN XST0 IS 0.
                ;
                BGNTST
                                                                        T9::
035256  005037  002224          CLR     EXTFEA                  ;CLEAR EXTENDED FEATURES SWITCH
035262  012700  040271          MOV     #TST9ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
035266  004737  016500          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
035272  012737  000024  002214  MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
035300                  T9LOOP:
035300                  BGNSUB                                  ;////////////// BEGIN SUBTEST ////////////
                                                                        T9.1:
035300  104402                                                          TRAP    C$BSUB

035302  004737  040316          JSR     PC,T9REST               ;SET PACKET TO INITIAL VALUES
035306                  SETPRI  #PRI00                          ;LOWER PRIORITY TO ALLOW INTERRUPTS
035306  012700  000000                                                  MOV     #PRI00,R0
035312  104441                                                          TRAP    C$SPRI
035314  012703  002762          MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
035320  012704  037220          MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
035324  012764  000010  000006  MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
035332                  5$:
035332                  BGNSEG                                  ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
035332  104404                                                          TRAP    C$BSEG

035334  004737  015764          JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
035340  103405                  BCS     10$                     ;BR IF SOFT INIT = OK
035342  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
035344                  ERRDF   ERRNO,SFIERR,SFIMSG             ;DEVICE FATAL ERROR DURING INIT
035344  104455                                                          TRAP    C$ERDF
035346  001605                                                          .WORD   901
035350  003646                                                          .WORD   SFIERR
035352  012024                                                          .WORD   SFIMSG
035354  005037  002220  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
035360  005037  002222          CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
035364  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
035370  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
035374  103407                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
035376  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
035400                  ERRDF   ERRNO,T9SSR,PKTSSR              ;DEVICE FATAL SSR FAILED TO SET
035400  104455                                                          TRAP    C$ERDF
```

```
                035402  001606                                                          .WORD   902
                035404  037737                                                          .WORD   T9SSR
                035406  012036                                                          .WORD   PKTSSR
1993            035410  005237  002220          INC     FATFLG          ;SET FATAL ERROR FLAG
1994            035414                  15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
                035414  104406                                                          TRAP    C$CLP1
1995            035416                  ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
                035416  104410                                                          TRAP    C$ESCAPE
                035420  000056                                                          .WORD   10000$-.
1996            035422  005737  002222  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1997            035426  001004          BNE     22$                     ;BRANCH IF YES
2001            035430                  ERRHRD  ERRNO,T9NINT,PKTSSR
                035430  104456                                                          TRAP    C$ERHRD
                035432  001607                                                          .WORD   903
                035434  040026                                                          .WORD   T9NINT
                035436  012036                                                          .WORD   PKTSSR
2002            035440  016501  000002  22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
2003            035444  012702  000200          MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
2004            035450  032701  000100          BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
2005            035454  001402                  BEQ     25$             ;BRANCH IF NOT OFF-LINE
2006            035456  052702  000100          BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
2007            035462  020201          25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
2008            035464  001404                  BEQ     30$             ;OKAY IF MATCH
2012            035466                  ERRHRD  ERRNO,T9NBA,PKTSSR      ;NBA NOT ZERO
                035466  104456                                                          TRAP    C$ERHRD
                035470  001610                                                          .WORD   904
                035472  037276                                                          .WORD   T9NBA
                035474  012036                                                          .WORD   PKTSSR
2013            035476                  30$:
2014            035476                  ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                035476                                                  10000$:
                035476  104405                                                          TRAP    C$ESEG
2015
2016            035500  012364  000006          MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
2017            035504  020327  003056          CMP     R3,#TBLEND              ;HAS ALL DATA BEEN TESTED ?
2018            035510  103002                  BHIS    55$                     ;BRANCH IF ALL DATA DONE
2019            035512  000137  035332          JMP     5$                      ;BRANCH TILL BACK TO ZERO
2020
2021            035516                  55$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                035516                                                  L10066:
                035516  104403                                                          TRAP    C$ESUB
2022
2023            035520  005737  002220          TST     FATFLG          ;ANY FATAL ERRORS ?
2024            035524  001402                  BEQ     60$             ;BRANCH IF NOT
2025            035526  004737  017172          JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
2026            035532  033727  037254  000200  60$:    BIT     T98FR+12,#BIT7  ;EXTENDED FEATURES SET?
2027            035540  001402                  BEQ     70$             ;BR IF NO
2028            035542  005237  002224          INC     EXTFEA          ;SET EXT FEATURE FLAG
2029            035546                  70$:
```

```
                                        ;+
                                        ;
                                        ;TEST 9, SUBTEST 2
                                        ;
                                        ;CHECK THAT UNUSED BITS BEING SET CAUSES
                                        ;WRITE CHARACTERISTICS COMMAND TO BE REJECTED
                                        ;
                                        ;-

                                                BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
                                                                                                        T9.2:
035546  104402                                                                                                TRAP    C$BSUB

                                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
035550  012700  000000                                                                                        MOV     #PRI00,R0
035554  104441                                                                                                TRAP    C$SPRI
035556  012703  037262                        MOV     #T92DATA,R3             ;START OF TEST DATA FOR SUBTEST
035562  012704  037220                5$:     MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
035566  004737  040316                        JSR     PC,T9REST               ;RESTORE PACKET TO STARTING VALUES

035572                                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
035572  104404                                                                                                TRAP    C$BSEG

035574  004737  015764                        JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
035600  103405                                BCS     10$                     ;BR IF SOFT INIT = OK
035602  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
035604                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
035604  104455                                                                                                TRAP    C$ERDF
035606  001611                                                                                                .WORD   905
035610  003645                                                                                                .WORD   SFIERR
035612  012024                                                                                                .WORD   SFIMSG
035614  005037  002222                10$:     CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
035620  010400                                MOV     R4,R0                   ;START OF THE COMMAND PACKET
035622  061300                                ADD     (R3),R0                 ;OFFSET TO THE DATA WORD TO TEST
035624  056310  000002                        BIS     2(R3),(R0)              ;SET THE DATA BITS TO BE TESTED
035630  010465  000000                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
035634  004737  016240                        JSR     PC,WAITF                ;WAIT FOR SSR TO SET
035640  103405                                BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
035642  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
035644                                          ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
035644  104455                                                                                                TRAP    C$ERDF
035646  001612                                                                                                .WORD   906
035650  037737                                                                                                .WORD   T9SSR
035652  012036                                                                                                .WORD   PKTSSR
035654                                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
035654  104406                                                                                                TRAP    C$CLP1
035656                                          ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
035656  104410                                                                                                TRAP    C$ESCAPE
035660  000056                                                                                                .WORD   10000$-.
035662  005737  002222                        TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
035666  001004                                BNE     22$                     ;BRANCH IF YES
035670                                          ERRHRD  ERRNO,T9NINT,PKTSSR
035670  104456                                                                                                TRAP    C$ERHRD
035672  001613                                                                                                .WORD   907
035674  040026                                                                                                .WORD   T9NINT
035676  012036                                                                                                .WORD   PKTSSR
035700  016501  000002                22$:     MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
```

```
2077 035704  012702  102206                MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
2078 035710  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
2079 035714  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
2080 035716  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2081 035722  020201                25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2082 035724  001404                        BEQ     30$                     ;OKAY IF MATCH
2086 035726                                ERRHRD  ERRNO,T92REJ,PKTSSR     ;COMMAND NOT REJECTED
     035726  104456                                                               TRAP    C$ERHRD
     035730  001614                                                               .WORD   908
     035732  037351                                                               .WORD   T92REJ
     035734  012036                                                               .WORD   PKTSSR
2087 035736                        30$:
2088 035736                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     035736                                                                     10000$:
     035736  104405                                                               TRAP    C$ESEG
2089
2090 035740  062703  000004                ADD     #4,R3                   ;POINT TO NEXT DATA PAIR
2091 035744  020327  037276                CMP     R3,#T92DONE             ;COMPARE TO END OF TEST DATA
2092 035750  103002                        BHIS    59$                     ;BRANCH IF ALL DATA TESTED
2093 035752  000137  035562                JMP     5$                      ;BRANCH TILL BACK TO ZERO
2094
2095 035756                        59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     035756                                                                     L10067:
     035756  104403                                                               TRAP    C$ESUB
```

```
                                ;+
                                ;
                                ;TEST 9, SUBTEST 3
                                ;
                                ;CHECK THE WRITE CHARACTERISTICS COMMAND IS REJECTED
                                ;IF ISSUED WITH AN INVALID DATA BLOCK BYTE COUNT
                                ;
                                ;-

                                        BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
                                                                                        T9.3:
                                                                                        TRAP    C$BSUB

                                        SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
                                                                                        MOV     #PRI00,R0
                                                                                        TRAP    C$SPRI
                                        MOV     #1,R3                   ;STARTING BYTE COUNT
                                5$:     MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
                                        JSR     PC,T9REST               ;RESTORE PACKET TO STARTING VALUES

                                        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$ERDF
                                                                                        .WORD   909
                                                                                        .WORD   SFIERR
                                                                                        .WORD   SFIMSG
                                10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
                                        MOV     R3,PKBCNT(R4)           ;INSERT THE BYTE COUNT FOR TEST
                                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
                                        JSR     PC,WAITF                ;WAIT FOR SSR TO SET
                                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                                                                        TRAP    C$ERDF
                                                                                        .WORD   910
                                                                                        .WORD   T9SSR
                                                                                        .WORD   PKTSSR
                                15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                        TRAP    C$CLP1
                                        ESCAPE  SFG                     ;BY-PASS SUBTEST IF FATAL ERROR
                                                                                        TRAP    C$ESCAPE
                                                                                        .WORD   10000$-.
                                        TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
                                        BNE     22$                     ;BRANCH IF YES
                                        ERRHRD  ERRNO,T9NINT,PKTSSR
                                                                                        TRAP    C$ERHRD
                                                                                        .WORD   911
                                                                                        .WORD   T9NINT
                                                                                        .WORD   PKTSSR
                                22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
                                        MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
                                        BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
```

```
2143 036122  001402                    BEQ     25$                     ;BRANCH IF NOT OFF-LINE
2144 036124  052702  000100            BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2145 036130  020201              25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2146 036132  001404                    BEQ     30$                     ;OKAY IF MATCH
2150 036134                            ERRHRD  ERRNO,T93REJ,PKTSSR     ;COMMAND NOT REJECTED
     036134  104456                                                            TRAP    C$ERHRD
     036136  001620                                                            .WORD   912
     036140  037450                                                            .WORD   T93REJ
     036142  012036                                                            .WORD   PKTSSR
2151 036144                      30$:
2152 036144                            ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     036144                                                                  10000$:
     036144  104405                                                            TRAP    C$ESEG
2153
2154 036146  005203                    INC     R3                      ;NEXT BYTE COUNT
2155 036150  020327  000006            CMP     R3,#6                   ;TESTED ALL INVALID ?
2156 036154  002002                    BGE     59$                     ;BRANCH IF TEST DONE
2157 036156  000137  035774            JMP     5$                      ;BRANCH TILL BACK TO ZERO
2158
2159 036162                      59$:  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     036162                                                                  L10070:
     036162  104403                                                            TRAP    C$ESUB
```

```
2161                                    ;+
2162                                    ;
2163                                    ;TEST 9, SUBTEST 4
2164                                    ;
2165                                    ;SUBTEST TO VERIFY THAT A WRITE CHARACTERISTICS COMMAND IS
2166                                    ;REJECTED IF AN ILLEGAL DATA BLOCK ADDRESS IS ISSUED.
2167                                    ;
2168                                    ;-
2169
2170 036164                                    BGNSUB                          ;////////// BEGIN SUBTEST ////////////
      036164                                                                                  T9.4:
      036164  104402                                                                            TRAP    C$BSUB
2171
2172 036166                                    SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      036166  012700  000000                                                                   MOV     #PRI00,R0
      036172  104441                                                                           TRAP    C$SPRI
2173 036174  012703  037262                    MOV     #T92DATA,R3             ;START OF TEST DATA FOR SUBTEST
2174 036200  012704  037220            5$:     MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
2175 036204  004737  040316                    JSR     PC,T9REST               ;RESTORE PACKET TO STARTING VALUES
2176
2177
2178 036210  004737  015764                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2179 036214  103405                            BCS     10$                     ;BR IF SOFT INIT = OK
2183 036216  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2184 036220                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      036220  104455                                                                           TRAP    C$ERDF
      036222  001621                                                                           .WORD   913
      036224  003646                                                                           .WORD   SFIERR
      036226  012024                                                                           .WORD   SFIMSG
2185 036230  005037  002222            10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2186 036234  052737  000001  037230            BIS     #1,T9DATA               ;MAKE ADDRESS ODD
2187 036242  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2188 036246  004737  016240                    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
2189 036252  103405                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2190 036254  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2194 036256                                    ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      036256  104455                                                                           TRAP    C$ERDF
      036260  001622                                                                           .WORD   914
      036262  037737                                                                           .WORD   T9SSR
      036264  012036                                                                           .WORD   PKTSSR
2195 036266                            15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      036266  104406                                                                           TRAP    C$CLP1
2196 036270                                    ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
      036270  104410                                                                           TRAP    C$ESCAPE
      036272  000056                                                                           .WORD   L10071-.
2197 036274  005737  002222                    TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2198 036300  001004                            BNE     22$                     ;BRANCH IF YES
2202 036302                                    ERRHRD  ERRNO,T9NINT,PKTSSR
      036302  104456                                                                           TRAP    C$ERHRD
      036304  001623                                                                           .WORD   915
      036306  040026                                                                           .WORD   T9NINT
      036310  012036                                                                           .WORD   PKTSSR
2203 036312  016501  000002            22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
2204 036316  012702  102206                    MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
2205 036322  032701  000100                    BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
2206 036326  001402                            BEQ     25$                     ;BRANCH IF NOT OFF-LINE
2207 036330  052702  000100                    BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
```

```
2208 036334  020201                25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2209 036336  001404                        BEQ     30$                     ;OKAY IF MATCH
2213 036340                              ERRHRD  ERRNO,T94REJ,PKTSSR     ;COMMAND NOT REJECTED
     036340  104456                                                                TRAP    C$ERHRD
     036342  001624                                                              .WORD   916
     036344  037543                                                              .WORD   T94REJ
     036346  012036                                                              .WORD   PKTSSR
2214 036350                          30$:
2215
2216 036350                              ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     036350                                                                   L10071:
     036350  104403                                                              TRAP    C$ESUB
2217
```

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

036352                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
036352                                                                          T9.5:
036352  104402                                                                  TRAP    C$BSUB

036354                          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
036354  012700  000000                                                          MOV     #PRI00,R0
036360  104441                                                                  TRAP    C$SPRI
036362  012703  000001          MOV     #1,R3                   ;STARTING BUFFER LENGTH
036366  012704  037220  5$:     MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
036372  004737  040316          JSR     PC,T9REST               ;RESTORE PACKET TO STARTING VALUES

036376                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
036376  104404                                                                  TRAP    C$BSEG

036400  004737  015764          JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
036404  103405                  BCS     10$                     ;BR IF SOFT INIT = OK
036406  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
036410                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
036410  104455                                                                  TRAP    C$ERDF
036412  001625                                                                  .WORD   917
036414  003646                                                                  .WORD   SFIERR
036416  012024                                                                  .WORD   SFIMSG
036420  005037  002222  10$:     CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
036424  010337  037234          MOV     R3,T9DATA+4             ;INSERT THE BAD MESSAGE LENGTH
036430  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
036434  004737  016240          JSR     PC,WAITF                ;WAIT FOR SSR TO SET
036440  103405                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
036442  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
036444                          ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
036444  104455                                                                  TRAP    C$ERDF
036446  001626                                                                  .WORD   918
036450  037737                                                                  .WORD   T9SSR
036452  012036                                                                  .WORD   PKTSSR
036454  104406  15$:     CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
                                                                                TRAP    C$CLP1
036456                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
036456  104410                                                                  TRAP    C$ESCAPE
036460  000056                                                                  .WORD   10000$-.
036462  005737  002222          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
036466  001004                  BNE     22$                     ;BRANCH IF YES
036470                          ERRHRD  ERRNO,T9NINT,PKTSSR
036470  104456                                                                  TRAP    C$ERHRD
036472  001627                                                                  .WORD   919
036474  040026                                                                  .WORD   T9NINT
036476  012036                                                                  .WORD   PKTSSR
036500  016501  000002  22$:     MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
036504  012702  102206          MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
036510  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
```

```
 2265 036514  001402                    BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 2266 036516  052702  000100            BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 2267 036522  020201              25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 2268 036524  001404                    BEQ     30$                     ;OKAY IF MATCH
 2272 036526                            ERRHRD  ERRNO,T95REJ,PKTSSR     ;COMMAND NOT REJECTED
      036526  104456                                                            TRAP    C$ERHRD
      036530  001630                                                            .WORD   920
      036532  037641                                                            .WORD   T95REJ
      036534  012036                                                            .WORD   PKTSSR
 2273 036536                      30$:
 2274 036536                            ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      036536                                                                  10000$:
      036536  104405                                                            TRAP    C$ESEG
 2275
 2276 036540  005203                    INC     R3                      ;NEXT BUFFER LENGTH
 2277 036542  020327  000014            CMP     R3,#14                  ;HAVE ALL BAD VALUES BEEN TESTED ?
 2278 036546  002002                    BGE     59$                     ;BRANCH IF ALL TESTED
 2279 036550  000137  036366            JMP     5$                      ;BRANCH TILL BACK TO ZERO
 2280
 2281 036554                      59$:  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      036554                                                                  L10072:
      036554  104403                                                            TRAP    C$ESUB
```

```
                                ;+
                                ;
                                ;TEST 9, 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.
                                ;
                                ;-
036556                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
036556                                                                                  .       T9.6:
036556  104402                                                                                  TRAP    C$BSUB
036560  005737  002224                  TST     EXTFEA                  ;IS EXTENDED FEATURES SOFT. SW SET?
036564  001002                          BNE     4$                      ;BR, IF SOFTWARE SWITCH IS SET (ON)
036566  000137  037006                  JMP     55$                     ;EXIT SUBTEST
036572  004737  040316          4$:     JSR     PC,T9REST               ;SET PACKET TO START-UP VALUES
036576                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
036576  012700  000000                                                                  MOV     #PRI00,R0
036602  104441                                                                          TRAP    C$SPRI
036604  012703  002762                  MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
036610  012704  037220                  MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
036614  012764  000012  000006          MOV     #10.,PKBCNT(R4)         ;START WITH EXTENDED FEATURES VALUE
036622                          5$:
036622                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
036622  104404                                                                          TRAP    C$BSEG
036624  004737  015764                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
036630  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
036632  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
036634                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
036634  104455                                                                          TRAP    C$ERDF
036636  001631                                                                          .WORD   921
036640  003646                                                                          .WORD   SFIERR
036642  012024                                                                          .WORD   SFIMSG
036644  005037  002220          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
036650  005037  002222                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
036654  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
036660  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
036664  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
036666  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
036670                                  ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
036670  104455                                                                          TRAP    C$ERDF
036672  001632                                                                          .WORD   922
036674  037737                                                                          .WORD   T9SSR
036676  012036                                                                          .WORD   PKTSSR
036700  005237  002220                  INC     FATFLG                  ;SET FATAL ERROR FLAG
036704                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
036704  104406                                                                          TRAP    C$CLP1
036706                                  ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
036706  104410                                                                          TRAP    C$ESCAPE
036710  000056                                                                          .WORD   10000$-.
036712  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
036716  001004                          BNE     22$                     ;BRANCH IF YES
```

```
2333 036720                                ERRHRD  ERRNO,T9NINT,PKTSSR
     036720  104456                                                      TRAP    C$ERHRD
     036722  001633                                                      .WORD   923
     036724  040026                                                      .WORD   T9NINT
     036726  012036                                                      .WORD   PKTSSR
2334 036730  016501  000002        22$:    MOV     TSSR(R5),R1         ;GET THE CONTENTS OF TSSR
2335 036734  012702  000200              MOV     #SSR,R2             ;EXPECTED CONTENTS OF TSSR
2336 036740  032701  000100              BIT     #OFL,R1             ;IS OFF-LINE BIT SET ?
2337 036744  001402                      BEQ     25$                 ;BRANCH IF NOT OFF-LINE
2338 036746  052702  000100              BIS     #OFL,R2             ;SET OFF-LINE IN EXPECTED DATA
2339 036752  020201              25$:    CMP     R2,R1               ;DOES EXPECTED MATCH RECEIVED ?
2340 036754  001404                      BEQ     30$                 ;OKAY IF MATCH
2344 036756                              ERRHRD  ERRNO,T9NBA,PKTSSR  ;NBA NOT ZERO
     036756  104456                                                      TRAP    C$ERHRD
     036760  001634                                                      .WORD   924
     036762  037276                                                      .WORD   T9NBA
     036764  012036                                                      .WORD   PKTSSR
2345 036766                        30$:
2346 036766                              ENDSEG                      ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     036766                                                                 10000$:
     036766  104405                                                      TRAP    C$ESEG
2347
2348 036770  012364  000006              MOV     (R3)+,PKBCNT(R4)    ;SET THE TEST WORD
2349 036774  020327  003056              CMP     R3,#TBLEND          ;HAS ALL DATA BEEN TESTED ?
2350 037000  103002                      BHIS    55$                 ;BRANCH IF ALL DATA DONE
2351 037002  000137  036622              JMP     5$                  ;BRANCH TILL BACK TO ZERO
2352
2353 037006                        55$:  ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     037006                                                                 L10073:
     037006  104403                                                      TRAP    C$ESUB
```

```
2355
2356                                    ;+
2357                                    ;
2358                                    ;TEST 9, SUBTEST 7
2359                                    ;
2360                                    ;TEST WRITE CHARACTERISTICS WITH/WITHOUT INTERRUPTS ENABLED
2361                                    ;
2362                                    ;-
2363
2364 037010                                 BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
    037010                                                                                      T9.7:
    037010  104402                                                                              TRAP    C$BSUB
2365
2366 037012                                 SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
    037012  012700  000000                                                                      MOV     #PRI00,R0
    037016  104441                                                                              TRAP    C$SPRI
2367 037020  012704  037220                 MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
2368 037024  004737  040316                 JSR     PC,T9REST               ;SET UP A VALID PACKET
2369 037030  004737  015764                 JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2370 037034  103405                        BCS     10$                     ;BR IF SOFT INIT = OK
2374 037036  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2375 037040                                 ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
    037040  104455                                                                              TRAP    C$ERDF
    037042  001635                                                                              .WORD   925
    037044  003646                                                                              .WORD   SFIERR
    037046  012024                                                                              .WORD   SFIMSG
2376 037050  005037  002222            10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2377 037054  052714  000200                    BIS     #BIT7,(R4)              ;ENABLE INTERRUPTS
2378 037060  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2379 037064  004737  016326                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2380 037070  103405                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2381 037072  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2385 037074                                     ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
    037074  104455                                                                              TRAP    C$ERDF
    037076  001636                                                                              .WORD   926
    037100  037737                                                                              .WORD   T9SSR
    037102  012036                                                                              .WORD   PKTSSR
2386 037104                             15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
    037104  104406                                                                              TRAP    C$CLP1
2387 037106                                     ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
    037106  104410                                                                              TRAP    C$ESCAPE
    037110  000102                                                                              .WORD   L10074-.
2388 037112  005737  002222                    TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2389 037116  001004                            BNE     22$                     ;BRANCH IF YES
2393 037120                                     ERRHRD  ERRNO,T9NINT,PKTSSR
    037120  104456                                                                              TRAP    C$ERHRD
    037122  001637                                                                              .WORD   927
    037124  040026                                                                              .WORD   T9NINT
    037126  012036                                                                              .WORD   PKTSSR
2394 037130                             22$:    CKLOOP                          ;LOOP ON ERROR ?
    037130  104406                                                                              TRAP    C$CLP1
2395
2396 037132  005037  002222                    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2397 037136  042714  000200                    BIC     #BIT7,(R4)              ;DISABLE INTERRUPTS
2398 037142  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2399 037146  004737  016326                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2400 037152  103405                            BCS     25$                     ;BR IF CARRY SET (GOOD RETURN)
```

```
2401 037154  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2405 037156                            ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     037156  104455                                                            TRAP    C$ERDF
     037160  001640                                                            .WORD   928
     037162  037737                                                            .WORD   T9SSR
     037164  012036                                                            .WORD   PKTSSR
2406 037166                    25$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     037166  104406                                                            TRAP    C$CLP1
2407 037170                            ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
     037170  104410                                                            TRAP    C$ESCAPE
     037172  000020                                                            .WORD   L10074-.
2408 037174  005737  002222            TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2409 037200  001404                    BEQ     30$                     ;BRANCH IF NOT
2413 037202                            ERRHRD  ERRNO,T9INT,PKTSSR
     037202  104456                                                            TRAP    C$ERHRD
     037204  001641                                                            .WORD   929
     037206  040117                                                            .WORD   T9INT
     037210  012036                                                            .WORD   PKTSSR
2414 037212                    30$:
2415 037212                            ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     037212                                                                  L10074:
     037212  104403                                                            TRAP    C$ESUB
2416
2417 037214                            EXIT    TST                     ;ALL DONE THIS TEST
     037214  104432                                                            TRAP    C$EXIT
     037216  001152                                                            .WORD   L10065-.
2418
2419                           ;+
2420                           ;LOCAL STORAGE FOR THIS TEST
2421                           ;-
2422
2426 037220                    T9PACKET:                               ;COMMAND PACKET FOR TEST
2427 037220  100204                    .WORD   100204                  ;WRITE CHAR COMMAND, WITH IE, ACK
2428 037222  037230                    .WORD   T9DATA                  ;ADDRESS OF CHARACTERISTICS BLOCK
2429 037224  000000                    .WORD   0
2430 037226  000010                    .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
2431
2432 037230                    T9DATA:                                 ;CHARACTERISTICS DATA BLOCK
2433 037230  037242                    .WORD   T9BFR                   ;ADDRESS OF MESSAGE BUFFER
2434 037232  000000                    .WORD   0
2435 037234  000016                    .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
2436 037236  000000  000000            .WORD   0,0
2437
2438 037242                    T9BFR:  .BLKW   8.                      ;MESSAGE BUFFER
2439
2440                           ;+
2441                           ;
2442                           ;TEST DATA FOR SUBTEST TWO
2443                           ;
2444                           ;DATA HAS FORMAT:
2445                           ;
2446                           ;       1ST WORD        OFFSET TO TEST WORD IN PACKET
2447                           ;       2ND WORD        BITS TO SET FOR TEST
2448                           ;
2449                           ;-
2450
2451 037262                    T92DATA:
```

```
2452 037262  000000  037140                    .WORD   0,BIT5!BIT6!BIT9!BIT10!BIT11!BIT12!BIT13
2453 037266  000002  000001                    .WORD   2,BIT0
2454 037272  000004  100100                    .WORD   4,BIT6!BIT15
2455         037276                  T92DONE=.
2456
2457
2458                                 ;+
2459                                 ;LOCAL TEXT MESSAGES FOR TEST
2460                                 ;-
2461
2462 037276      127     122     111  T9NBA:   .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
2463 037351      127     122     111  T92REJ:  .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Non-Zero Unused Fields'
2464 037450      127     122     111  T93REJ:  .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Data Count'
2465 037543      127     122     111  T94REJ:  .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Block Address'
2466 u37641      127     122     111  T95REJ:  .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Buffer Length'
2467 037737      103     157     156  T9SSR:   .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
2468 040026      105     170     160  T9NINT:  .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
2469 040117      125     156     145  T9INT:   .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
2470 040206      111     156     143  T9TSBA:  .ASCIZ  'Incorrect TSBA Address After WRITE CHARACTERISTICS'
2471 040271      103     157     155  TST9ID:  .ASCIZ  'Completion Interrupt'
2472                                              .EVEN
2473
```

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

040316                                  T9REST:
040316                                          SAVREG                  ;SAVE THE REGISTERS
040322  012701  037220                          MOV     #T9PACKET,R1    ;START OF THE PACKET
040326  012721  100204                          MOV     #100204,(R1)+   ;WRITE CHARACTERISTICS WITH ACK, IE
040332  012721  037230                          MOV     #T9DATA,(R1)+   ;ADDRESS OF CHAR DATA BLOCK
040336  005021                                  CLR     (R1)+           ;EXTENDED ADDRESS
040340  012721  000010                          MOV     #8.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
040344  012721  037242                          MOV     #T9BFR,(R1)+    ;ADDRESS OF MESSAGE BUFFER
040350  005021                                  CLR     (R1)+
040352  012721  000016                          MOV     #14.,(R1)+      ;LENGTH OF MESSAGE BUFFER
040356  005021                                  CLR     (R1)+
040360  005011                                  CLR     (R1)
040362  005037  037242                          CLR     T9BFR           ;CLEAR 1ST LOC IN MESSAGE BUFFER
040366  000207                                  RTS     PC              ;RETURN
040370                                          ENDTST
040370                                                                          L10065:
040370  104401                                                                          TRAP    C$ETST
```

```
                                        .SBTTL  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 10 SUBTEST 1
                                ;
                                ;CHECKS THAT THE MESSAGE BUFFER RELEASE COMMAND WORKS
                                ;PROPERLY AND THAT NO INTERRUPT IS GENERATED EVEN
                                ;IF THE "IE" BIT IS SET IN THE COMMAND PACKET
                                ;
                                ;-
040372                                  BGNTST
040372                                                                  T10::
040372  012700  043317                 MOV     #TST10ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
040376  004737  016500                 JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
040402  012737  000024  002214         MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
040410                          T10LOOP:
                                ;
040410                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
040410                                                                          T10.1:
040410  104402                                                                    TRAP    C$BSUB
                                ;
040412  004737  043346                 JSR     PC,T10RST               ;SET PACKET TO INITIAL VALUES
040416                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
040416  012700  000000                                                          MOV     #PRI00,R0
040422  104441                                                                  TRAP    C$SPRI
040424  012704  042510                 MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
040430  012764  000010  000006         MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
040436                          5$:
040436                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
040436  104404                                                                  TRAP    C$BSEG
                                ;
040440  004737  015764                 JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
040444  103405                         BCS     10$                     ;BR IF SOFT INIT = OK
040446  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040450                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
040450  104455                                                                  TRAP    C$ERDF
040452  001751                                                                  .WORD   1001
040454  003646                                                                  .WORD   SFIERR
040456  012024                                                                  .WORD   SFIMSG
040460  005037  002220         10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
040464  005037  002222                 CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
040470  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
040474  004737  016326                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
040500  103407                         BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
040502  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040504                                  ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
040504  104455                                                                  TRAP    C$ERDF
040506  001752                                                                  .WORD   1002
040510  043050                                                                  .WORD   T10SSR
040512  012036                                                                  .WORD   PKTSSR
040514  005237  002220                 INC     FATFLG                  ;SET FATAL ERROR FLAG
040520                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

```
                040520  104406                                                          TRAP    C$CLP1
2551            040522                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
                040522  104410                                                          TRAP    C$ESCAPE
                040524  000056                                                          .WORD   10000$-.
2552            040526  005737  002222          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2553            040532  001004                  BNE     22$                     ;BRANCH IF YES
2557            040534                          ERRHRD  ERRNO,T10NINT,PKTSSR
                040534  104456                                                          TRAP    C$ERHRD
                040536  001753                                                          .WORD   1003
                040540  043137                                                          .WORD   T10NINT
                040542  012036                                                          .WORD   PKTSSR
2558            040544  016501  000002  22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
2559            040550  012702  000200          MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
2560            040554  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
2561            040560  001402                  BEQ     25$                     ;BRANCH IF NOT OFF-LINE
2562            040562  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2563            040566  020201          25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2564            040570  001404                  BEQ     30$                     ;OKAY IF MATCH
2568            040572                          ERRHRD  ERRNO,T10NBA,PKTSSR     ;NBA NOT ZERO
                040572  104456                                                          TRAP    C$ERHRD
                040574  001754                                                          .WORD   1004
                040576  042711                                                          .WORD   T10NBA
                040600  012036                                                          .WORD   PKTSSR
2569            040602          30$:
2570            040602                          ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                040602                                                                  10000$:
                040602  104405                                                          TRAP    C$ESEG
2571            040604                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
                040604  104404                                                          TRAP    C$BSEG
2572
2573            040606  005037  002222          CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2574            040612  012737  025252  042532  MOV     #025252,T10BFR          ;WIPE OUT MESSAGE BUFFER AREA
2575            040620  012714  100212          MOV     #100212,(R4)            ;SET COMMAND PACKET TO MESS BUF REL
2576            040624  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2577            040630  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2578            040634  103407                  BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
2579            040636  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2583            040640                          ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                040640  104455                                                          TRAP    C$ERDF
                040642  001755                                                          .WORD   1005
                040644  043050                                                          .WORD   T10SSR
                040646  012036                                                          .WORD   PKTSSR
2584            040650  005237  002220          INC     FATFLG                  ;SET FATAL ERROR FLAG
2585            040654          45$:    CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
                040654  104406                                                          TRAP    C$CLP1
2586            040656  005737  002222          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2587            040662  001404                  BEQ     52$                     ;BRANCH IF NO
2591            040664                          ERRHRD  ERRNO,T10INT,PKTSSR
                040664  104456                                                          TRAP    C$ERHRD
                040666  001756                                                          .WORD   1006
                040670  043230                                                          .WORD   T10INT
                040672  012036                                                          .WORD   PKTSSR
2592            040674  016501  000002  52$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
2593            040700  012702  000200          MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
2594            040704  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
2595            040710  001402                  BEQ     55$                     ;BRANCH IF NOT OFF-LINE
2596            040712  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
```

```
2597 040716  020201                55$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2598 040720  001404                        BEQ     60$                     ;OKAY IF MATCH
2602 040722                               ERRHRD  ERRNO,T10NNBA,PKTSSR    ;NBA  SET
     040722  104456                                                              TRAP    C$ERHRD
     040724  001757                                                              .WORD   1007
     040726  042773                                                              .WORD   T10NNBA
     040730  012036                                                              .WORD   PKTSSR
2603 040732                        60$:
2604 040732  013701  042532              MOV     T10BFR,R1               ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
2605 040736  012702  025252              MOV     #025252,R2              ;SET UP EXPECTED DATA
2606 040742  020102                      CMP     R1,R2                   ;WAS ANY MESSAGE REC'D
2607 040744  001404                      BEQ     70$                     ;BR, IF OK (EQUAL)
2611 040746                              ERRHRD  ERRNO,T10MBF,EXPREC     ;MESSAGE BUFFER WAS MODIFIED
     040746  104456                                                              TRAP    C$ERHRD
     040750  001760                                                              .WORD   1008
     040752  042614                                                              .WORD   T10MBF
     040754  015464                                                              .WORD   EXPREC
2612 040756                        70$:
2613 040756  005737  002220              TST     FATFLG                  ;ANY FATAL ERRORS
2614 040762  001403                      BEQ     80$                     ;BR, IF NO FATAL ERRORS
2615 040764  004737  017172              JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2616 040770                              ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     040770                                                                  10001$:
     040770  104405                                                              TRAP    C$ESEG
2617 040772                        80$:
2618 040772                              ENDSUB                          ;////////////// END SUBTEST ////////////
     040772                                                                  L10076:
     040772  104403                                                              TRAP    C$ESUB
2619
```

```
                                        ;+
                                        ;TEST 10 SUBTEST 2
                                        ;
                                        ;CHECKS THAT THE MESSAGE BUFFER RELEASE COMMAND WORKS
                                        ;PROPERLY AND THAT THERE IS AN INTERRUPT IF THE "IE"
                                        ;BIT IS SET IN THE COMMAND PACKET AND THE "ERI" BIT
                                        ;IS SET IN THE CHARACTERISTICS DATA PACKET
                                        ;
                                        ;-
040774                                          BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
040774                                                                                      T10.2:
040774  104402                                                                              TRAP    C$BSUB

040776  004737  043346                          JSR     PC,T10RST               ;SET PACKET TO INITIAL VALUES
041002                                          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
041002  012700  000000                                                                      MOV     #PRI00,R0
041006  104441                                                                              TRAP    C$SPRI
041010  012704  042510                          MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
041014  012764  000010  000006                  MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
041022                                  5$:
041022                                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
041022  104404                                                                              TRAP    C$BSEG

041024  004737  015764                          JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
041030  103405                                  BCS     10$                     ;BR IF SOFT INIT = OK
041032  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
041034                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
041034  104455                                                                              TRAP    C$ERDF
041036  001761                                                                              .WORD   1009
041040  003646                                                                              .WORD   SFIERR
041042  012024                                                                              .WORD   SFIMSG
041044  005037  002220          10$:            CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
041050  005037  002222                          CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
041054  012737  000020  042526                  MOV     #000020,T10DATA+6       ;SET ERI IN CHARACTERISTICS DATA
041062  010465  000000                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
041066  004737  016326                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
041072  103407                                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
041074  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
041076                                          ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
041076  104455                                                                              TRAP    C$ERDF
041100  001762                                                                              .WORD   1010
041102  043050                                                                              .WORD   T10SSR
041104  012036                                                                              .WORD   PKTSSR
041106  005237  002220                          INC     FATFLG                  ;SET FATAL ERROR FLAG
041112                          15$:            CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
041112  104406                                                                              TRAP    C$CLP1
041114                                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
041114  104410                                                                              TRAP    C$ESCAPE
041116  000056                                                                              .WORD   10000$-.
041120  005737  002222                          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
041124  001004                                  BNE     22$                     ;BRANCH IF YES
041126                                          ERRHRD  ERRNO,T10NINT,PKTSSR
041126  104456                                                                              TRAP    C$ERHRD
041130  001763                                                                              .WORD   1011
041132  043137                                                                              .WORD   T10NINT
041134  012036                                                                              .WORD   PKTSSR
041136  016501  000002          22$:            MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
```

```
 2667 041142  012702  000200                MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
 2668 041146  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 2669 041152  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 2670 041154  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 2671 041160  020201                25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 2672 041162  001404                        BEQ     30$                     ;OKAY IF MATCH
 2676 041164                                ERRHRD  ERRNO,T10NBA,PKTSSR     ;NBA NOT ZERO
      041164  104456                                                            TRAP    C$ERHRD
      041166  001764                                                            .WORD   1012
      041170  042711                                                            .WORD   T10NBA
      041172  012036                                                            .WORD   PKTSSR
 2677 041174                        30$:
 2678 041174                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      041174                                                                    10000$:
      041174  104405                                                            TRAP    C$ESEG
 2679 041176                                BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      041176  104404                                                            TRAP    C$BSEG
 2680
 2681 041200  005037  002222                CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 2682 041204  012737  025252  042532        MOV     #025252,T10BFR          ;WIPE OUT MESSAGE BUFFER AREA
 2683 041212  012714  100212                MOV     #100212,(R4)            ;SET COMMAND PACKET TO MESS BUF REL
 2684 041216  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 2685 041222  004737  016326                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 2686 041226  103407                        BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
 2687 041230  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2691 041232                                ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      041232  104455                                                            TRAP    C$ERDF
      041234  001765                                                            .WORD   1013
      041236  043050                                                            .WORD   T10SSR
      041240  012036                                                            .WORD   PKTSSR
 2692 041242  005237  002220                INC     FATFLG                  ;SET FATAL ERROR FLAG
 2693 041246                        45$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      041246  104406                                                            TRAP    C$CLP1
 2694 041250  005737  002222                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 2695 041254  001004                        BNE     52$                     ;BRANCH IF YES
 2699 041256                                ERRHRD  ERRNO,T10INT,PKTSSR
      041256  104456                                                            TRAP    C$ERHRD
      041260  001766                                                            .WORD   1014
      041262  043230                                                            .WORD   T10INT
      041264  012036                                                            .WORD   PKTSSR
 2700 041266  016501  000002        52$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 2701 041272  012702  000200                MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
 2702 041276  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 2703 041302  001402                        BEQ     55$                     ;BRANCH IF NOT OFF-LINE
 2704 041304  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 2705 041310  020201                55$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 2706 041312  001404                        BEQ     60$                     ;OKAY IF MATCH
 2710 041314                                ERRHRD  ERRNO,T10NNBA,PKTSSR    ;NBA NOT SET
      041314  104456                                                            TRAP    C$ERHRD
      041316  001767                                                            .WORD   1015
      041320  042773                                                            .WORD   T10NNBA
      041322  012036                                                            .WORD   PKTSSR
 2711 041324                        60$:
 2712 041324  013701  042532                MOV     T10BFR,R1               ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
 2713 041330  012702  025252                MOV     #025252,R2              ;SET UP EXPECTED DATA
 2714 041334  020102                        CMP     R1,R2                   ;WAS ANY MESSAGE REC'D
 2715 041336  001404                        BEQ     70$                     ;BR, IF OK (EQUAL)
```

```
2719 041340                                    ERRHRD  ERRNO,T10MBF,EXPREC     ;MESSAGE BUFFER WAS MODIFIED
     041340  104456                                                                TRAP    C$ERHRD
     041342  001770                                                                .WORD   1016
     041344  042614                                                                .WORD   T10MBF
     041346  015464                                                                .WORD   EXPREC
2720
2721 041350                                70$:
2722 041350  005737  002220                    TST     FATFLG                  ;ANY FATAL ERRORS
2723 041354  001402                            BEQ     80$                     ;BR, IF NO FATAL ERRORS
2724 041356  004737  017172                    JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2725 041362                                80$:
2726 041362                                    ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     041362                                                                       10001$:
     041362  104405                                                                TRAP    C$ESEG
2727 041364                                    ENDSUB                          ;/////////// END SUBTEST ///////////
     041364                                                                       L10077:
     041364  104403                                                                TRAP    C$ESUB
```

```
                                        ;+
                                        ;TEST 10 SUBTEST 3
                                        ;
                                        ;CHECKS THAT THE CPU GIVES UP OWNERSHIP OF THE MESSAGE BUFFER
                                        ;AFTER THE MESSAGE BUFFER RELEASE, AND THAT FOLLOWING COMMANDS
                                        ;WORK CORRECTLY
                                        ;
                                        ;-
2737 041366                                     BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     041366                                                                                       T10.3:
     041366  104402                                                                                     TRAP    C$BSUB
2738
2739 041370  004737  043346                     JSR     PC,T10RST               ;SET PACKET TO INITIAL VALUES
2740 041374                                     SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     041374  012700  000000                                                                          MOV     #PRI00,R0
     041400  104441                                                                                  TRAP    C$SPRI
2741 041402  012704  042510                     MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
2742 041406  012764  000010  000006             MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
2743 041414                             5$:
2744 041414                                     BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     041414  104404                                                                                  TRAP    C$BSEG
2745
2746 041416  004737  015764                     JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2747 041422  103405                             BCS     10$                     ;BR IF SOFT INIT = OK
2751 041424  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2752 041426                                     ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     041426  104455                                                                                  TRAP    C$ERDF
     041430  001771                                                                                  .WORD   1017
     041432  003646                                                                                  .WORD   SFIERR
     041434  012024                                                                                  .WORD   SFIMSG
2753 041436  005037  002220             10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
2754 041442  005037  002222                     CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2755 041446  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2756 041452  004737  016326                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2757 041456  103407                             BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2758 041460  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2762 041462                                     ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     041462  104455                                                                                  TRAP    C$ERDF
     041464  001772                                                                                  .WORD   1018
     041466  043050                                                                                  .WORD   T10SSR
     041470  012036                                                                                  .WORD   PKTSSR
2763 041472  005237  002220                     INC     FATFLG                  ;SET FATAL ERROR FLAG
2764 041476                             15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041476  104406                                                                                  TRAP    C$CLP1
2765 041500                                     ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     041500  104410                                                                                  TRAP    C$ESCAPE
     041502  000056                                                                                  .WORD   10000$-.
2766 041504  005737  002222                     TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2767 041510  001004                             BNE     22$                     ;BRANCH IF YES
2771 041512                                     ERRHRD  ERRNO,T10NINT,PKTSSR
     041512  104456                                                                                  TRAP    C$ERHRD
     041514  001773                                                                                  .WORD   1019
     041516  043137                                                                                  .WORD   T10NINT
     041520  012036                                                                                  .WORD   PKTSSR
2772 041522  016501  000002             22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
2773 041526  012702  000200                     MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
2774 041532  032701  000100                     BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
```

```
 2775 041536  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 2776 041540  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 2777 041544  020201                 25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 2778 041546  001404                        BEQ     30$                     ;OKAY IF MATCH
 2782 041550                                 ERRHRD  ERRNO,T10NBA,PKTSSR     ;NBA NOT ZERO
      041550  104456                                                                TRAP    C$ERHRD
      041552  001774                                                                .WORD   1020
      041554  042711                                                                .WORD   T10NBA
      041556  012036                                                                .WORD   PKTSSR
 2783 041560                         30$:
 2784 041560                                 ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      041560                                                                         10000$:
      041560  104405                                                                TRAP    C$ESEG
 2785 041562                                 BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
      041562  104404                                                                TRAP    C$BSEG
 2786 041564  004737  043346                JSR     PC,T10RST               ;RESET PACKETS AND COMMANDS
 2787 041570  005037  002222                CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 2788 041574  012737  025252  042532        MOV     #025252,T10BFR          ;WIPE OUT MESSAGE BUFFER AREA
 2789 041602  012714  100212                MOV     #100212,(R4)            ;SET COMMAND PACKET TO MESS BUF REL
 2790 041606  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 2791 041612  004737  016326                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 2792 041616  103407                        BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
 2793 041620  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2797 041622                                 ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      041622  104455                                                                TRAP    C$ERDF
      041624  001775                                                                .WORD   1021
      041626  043050                                                                .WORD   T10SSR
      041630  012036                                                                .WORD   PKTSSR
 2798 041632  005237  002220                INC     FATFLG                  ;SET FATAL ERROR FLAG
 2799 041636                         45$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      041636  104406                                                                TRAP    C$CLP1
 2800 041640  005737  002222                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 2801 041644  001404                        BEQ     52$                     ;BRANCH IF NO
 2805 041646                                 ERRHRD  ERRNO,T10INT,PKTSSR
      041646  104456                                                                TRAP    C$ERHRD
      041650  001776                                                                .WORD   1022
      041652  043230                                                                .WORD   T10INT
      041654  012036                                                                .WORD   PKTSSR
 2806 041656  016501  000002         52$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 2807 041662  012702  000200                MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
 2808 041666  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 2809 041672  001402                        BEQ     55$                     ;BRANCH IF NOT OFF-LINE
 2810 041674  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 2811 041700  020201                 55$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 2812 041702  001404                        BEQ     60$                     ;OKAY IF MATCH
 2816 041704                                 ERRHRD  ERRNO,T10NNBA,PKTSSR    ;NBA NOT SET
      041704  104456                                                                TRAP    C$ERHRD
      041706  001777                                                                .WORD   1023
      041710  042773                                                                .WORD   T10NNBA
      041712  012036                                                                .WORD   PKTSSR
 2817 041714                         60$:
 2818 041714  013701  042532                MOV     T10BFR,R1               ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
 2819 041720  012702  025252                MOV     #025252,R2              ;SET UP EXPECTED DATA
 2820 041724  020102                        CMP     R1,R2                   ;WAS ANY MESSAGE REC'D
 2821 041726  001404                        BEQ     70$                     ;BR, IF OK (EQUAL)
 2825 041730                                 ERRHRD  ERRNO,T10MBF,EXPREC     ;MESSAGE BUFFER WAS MODIFIED
      041730  104456                                                                TRAP    C$ERHRD
```

```
                041732  002000                                                          .WORD   1024
                041734  042614                                                          .WORD   T10MBF
                041736  015464                                                          .WORD   EXPREC
     2826
     2827 041740                        70$:    CKLOOP                  ;LOOP ON ERROR IF FLAG SET
                041740  104406                                                          TRAP    C$CLP1
     2828
     2829 041742  005037  002222                CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
     2830 041746  004737  043346                JSR     PC,T10RST       ;RESET THE PACKETS AND COMMANDS
     2831 041752  042714  100000                BIC     #100000,(R4)    ;CLEAR THE ACK BIT
     2832 041756  010465  000000                MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
     2833 041762  004737  016326                JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
     2834 041766  103407                        BCS     75$             ;BR IF CARRY SET (GOOD RETURN)
     2835 041770  010001                        MOV     R0,R1           ;SAVE CONTENTS OF TSSR
     2839 041772                                ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                041772  104455                                                          TRAP    C$ERDF
                041774  002001                                                          .WORD   1025
                041776  043050                                                          .WORD   T10SSR
                042000  012036                                                          .WORD   PKTSSR
     2840 042002  005237  002220                INC     FATFLG          ;SET FATAL ERROR FLAG
     2841 042006                        75$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
                042006  104406                                                          TRAP    C$CLP1
     2842 042010                                ESCAPE  SEG             ;BY-PASS SUBTEST IF FATAL ERROR
                042010  104410                                                          TRAP    C$ESCAPE
                042012  000062                                                          .WORD   10001$-.
     2843 042014  005737  002222                TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
     2844 042020  001006                        BNE     82$             ;BRANCH IF YES
     2848 042022  016500  000002                MOV     TSSR(R5),R0     ;GET TSSR FOR ERROR REPORT
     2849 042026                                ERRHRD  ERRNO,T10NINT,PKTSSR
                042026  104456                                                          TRAP    C$ERHRD
                042030  002002                                                          .WORD   1026
                042032  043137                                                          .WORD   T10NINT
                042034  012036                                                          .WORD   PKTSSR
     2850 042036  016501  000002        82$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
     2851 042042  012702  000200                MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
     2852 042046  032701  000100                BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
     2853 042052  001402                        BEQ     85$             ;BRANCH IF NOT OFF-LINE
     2854 042054  052702  000100                BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
     2855 042060  020201        85$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
     2856 042062  001404                        BEQ     90$             ;OKAY IF MATCH
     2860 042064                                ERRHRD  ERRNO,T10SSR,PKTSSR     ;NBA NOT ZERO
                042064  104456                                                          TRAP    C$ERHRD
                042066  002003                                                          .WORD   1027
                042070  043050                                                          .WORD   T10SSR
                042072  012036                                                          .WORD   PKTSSR
     2861 042074                        90$:
     2862 042074                                ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                042074                                                                  10001$:
                042074  104405                                                          TRAP    C$ESEG
     2863 042076  005737  002220                TST     FATFLG          ;ANY FATAL ERRORS
     2864 042102  001403                        BEQ     95$             ;BR, IF NO FATAL ERRORS
     2865 042104  004737  017172                JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
     2866
     2867 042110                                BGNSEG                  ;<<<<<<<<<<<<< BGN SEGMENT <<<<<<<<<<<<<
                042110  104404                                                          TRAP    C$BSEG
     2868 042112  005037  002222        95$:    CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
     2869 042116  004737  043346                JSR     PC,T10RST       ;RESET THE PACKETS AND COMMANDS
```

```
 2870 042122  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 2871 042126  004737  016326                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 2872 042132  103407                        BCS     100$                    ;BR IF CARRY SET (GOOD RETURN)
 2873 042134  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2877 042136                                ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      042136  104455                                                                TRAP    C$ERDF
      042140  002004                                                               .WORD   1028
      042142  043050                                                               .WORD   T10SSR
      042144  012036                                                               .WORD   PKTSSR
 2878 042146  005237  002220                INC     FATFLG                  ;SET FATAL ERROR FLAG
 2879 042152                        100$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      042152  104406                                                               TRAP    C$CLP1
 2880 042154                                ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
      042154  104410                                                               TRAP    C$ESCAPE
      042156  000062                                                               .WORD   10002$-.
 2881 042160  005737  002222                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 2882 042164  001006                        BNE     112$                    ;BRANCH IF YES
 2886 042166  016500  000002                MOV     TSSR(R5),R0             ;GET TSSR FOR ERROR REPORT
 2887 042172                                ERRHRD  ERRNO,T10NINT,PKTSSR
      042172  104456                                                               TRAP    C$ERHRD
      042174  002005                                                               .WORD   1029
      042176  043137                                                               .WORD   T10NINT
      042200  012036                                                               .WORD   PKTSSR
 2888 042202  016501  000002        112$:   MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 2889 042206  012702  000200                MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
 2890 042212  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 2891 042216  001402                        BEQ     125$                    ;BRANCH IF NOT OFF-LINE
 2892 042220  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 2893 042224  020201        125$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 2894 042226  001405                        BEQ     130$                    ;OKAY IF MATCH
 2898 042230                                ERRHRD  ERRNO,T10SSR,PKTSSR     ;NBA NOT ZERO
      042230  104456                                                               TRAP    C$ERHRD
      042232  002006                                                               .WORD   1030
      042234  043050                                                               .WORD   T10SSR
      042236  012036                                                               .WORD   PKTSSR
 2899 042240                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      042240                                                                    10002$:
      042240  104405                                                               TRAP    C$ESEG
 2900 042242  005737  002220        130$:   TST     FATFLG                  ;CHECK FATAL FLAG FOR SET
 2901 042246  001402                        BEQ     140$                    ;BR, IF NOT SET (OK)
 2902 042250  004737  017172                JSR     PC,CKDROP               ;DROP DEVICE IF INDICATED
 2903 042254                        140$:
 2904 042254                                ENDSUB                          ;////////////// END SUBTEST ////////////
      042254                                                                    L10100:
      042254  104403                                                               TRAP    C$ESUB
 2905 042256                        30$:
```

```
                                        ;+
                                        ;TEST 10 SUBTEST 4
                                        ;
                                        ;CHECKS THAT THE REGISTER MODIFICATION REFUSED (RMR) BIT IN
                                        ;THE TSSR WILL BE SET IF A WRITE CHARACTERISTICS COMMAND
                                        ;BEING EXECUTED AND ANOTHER 'WC" COMMAND IS ATTEMPTED
                                        ;
                                        ;-

042256                                          BGNSUB                  ;////////////// BEGIN SUBTEST //////////////
042256                                                                                          T10.4:
042256  104402                                                                                    TRAP    C$BSUB

042260  004737  043420                          JSR     PC,T10RT2       ;SET SECOND PACKET UP
042264  004737  043346                          JSR     PC,T10RST       ;SET PACKET TO INITIAL VALUES
042270                                          SETPRI  #PRI00          ;LOWER PRIORITY TO ALLOW INTERRUPTS
042270  012700  000000                                                                          MOV     #PRI00,R0
042274  104441                                                                                  TRAP    C$SPRI
042276  012704  042510                          MOV     #T10PACKET,R4   ;GET THE ADDRESS OF COMMAND PACKET
042302  012703  042552                          MOV     #T10PKT,R3      ;GET THE ADDRESS OF 2ND CMD PACKET
042306  012764  000010  000006                  MOV     #8.,PKBCNT(R4)  ;START WITH MINIMUM ALLOWABLE VALUE
042314  012763  000010  000006                  MOV     #8.,PKBCNT(R3)  ;START WITH MINIMUM ALLOWABLE VALUE
042322                                  5$:
042322                                          BGNSEG                  ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
042322  104404                                                                                  TRAP    C$BSEG
042324  004737  015764                          JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
042330  103405                                  BCS     10$             ;BR IF SOFT INIT = OK
042332  010001                                  MOV     R0,R1           ;SAVE CONTENTS OF TSSR
042334                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
042334  104455                                                                                  TRAP    C$ERDF
042336  002007                                                                                  .WORD   1031
042340  003646                                                                                  .WORD   SFIERR
042342  012024                                                                                  .WORD   SFIMSG
042344  005037  002220                  10$:    CLR     FATFLG          ;CLEAR FATAL ERROR FLAG
042350  005037  002222                          CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
042354  010465  000000                          MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
042360  010365  000000                          MOV     R3,TSDB(R5)     ;SECOND COMMAND PACKET
042364  004737  016240                          JSR     PC,WAITF        ;WAIT FOR SSR TO SET
042370  016501  000002                          MOV     TSSR(R5),R1     ;GET CONTENTS OF TSSR REGISTER
042374  032701  000200                          BIT     #SSR,R1         ;CHECK FOR SSR (TSSR) SET
042400  001006                                  BNE     15$             ;BR, IF SSR SET (GOOD)
042402                                          ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
042402  104455                                                                                  TRAP    C$ERDF
042404  002010                                                                                  .WORD   1032
042406  043050                                                                                  .WORD   T10SSR
042410  012036                                                                                  .WORD   PKTSSR
042412  005237  002220                          INC     FATFLG          ;SET FATAL ERROR FLAG
042416                                  15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
042416  104406                                                                                  TRAP    C$CLP1
042420                                          ESCAPE  SEG             ;BY-PASS SUBTEST IF FATAL ERROR
042420  104410                                                                                  TRAP    C$ESCAPE
042422  000056                                                                                  .WORD   10000$-.
042424  005737  002222                          TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
042430  001004                                  BNE     22$             ;BRANCH IF YES

042432                                          ERRHRD  ERRNO,T10NINT,PKTSSR
```

```
        042432  104456                                                          TRAP    C$ERHRD
        042434  002011                                                          .WORD   1033
        042436  043137                                                          .WORD   T10NINT
        042440  012036                                                          .WORD   PKTSSR
  2957  042442  016501  000002        22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
  2958  042446  012702  110200                MOV     #SSR!RMR!SC,R2          ;EXPECTED CONTENTS OF TSSR
  2959  042452  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
  2960  042456  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
  2961  042460  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
  2962  042464  020201                25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
  2963  042466  001404                        BEQ     30$                     ;OKAY IF MATCH
  2967  042470                                ERRHRD  ERRNO,T10SSR,PKTSSR     ;NBA NOT ZERO
        042470  104456                                                          TRAP    C$ERHRD
        042472  002012                                                          .WORD   1034
        042474  043050                                                          .WORD   T10SSR
        042476  012036                                                          .WORD   PKTSSR
  2968  042500                        30$:
  2969  042500                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        042500                                                                  10000$:
        042500  104405                                                          TRAP    C$ESEG
  2970  042502                                ENDSUB                          ;////////////// END SUBTEST //////////////
        042502                                                                  L10101:
        042502  104403                                                          TRAP    C$ESUB
  2971  042504                                EXIT    TST                     ;ALL DONE WITH THIS TEST
        042504  104432                                                          TRAP    C$EXIT
        042506  000764                                                          .WORD   L10075-.
```

```
                                        ;+
                                        ;LOCAL STORAGE FOR THIS TEST
                                        ;-

042510                                  T10PACKET:                      ;COMMAND PACKET FOR TEST
042510  100204                                  .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
042512  042520                                  .WORD   T10DATA         ;ADDRESS OF CHARACTERISTICS BLOCK
042514  000000                                  .WORD   0
042516  000010                                  .WORD   8.              ;STARTING VALUE OF BLOCK SIZE

042520                                  T10DATA:                                ;CHARACTERISTICS DATA BLOCK
042520  042532                                  .WORD   T10BFR          ;ADDRESS OF MESSAGE BUFFER
042522  000000                                  .WORD   0
042524  000016                                  .WORD   14.             ;LENGTH OF MESSAGE BUFFER
042526  000000  000000                          .WORD   0,0

042532                                  T10BFR: .BLKW   8.              ;MESSAGE BUFFER

                                        ;+
                                        ;
                                        ;TEST DATA FOR SUBTEST FOUR
                                        ;
042552                                  T10PKT:                         ;COMMAND PACKET FOR TEST
042552  100204                                  .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
042554  042562                                  .WORD   T10DTA          ;ADDRESS OF CHARACTERISTICS BLOCK
042556  000000                                  .WORD   0
042560  000010                                  .WORD   8.              ;STARTING VALUE OF BLOCK SIZE

042562                                  T10DTA:                         ;CHARACTERISTICS DATA BLOCK
042562  042574                                  .WORD   T10BUFR         ;ADDRESS OF MESSAGE BUFFER
042564  000000                                  .WORD   0
042566  000016                                  .WORD   14.             ;LENGTH OF MESSAGE BUFFER
042570  000000  000000                          .WORD   0,0

042574                                  T10BUFR:.BLKW   8.              ;MESSAGE BUFFER

                                        ;+
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-


042614      115     145     163 T10MBF: .ASCIZ  'Message Buffer Modified after MESSAGE BUFFER RELEASE Command'
042711      116     102     101 T10NBA: .ASCIZ  'NBA Not Clear After WRITE CHARACTERISTICS Command'
042773      116     102     101 T10NNBA:        .ASCIZ  'NBA Set After MESSAGE BUFFER RELEASE Command'
043050      103     157     156 T10SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
043137      105     170     160 T10NINT:        .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
043230      125     156     145 T10INT: .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
043317      102     141     163 TST10ID:        .ASCIZ  'Basic Packet Protocol'
                                            .EVEN
```

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

043346                                  T10RST:
043346                                          SAVREG                          ;SAVE THE REGISTERS
043352  012701  042510                          MOV     #T10PACKET,R1           ;START OF THE PACKET
043356  012721  100204                          MOV     #100204,(R1)+           ;WRITE CHARACTERISTICS WITH ACK, IE
043362  012721  042520                          MOV     #T10DATA,(R1)+          ;ADDRESS OF CHAR DATA BLOCK
043366  005021                                  CLR     (R1)+                   ;EXTENDED ADDRESS
043370  012721  000010                          MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
043374  012721  042532                          MOV     #T10BFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
043400  005021                                  CLR     (R1)+
043402  012721  000016                          MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
043406  005021                                  CLR     (R1)+
043410  005011                                  CLR     (R1)
043412  005037  042532                          CLR     T10BFR                  ;CLEAR 1ST LOC IN MESSAGE BUFFER
043416  000207                                  RTS     PC                      ;RETURN
                                        ;+
                                        ;
                                        ;ROUTINE TO RESTORE COMMAND PACKET #2 TO START-UP (DEFAULT) VALUES
                                        ;
                                        ;-

043420                                  T10RT2:
043420                                          SAVREG                          ;SAVE THE REGISTERS
043424  012701  042552                          MOV     #T10PKT,R1              ;START OF THE PACKET
043430  012721  100204                          MOV     #100204,(R1)+           ;WRITE CHARACTERISTICS WITH ACK, IE
043434  012721  042562                          MOV     #T10DTA,(R1)+           ;ADDRESS OF CHAR DATA BLOCK
043440  005021                                  CLR     (R1)+                   ;EXTENDED ADDRESS
043442  012721  000010                          MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
043446  012721  042574                          MOV     #T10BUFR,(R1)+          ;ADDRESS OF MESSAGE BUFFER
043452  005021                                  CLR     (R1)+
043454  012721  000016                          MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
043460  005021                                  CLR     (R1)+
043462  005011                                  CLR     (R1)
043464  005037  042574                          CLR     T10BUFR                 ;CLEAR 1ST LOC IN MESSAGE BUFFER
043470  000207                                  RTS     PC                      ;RETURN
043472                                          ENDTST
043472                                                                  L10075:
043472  104401                                                                  TRAP    C$ETST
```

```
                                        .SBTTL  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.
                                ;
                                ;-

043474                                  BGNTST
043474                                                                  T11::
043474  012700  045472                  MOV     #TST11ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
043500  004737  016500                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
043504  012737  000024  002214          MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
043512                          T11LOOP:
043512                                  BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
043512                                                                          T11.1:
043512  104402                                                                  TRAP    C$BSUB

043514                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
043514  012700  000000                                                          MOV     #PRI00,R0
043520  104441                                                                  TRAP    C$SPRI
043522  004737  015764                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
043526  103405                          BCS     3$                      ;BR IF SOFT INIT = OK
043530  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043532                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
043532  104455                                                                  TRAP    C$ERDF
043534  002115                                                                  .WORD   1101
043536  003646                                                                  .WORD   SFIERR
043540  012024                                                                  .WORD   SFIMSG
043542                          3$:
043542  012704  044720                  MOV     #T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
043546  004737  010652                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
043552  103404                          BCS     4$                      ;BR, IF COMMAND ISSUED OK
043554                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
043554  104456                                                                  TRAP    C$ERHRD
043556  002116                                                                  .WORD   1102
043560  005052                                                                  .WORD   WRTMSG
043562  012024                                                                  .WORD   SFIMSG
043564                          4$:
043564  004737  045524                  JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
043570  012704  044650                  MOV     #T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
043574                          5$:
043574                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
043574  104404                                                                  TRAP    C$BSEG

043576  005037  002220          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
043602  005037  002222                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
043606  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
043612  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
043616  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
043620  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043622                                  ERRDF   ERRNO,T11SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
043622  104455                                                                  TRAP    C$ERDF
```

```
                043624  002117                                                          .WORD   1103
                043626  045214                                                          .WORD   T11SSR
                043630  012036                                                          .WORD   PKTSSR
3126            043632  005237  002220          INC     FATFLG          ;SET FATAL ERROR FLAG
3127            043636                  15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
                043636  104406                                                          TRAP    C$CLP1
3128            043640                          ESCAPE  SEG             ;BY-PASS SUBTEST IF FATAL ERROR
                043640  104410                                                          TRAP    C$ESCAPE
                043642  000074                                                          .WORD   10000$-.
3129            043644  005737  002222          TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
3130            043650  001004                  BNE     22$             ;BRANCH IF YES
3134            043652                          ERRHRD  ERRNO,T11NINT,PKTSSR
                043652  104456                                                          TRAP    C$ERHRD
                043654  002120                                                          .WORD   1104
                043656  045344                                                          .WORD   T11NINT
                043660  012036                                                          .WORD   PKTSSR
3135            043662  016501  000002  22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
3136            043666  012702  000200          MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
3137            043672  032701  000100          BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
3138            043676  001402                  BEQ     25$             ;BRANCH IF NOT OFF-LINE
3139            043700  052702  000100          BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
3140            043704  020201          25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
3141            043706  001404                  BEQ     30$             ;OKAY IF MATCH
3145            043710                          ERRHRD  ERRNO,T11NBA,PKTSSR     ;NBA NOT ZERO
                043710  104456                                                          TRAP    C$ERHRD
                043712  002121                                                          .WORD   1105
                043714  044762                                                          .WORD   T11NBA
                043716  012036                                                          .WORD   PKTSSR
3146            043720                  30$:
3147            043720  004737  011104  35$:    JSR     PC,CKRAM        ;CHECK RAM TO MEMORY
3148            043724  103405                  BCS     59$             ;RAM OK GO ON
3152            043726                          ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
                043726  104456                                                          TRAP    C$ERHRD
                043730  002122                                                          .WORD   1106
                043732  004741                                                          .WORD   PKTRAM
                043734  015500                                                          .WORD   RAMERR
3153            043736                          ENDSEG                  ;<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<
                043736                                                          10000$:
                043736  104405                                                          TRAP    C$ESEG
3154
3155
3156            043740                  59$:    ENDSUB                  ;\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\
                043740                                                          L10103:
                043740  104403                                                          TRAP    C$ESUB
3157
3158            043742  005737  002220          TST     FATFLG          ;ANY FATAL ERRORS ?
3159            043746  001402                  BEQ     60$             ;BRANCH IF NOT
3160            043750  004737  017172          JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
3161            043754                  60$:
```

```
                        ;+
                        ;
                        ;TEST 11, SUBTEST 2
                        ;
                        ;CHECK THAT NON-ZERO MODE BITS BEING SET CAUSES
                        ;INITIALIZE COMMAND TO BE REJECTED
                        ;
                        ;-

                                BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
                                                                        T11.2:
                                                                                TRAP    C$BSUB

                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
                                                                                MOV     #PRI00,R0
                                                                                TRAP    C$SPRI
                                BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
                                                                                TRAP    C$BSEG

                                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
                                BCS     3$                      ;BR IF SOFT INIT = OK
                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
                                                                                TRAP    C$ERDF
                                                                                .WORD   1107
                                                                                .WORD   SFIERR
                                                                                .WORD   SFIMSG
                        3$:
                                MOV     #T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
                                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
                                BCS     4$                      ;BR, IF COMMAND ISSUED OK
                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
                                                                                TRAP    C$ERHRD
                                                                                .WORD   1108
                                                                                .WORD   WRTMSG
                                                                                .WORD   SFIMSG
                        4$:
                                JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
                                MOV     #T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
                        5$:
                        10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
                                BIS     #P.MODE,(R4)            ;NON-ZERO COMMAND MODE BITS
                                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,T11SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                                                                                TRAP    C$ERDF
                                                                                .WORD   1109
                                                                                .WORD   T11SSR
                                                                                .WORD   PKTSSR
                        15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                TRAP    C$CLP1
                                ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
                                                                                TRAP    C$ESCAPE
                                                                                .WORD   10000$-.
```

```
 3209 044102  005737  002222          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 3210 044106  001004                  BNE     22$                     ;BRANCH IF YES
 3214 044110                          ERRHRD  ERRNO,T11NINT,PKTSSR
      044110  104456                                                  TRAP    C$ERHRD
      044112  002126                                                  .WORD   1110
      044114  045344                                                  .WORD   T11NINT
      044116  012036                                                  .WORD   PKTSSR
 3215 044120  016501  000002  22$:   MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 3216 044124  012702  100206          MOV     #SC!SSR!TSREJ,R2        ;EXPECTED CONTENTS OF TSSR
 3217 044130  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 3218 044134  001402                  BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 3219 044136  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 3220 044142  020201          25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 3221 044144  001404                  BEQ     30$                     ;OKAY IF MATCH
 3225 044146                          ERRHRD  ERRNO,T112REJ,PKTSSR    ;COMMAND NOT REJECTED
      044146  104456                                                  TRAP    C$ERHRD
      044150  002127                                                  .WORD   1111
      044152  045022                                                  .WORD   T112REJ
      044154  012036                                                  .WORD   PKTSSR
 3226 044156                  30$:
 3227 044156  004737  011104  35$:   JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
 3228 044162  103405                  BCS     59$                     ;RAM OK GO ON
 3232 044164                          ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
      044164  104456                                                  TRAP    C$ERHRD
      044166  002130                                                  .WORD   1112
      044170  004741                                                  .WORD   PKTRAM
      044172  015500                                                  .WORD   RAMERR
 3233 044174                          ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      044174                                                  10000$:
      044174  104405                                                  TRAP    C$ESEG
 3234
 3235 044176          59$:    ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      044176                                                  L10104:
      044176  104403                                                  TRAP    C$ESUB
```

```
                                ;+
                                ;
                                ;TEST 11, SUBTEST 3
                                ;
                                ;CHECK THAT THE GET STATUS COMMAND IS ACCEPTED
                                ;
                                ;-
                                
044200                          BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
044200                                                                          T11.3:
044200  104402                                                                  TRAP    C$BSUB
                                
044202                          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
044202  012700  000000                                                          MOV     #PRI00,R0
044206  104441                                                                  TRAP    C$SPRI
044210                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
044210  104404                                                                  TRAP    C$BSEG
044212  004737  015764          JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
044216  103405                  BCS     3$                      ;BR IF SOFT INIT = OK
044220  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
044222                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
044222  104455                                                                  TRAP    C$ERDF
044224  002131                                                                  .WORD   1113
044226  003646                                                                  .WORD   SFIERR
044230  012024                                                                  .WORD   SFIMSG
044232                  3$:
044232  012704  044720          MOV     #T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
044236  004737  010652          JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
044242  103404                  BCS     4$                      ;BR, IF COMMAND ISSUED OK
044244                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
044244  104456                                                                  TRAP    C$ERHRD
044246  002132                                                                  .WORD   1114
044250  005052                                                                  .WORD   WRTMSG
044252  012024                                                                  .WORD   SFIMSG
044254                  4$:
044254  004737  045524          JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
044260  012704  044650          MOV     #T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
044264                  5$:
044264  005037  002222  10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
044270  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
044274  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
044300  103405                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
044302  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
044304                          ERRDF   ERRNO,T11SR2,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
044304  104455                                                                  TRAP    C$ERDF
044306  002133                                                                  .WORD   1115
044310  045270                                                                  .WORD   T11SR2
044312  012036                                                                  .WORD   PKTSSR
044314                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
044314  104406                                                                  TRAP    C$CLP1
044316                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
044316  104410                                                                  TRAP    C$ESCAPE
044320  000074                                                                  .WORD   10000$-.
044322  005737  002222          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
044326  001004                  BNE     22$                     ;BRANCH IF YES
044330                          ERRHRD  ERRNO,T11NINT,PKTSSR
044330  104456                                                                  TRAP    C$ERHRD
```

```
                044332  002134                                                          .WORD   1116
                044334  045344                                                          .WORD   T11NINT
                044336  012036                                                          .WORD   PKTSSR
3285            044340  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
3286            044344  012702  000200                  MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
3287            044350  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
3288            044354  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
3289            044356  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
3290            044362  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
3291            044364  001404                          BEQ     30$                     ;OKAY IF MATCH
3295            044366                                  ERRHRD  ERRNO,T113REJ,PKTSSR    ;COMMAND NOT ACCEPTED
                044366  104456                                                          TRAP    C$ERHRD
                044370  002135                                                          .WORD   1117
                044372  045103                                                          .WORD   T113REJ
                044374  012036                                                          .WORD   PKTSSR
3296            044376                  30$:
3297            044376  004737  011104  35$:    JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
3298            044402  103405                  BCS     59$                     ;RAM OK GO ON
3302            044404                          ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
                044404  104456                                                          TRAP    C$ERHRD
                044406  002136                                                          .WORD   1118
                044410  004741                                                          .WORD   PKTRAM
                044412  015500                                                          .WORD   RAMERR
3303            044414                          ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                044414                                                                  10000$:
                044414  104405                                                          TRAP    C$ESEG
3304
3305
3306            044416                  59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                044416                                                                  L10105:
                044416  104403                                                          TRAP    C$ESUB
```

```
 3308                                   ;+
 3309                                   ;
 3310                                   ;TEST 11, SUBTEST 4
 3311                                   ;
 3312                                   ;SUBTEST TO VERIFY THAT A GET STATUS COMMAND IS
 3313                                   ;REJECTED IF A NON-ZERO COMMAND MODE FIELD IS USED
 3314                                   ;
 3315                                   ;-
 3316
 3317 044420                                   BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
      044420                                                                                    T11.4:
      044420  104402                                                                           TRAP    C$BSUB
 3318
 3319 044422                                   SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      044422  012700  000000                                                                   MOV     #PRI00,R0
      044426  104441                                                                           TRAP    C$SPRI
 3320 044430                                   BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      044430  104404                                                                           TRAP    C$BSEG
 3321 044432  004737  015764                   JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 3322 044436  103405                           BCS     3$                      ;BR IF SOFT INIT = OK
 3326 044440  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 3327 044442                                   ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      044442  104455                                                                           TRAP    C$ERDF
      044444  002137                                                                           .WORD   1119
      044446  003646                                                                           .WORD   SFIERR
      044450  012024                                                                           .WORD   SFIMSG
 3328 044452                            3$:
 3329 044452  012704  044720                    MOV     #T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
 3330 044456  004737  010652                    JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 3331 044462  103404                            BCS     4$                      ;BR, IF COMMAND ISSUED OK
 3335 044464                                    ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      044464  104456                                                                           TRAP    C$ERHRD
      044466  002140                                                                           .WORD   1120
      044470  005052                                                                           .WORD   WRTMSG
      044472  012024                                                                           .WORD   SFIMSG
 3336 044474                            4$:
 3337 044474  004737  045524                    JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
 3338 044500  012704  044650                    MOV     #T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
 3339 044504                            5$:
 3340 044504  005037  002222            10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 3341 044510  052714  007000                    BIS     #007000,(R4)            ;SET TO NON-ZERO MODE
 3342 044514  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 3343 044520  004737  016326                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 3344 044524  103405                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 3345 044526  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 3349 044530                                    ERRDF   ERRNO,T11SR2,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      044530  104455                                                                           TRAP    C$ERDF
      044532  002141                                                                           .WORD   1121
      044534  045270                                                                           .WORD   T11SR2
      044536  012036                                                                           .WORD   PKTSSR
 3350 044540                            15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      044540  104406                                                                           TRAP    C$CLP1
 3351 044542                                    ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
      044542  104410                                                                           TRAP    C$ESCAPE
      044544  000076                                                                           .WORD   L10106-.
 3352 044546  005737  002222                    TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 3353 044552  001004                            BNE     22$                     ;BRANCH IF YES
```

```
3357 044554                                    ERRHRD  ERRNO,T11NINT,PKTSSR
     044554  104456                                                        TRAP    C$ERHRD
     044556  002142                                                        .WORD   1122
     044560  045344                                                        .WORD   T11NINT
     044562  012036                                                        .WORD   PKTSSR
3358 044564  016501  000002          22$:      MOV     TSSR(R5),R1         ;GET THE CONTENTS OF TSSR
3359 044570  012702  100206                    MOV     #SC!SSR!TSREJ,R2    ;EXPECTED CONTENTS OF TSSR
3360 044574  032701  000100                    BIT     #OFL,R1             ;IS OFF-LINE BIT SET ?
3361 044600  001402                            BEQ     25$                 ;BRANCH IF NOT OFF-LINE
3362 044602  052702  000100                    BIS     #OFL,R2             ;SET OFF-LINE IN EXPECTED DATA
3363 044606  020201                  25$:      CMP     R2,R1               ;DOES EXPECTED MATCH RECEIVED ?
3364 044610  001404                            BEQ     30$                 ;OKAY IF MATCH
3368 044612                                    ERRHRD  ERRNO,T114REJ,PKTSSR ;COMMAND NOT REJECTED
     044612  104456                                                        TRAP    C$ERHRD
     044614  002143                                                        .WORD   1123
     044616  045133                                                        .WORD   T114REJ
     044620  012036                                                        .WORD   PKTSSR
3369 044622                            30$:
3370
3371 044622  004737  011104          35$:      JSR     PC,CKRAM            ;CHECK RAM TO MEMORY
3372 044626  103404                            BCS     59$                 ;RAM OK GO ON
3376 044630                                    ERRHRD  ERRNO,PKTRAM,RAMERR ;THEY DON'T MATCH
     044630  104456                                                        TRAP    C$ERHRD
     044632  002144                                                        .WORD   1124
     044634  004741                                                        .WORD   PKTRAM
     044636  015500                                                        .WORD   RAMERR
3377 044640                            59$:
3378 044640                                    ENDSEG                      ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<
     044640                                                                10000$:
     044640  104405                                                        TRAP    C$ESEG
3379 044642                                    ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     044642                                                                L10106:
     044642  104403                                                        TRAP    C$ESUB
3380
```

```
3382 044644                             EXIT    TST                     ;ALL DONE THIS TEST
     044644  104432                                                             TRAP    C$EXIT
     044646  000762                                                             .WORD   L10102-.
3383
3384                                    ;+
3385                                    ;LOCAL STORAGE FOR THIS TEST
3386                                    ;-
3387
3391 044650                             T11PACKET:                      ;COMMAND PACKET FOR TEST
3392 044650  100204                             .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
3393 044652  044660                             .WORD   T11DATA         ;ADDRESS OF CHARACTERISTICS BLOCK
3394 044654  000000                             .WORD   0
3395 044656  000010                             .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
3396
3397 044660                             T11DATA:                                ;CHARACTERISTICS DATA BLOCK
3398 044660  044672                             .WORD   T11BFR          ;ADDRESS OF MESSAGE BUFFER
3399 044662  000000                             .WORD   0
3400 044664  000016                             .WORD   14.             ;LENGTH OF MESSAGE BUFFER
3401 044666  000000  000000                     .WORD   0,0
3402
3403 044672                             T11BFR: .BLKW   8.              ;MESSAGE BUFFER
3404
3405
3407         044720                             .=<.+10>&177770
3409 044720                             T11PK2:                         ;COMMAND PACKET FOR TEST
3410 044720  100204                             .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
3411 044722  044730                             .WORD   T11DTA          ;ADDRESS OF CHARACTERISTICS BLOCK
3412 044724  000000                             .WORD   0
3413 044726  000010                             .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
3414
3415 044730                             T11DTA:                         ;CHARACTERISTICS DATA BLOCK
3416 044730  044742                             .WORD   T11BF2          ;ADDRESS OF MESSAGE BUFFER
3417 044732  000000                             .WORD   0
3418 044734  000016                             .WORD   14.             ;LENGTH OF MESSAGE BUFFER
3419 044736  000000  000000                     .WORD   0,0
3420
3421 044742                             T11BF2: .BLKW   8.              ;MESSAGE BUFFER
3422
3423
3424
3425                                    ;+
3426                                    ;LOCAL TEXT MESSAGES FOR TEST
3427                                    ;-
3428
3429 044762      111     116     111    T11NBA: .ASCIZ  'INITIALIZE Command Not Accepted'
3430 045022      111     116     111    T112REJ:        .ASCIZ  'INITIALIZE Not Rejected With Non-Zero Mode Field'
3431 045103      107     105     124    T113REJ:        .ASCIZ  'GET STATUS Not Accepted'
3432 045133      107     105     124    T114REJ:        .ASCIZ  'GET STATUS Not Rejected With Non-Zero Mode Field'
3433 045214      103     157     156    T11SSR: .ASCIZ  'Contents of TSSR Incorrect After INITIALIZE'
3434 045270      103     157     156    T11SR2: .ASCIZ  'Contents of TSSR Incorrect After GET STATUS'
3435 045344      105     170     160    T11NINT:        .ASCIZ  'Expected Interrupt Not Received On INITIALIZE'
3436 045422      111     156     143    T11TSBA:        .ASCIZ  'Incorrect TSBA Address After INITIALIZE'
3437 045472      116     157     156    TST11ID:        .ASCIZ  'Non-Tape Motion Commands'
3438                                            .EVEN
3439
```

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

045524                                  T11REST:
045524                                          SAVREG                  ;SAVE THE REGISTERS
045530  012701  044650                          MOV     #T11PACKET,R1   ;START OF THE PACKET
045534  012721  100213                          MOV     #100213,(R1)+   ;INITIALIZE WITH ACK, IE
045540  005021                                  CLR     (R1)+           ;ADDRESS OF CHAR DATA BLOCK
045542  005021                                  CLR     (R1)+           ;EXTENDED ADDRESS
045544  005021                                  CLR     (R1)+           ;SIZE OF DATA BLOCK IN BYTES
045546  005021                                  CLR     (R1)+           ;ADDRESS OF MESSAGE BUFFER
045550  005021                                  CLR     (R1)+
045552  005021                                  CLR     (R1)+           ;LENGTH OF MESSAGE BUFFER
045554  005021                                  CLR     (R1)+
045556  005011                                  CLR     (R1)
045560  005037  044672                          CLR     T11BFR          ;CLEAR 1ST LOC IN MESSAGE BUFFER
045564  000207                                  RTS     PC              ;RETURN
                                        ;+
                                        ;
                                        ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
                                        ;GET STATUS COMMAND
                                        ;
                                        ;-

045566                                  T11RT2:
045566                                          SAVREG                  ;SAVE THE REGISTERS
045572  012701  044650                          MOV     #T11PACKET,R1   ;START OF THE PACKET
045576  012721  100217                          MOV     #100217,(R1)+   ;GET STATUS WITH ACK, IE
045602  005021                                  CLR     (R1)+           ;ADDRESS OF CHAR DATA BLOCK
045604  005021                                  CLR     (R1)+           ;EXTENDED ADDRESS
045606  005021                                  CLR     (R1)+           ;SIZE OF DATA BLOCK IN BYTES
045610  005021                                  CLR     (R1)+           ;ADDRESS OF MESSAGE BUFFER
045612  005021                                  CLR     (R1)+
045614  005021                                  CLR     (R1)+           ;LENGTH OF MESSAGE BUFFER
045616  005021                                  CLR     (R1)+
045620  005011                                  CLR     (R1)
045622  005037  044672                          CLR     T11BFR          ;CLEAR 1ST LOC IN MESSAGE BUFFER
045626  000207                                  RTS     PC              ;RETURN
045630                                          ENDTST
045630                                                                          L10102:
045630  104401                                                                          TRAP    C$ETST
045632                                  ENDMOD
```

```
                                        .TITLE  TSV6 - PARAMETER CODING


                                        BGNMOD  TSV6
                                TSV6::


                                        .SBTTL  HARDWARE PARAMETER CODING SECTION

                                ;++
                                ; THE HARDWARE 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.
                                ;--
045632                          BGNHRD
045632  000010                  .WORD L10107-L$HARD/2
045634                  L$HARD::
                                GPRMA   HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
045634  000031                  .WORD   T$CODE
045636  045654                  .WORD   HPM1
045640  160010                  .WORD   T$LOLIM
045642  177776                  .WORD   T$HILIM
045644                          GPRMA   HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
045644  001031                  .WORD   T$CODE
045646  045710                  .WORD   HPM2
045650  000000                  .WORD   T$LOLIM
045652  000776                  .WORD   T$HILIM
                                ;GPRMD  HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
045654                          ENDHRD
                                .EVEN
045654                  L10107:
045654  104     105     126     HPM1:   .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
045710  111     116     124     HPM2:   .ASCIZ  'INTERRUPT VECTOR    '
045734  111     116     124     HPM3:   .ASCIZ  'INTERRUPT PRIORITY      '
                                .EVEN
```

```
                                        .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.
                                ;--
045764                                  BGNSFT
045764  000003                          .WORD L10110-L$SOFT/2
045766                          L$SOFT::
                                ;       GPRML   SPM1,0,-1,YES                   ; GET TRANSPORT TEST FLAG.
045766                                  GPRML   SPM4,2,-1,YES                   ; GET ITERATION CONTROL.
045766  001130                          .WORD   T$CODE
045770  046024                          .WORD   SPM4
045772  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
045774                                  ENDSFT
                                        .EVEN
045774                          L10110:

045774  105  116  101  SPM1:    .ASCIZ  'ENABLE TRANSPORT TESTS '
046024  111  116  110  SPM4:    .ASCIZ  'INHIBIT ITERATIONS     '
046054  120  105  122  SPM6:    .ASCIZ  'PER TEST ERROR LIMIT   '
046104  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).
                                ;

046134                          PATCH::

                                        .BLKW   32.

                                        .=.!377+1
046400                                  LASTAD          ;SET LAST USED ADDRESS.
046400                                  .EVEN
046400  000000                          .WORD   0
046402  000000                          .WORD   0
046404                          L$LAST::
046404                                  ENDMOD
000001                                  .END
```

```
ADDSSR  012116 G        C$AU  = 000052        DEVDRO  023366          FRESIZ  003122 G        INTFLA  016135
ADR   = 000020 G        C$AUTO= 000061        DEVNRD  023305          FUSI    004113           INTMAS  016134
AMBTSS  006631           C$BRK = 000022        DEVNXR  023223          F$AU  = 000015           INTR    016206 G
ASSEMB= 000010           C$BSEG= 000004        DEVONL  023153          F$AUTO= 000020           INTREC  002222 G
A1716 = 000003           C$BSUB= 000002        DEVSUM  023116          F$BGN = 000040           INTVEC  016136
BADDAT  003152 G         C$CEFG= 000045        DFPTBL  002154 G        F$CLEA= 000007           INTX    004274
BADSSR  015670 G         C$CLCK= 000062        DIAGMC= 000000          F$DU  = 000016           INVERT  021176 G
BDVPCR= 177520 G         C$CLEA= 000012        DICEA = 000001          F$END = 000041           IOKCKI= 000200
BENBSW  002226 G         C$CLOS= 000035        DSBINT  016174          F$HARD= 000004           IOKSTP= 000001
BIE   = 040000           C$CLP1= 000006        DUAD12  004637          F$HW  = 000013           IPRI    002210 G
BIT0  = 000001 G         C$CVEC= 000036        DUFLG   003106 G        F$INIT= 000006           ISR   = 000100 G
BIT00 = 000001 G         C$DCLN= 000044        DUMMY   003056          F$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$ERHR= 000056        EMAXDU  016767          F$SEG = 000003           I$HRD = 000041
BIT07 = 000200 G         C$ERRO= 000060        EN    = 000000          F$SOFT= 000005           I$INIT= 000041
BIT08 = 000400 G         C$ERSF= 000054        ENAINT  016142          F$SRV = 000010           I$MOD = 000041
BIT09 = 001000 G         C$ERSO= 000057        ENVIRN  020620          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   006354          F$TEST= 000001           I$PTAB= 000041
BIT11 = 004000 G         C$ESUB= 000003        EPRT2   006354          GDDAT   003154 G         I$PWR = 000041
BIT12 = 010000 G         C$ETST= 000001        ERCM    011723          GERRMA  002172 G         I$RPT = 000041
BIT13 = 020000 G         C$EXIT= 000032        ERRHI   002234 G        GETPAT  020164 G         I$SEG = 000041
BIT14 = 040000 G         C$GETB= 000026        ERRK    016746          GETSEL  020246 G         I$SETU= 000041
BIT15 = 100000 G         C$GETW= 000027        ERRLO   002236 G        G$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  003372          G$EXCP= 000400           I$TST = 000041
BIT5  = 000040 G         C$GPRI= 000040        ERTABL  003172          G$HILI= 000002           J$JMP = 000167
BIT6  = 000100 G         C$INIT= 000011        ESUM    016750          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  015472 G        G$OFSI= 000376           KIPAR3= 172346
BOE   = 000400 G         C$MSG = 000023        EXPD    002230 G        G$PRMA= 000001           KIPAR4= 172350
BRINIT  004453           C$OPEN= 000034        EXPGOT  004527          G$PRMD= 000002           KIPAR5= 172352
BSEL0 = 000000           C$PNTB= 000014        EXPGT2  004563          G$PRML= 000000           KIPAR6= 172354
BSEL1 = 000001           C$PNTF= 000017        EXPMSG  002320 G        G$RADA= 000140           KIPAR7= 172356
CHKAMB  016034           C$PNTS= 000016        EXPREC  015464 G        G$RADB= 000000           KIPDR0= 172300
CHKMAN  020470 G         C$PNTX= 000015        EXTA    005766          G$RADD= 000040           KIPDR1= 172302
CHKTSS  016326           C$QIO = 000377        EXTEND  005764          G$RADL= 000120           KIPDR2= 172304
CKDROP  017172           C$RDBU= 000007        EXTFEA  002224 G        G$RADO= 000020           KIPDR3= 172306
CKEMAX  017072           C$REFG= 000047        E$END = 002100          G$XFER= 000004           KIPDR4= 172310
CKMSG   011350 G         C$RESE= 000033        E$LOAD= 000035          G$YES = 000010           KIPDR5= 172312
CKMSG2  011470 G         C$REVI= 000003        FATERR= 000060          HIADDR= 001400           KIPDR6= 172314
CKRAM   011104 G         C$RFLA= 000021        FATFLG  002220 G        HOE   = 100000 G         KIPDR7= 172316
CKRAM2  011214 G         C$RPT = 000025        FERCM   011712          HPM1    045654           KTENAB  003130 G
CMDPKT  021250 G         C$SEFG= 000046        FIFEXP  012160 G        HPM2    045710           KTFLG   003126 G
CMPMEM  017650           C$SPRI= 000041        FIF1MS  012232          HPM3    045734           KTINIT  021044
CONFIG  017240           C$SVEC= 000037        FIF2MS  012301          IBE   = 010000 G         KTOFF   017264
COUNT   002306 G         C$TPRI= 000013        FILLME  017412          IDU   = 000040 G         KTON    017246
CSRADD  002204 G         DATA    002310 G      FNDINT  004211          IER   = 020000 G         LERRMA  002170 G
CTAB    003160 G         DATASC  020222        FORCER  002174 G        IFAULT  004252           LISTAL= 000001
CTABE   003172 G         DEBUGM  011622        FREE    003120 G        INCERK  017034           LOE   = 040000 G
CTABM   003160 G         DEVCNT  002216 G      FREEHI  003124          INTCPC  016140           LOOPCN  002214 G
```

```
LOOPCO  013116          L10001  002174          L10073  037006          NXR     003734          PRIO5 = 000240 G
LOOPFL  003156 G        L10002  005762          L10074  037212          NXRERR  005732 G        PRIO6 = 000300 G
LOT   = 000010 G        L10003  012034          L10075  043472          NXRX    003773          PRIO7 = 000340 G
L$ACP   002110 G        L10004  012052          L10076  040772          NXTU    022012          PRMESS  014232
L$APT   002036 G        L10005  012070          L10077  041364          OFL   = 000100          PRMNO   002316 G
L$AU    022342 G        L10006  012076          L10100  042254          ONEFIL= 000000          PRMSGE  014542 G
L$AUT   002070 G        L10007  012114          L10101  042502          O$APTS= 000000          PRMSG0  014722
L$AUTO  022546 G        L10010  012132          L10102  045630          O$AU  = 000001          PRMSG1  014767
L$CCP   002106 G        L10011  012156          L10103  043740          O$BGNR= 000001          PRMSG2  015025
L$CLEA  022626 G        L10012  012230          L10104  044176          O$BGNS= 000001          PR0ASC  014410
L$CO    002032 G        L10013  012400          L10105  044416          O$DU  = 000001          PR1ASC  014455
L$DEPO  002011 G        L10014  013114          L10106  044642          O$ERRT= 000000          PST32W  003146 G
L$DESC  003404 G        L10015  013742          L10107  045654          O$GNSW= 000001          PUNIT   022274
L$DESP  002076 G        L10016  013764          L10110  045774          O$POIN= 000001          PW.D11= 000021
L$DEVP  002060 G        L10017  015470          MEMADD  013744 G        O$SETU= 000000          PW.D13= 000022
L$DISP  002124 G        L10020  015476          MEMCK   021266 G        PASRPT  022044          PW.D22= 000020
L$DLY   002116 G        L10021  015504          MENASC  020437          PATCH   046134 G        PW.NOP= 000000
L$DTP   002040 G        L10022  015516          MENERR  020364          PATDAT  020220          PW.NO1= 000023
L$DTYP  002034 G        L10023  015540          MENRES  020466          PC.ERA= 002400          PW.RDE= 000024
L$DU    022440 G        L10024  015566          MMVEC = 000250          PC.IER= 002000          PW.RDR= 000001
L$DUT   002072 G        L10025  015726          MSA.FR= 000006          PC.NOO= 001000          PW.RDS= 000005
L$DVTY  003376 G        L10026  016236          MSA.NO= 000000          PC.REL= 000000          PW.RFI= 000003
L$EF    002052 G        L10030  022272          MSA.NR= 000004          PC.REW= 000400          PW.WCT= 000006
L$ENVI  002044 G        L10031  022436          MSA.VO= 000002          PKBCNT= 000006          PW.WFI= 000004
L$ETP   002102 G        L10032  022544          MSGEXP  012134 G        PKHI  = 000004          PW.WFM= 000007
L$EXP1  002046 G        L10033  022624          MSGLOO  013054 G        PKLOW = 000002          PW.WMI= 000010
L$EXP4  002064 G        L10034  022652          MSGSTA  012340 G        PKTADD  007550          PW.WNP= 000011
L$EXP5  002066 G        L10035  023114          MSGSUB  013732 G        PKTFRM  007512          PW.WTR= 000002
L$HARD  045634 G        L10036  023654          MS.ATT= 000006          PKTGET  012054 G        P.ACK = 100000
L$HIME  002120 G        L10037  023522          MS.EXT= 000200          PKTMES  012100 G        P.CMD = 000037
L$HPCP  002016 G        L10040  023604          MS.RSD= 000001          PKTRAM  004741 G        P.CONT= 000012
L$HPTP  002022 G        L10041  024352          MS.RSF= 000020          PKTSSR  012036 G        P.CVC = 040000
L$HW    002154 G        L10042  025044          MS.RST= 000010          PNT   = 001000 G        P.FMT = 000140
L$ICP   002104 G        L10043  026400          M2901   026124          PRAMPK  013766          P.FORM= 000011
L$INIT  021546 G        L10044  025306          M8186   005550          PRASC   014513          P.GETS= 000017
L$LADP  002026 G        L10045  025606          M8189   005641          PRBEXP  015460          P.IE  = 000200
L$LAST  046404 G        L10046  026104          NBA   = 002000          PRBMSG  015326          P.INIT= 000013
L$LOAD  002100 G        L10047  027504          NEWPAS  022000          PRBREC  015462          P.MODE= 007400
L$LUN   002074 G        L10050  026752          NODEV   003110 G        PRBTOT  015413          P.OPP = 020000
L$MREV  002050 G        L10051  027322          NOINIT  004331          PRBYTE  015112 G        P.POSI= 000010
L$NAME  002000 G        L10052  030762          NOINTR  004215          PRI   = 002000 G        P.READ= 000001
L$PRIO  002042 G        L10053  030034          NOITS   002166 G        PRIADD  010154          P.SWB = 010000
L$PROT  021536 G        L10054  030360          NOMAN   020524          PRIA0   010224          P.WRIT= 000005
L$PRT   002112 G        L10055  034350          NOMEM   005454          PRIBX0  007606 G        P.WRTC= 000004
L$REPP  002062 G        L10056  031314          NP.IR = 000200          PRIEQU  010054          P.WRTS= 000006
L$REV   002010 G        L10057  031600          NP.LOO= 000040          PRIPKT  007364 G        QVP     002202 G
L$RPT   02265. G        L10060  032044          NP.OUT= 000100          PRIRAM  010062          RAMASC  014146
L$SOFT  045766 G        L10061  032272          NP.WRP= 000020          PRITAD  010270          RAMDAT  002240 G
L$SPC   002056 G        L10062  032536          NSI     004146          PRITSS  006020          RAMERR  015500 G
L$SPCP  002020 G        L10063  033076          NSINIT  004403          PRIT0   010352          RAMEXP  015520 G
L$SPTP  002024 G        L10064  035254          NUL     004523          PRIT1   010415          RAMFOR  010112
L$STA   002030 G        L10065  040370          NULCR   004524          PRIXOR  007736 G        RAMSIZ  002300 G
L$SW    002164 G        L10066  035516          NXM   = 004000          PRI00 = 000000 G        RAMTAD  015506 G
L$TEST  002114 G        L10067  035756          NXMFLG  003132 G        PRI01 = 000040 G        RCVHIA  002302 G
L$TIML  002014 G        L10070  036162          NXMHI   003136 G        PRI02 = 000100 G        RCVLOA  002304 G
L$UNIT  002012 G        L10071  036350          NXMLO   003134 G        PRI03 = 000140 G        RDERR   005202
L10000  002162          L10072  036554          NXMTST  021442          PRI04 = 000200 G        RECMSG  002464 G
```

```
RECV    002232 G    S1.IEO= 010000      TST6ID  030743      T1.1    023456      T4      025046 G
REGSAV  020130      S1.IFM= 001000      TST7ID  034253      T1.2    023536      T4LOOP  025060
RETERR  005366      S1.IHE= 000400      TST8ID  035237      T10     040372 G    T4.1    025046
REWIND  011004 G    S1.IID= 004000      TST9ID  040271      T10BFR  042532      T4.2    025310
RMCHBE= 000167      S1.I1R= 020000      TSV2    002000 G    T10BUF  042574      T4.3    025610
RMCHEN= 000200      S1.I2R= 040000      TSV3    002174 G    T10DAT  042520      T5      026402 G
RMMSGB= 000215      S1.PAR= 100000      TSV4    021536 G    T10DTA  042562      T5ADDR  027442
RMMSGE= 000234      S2.ATI= 000010      TSV5    023436 G    T10INT  043230      T5LOOP  026420
RMPKTB= 000201      S2.BTI= 000004      TSV6    045632 G    T10LOO  040410      T5MEM   027404
RMPKTE= 000210      S2.DIM= 000200      TTIBFR= 177562 G    T10MBF  042614      T5.1    026424
RMR   = 010000      S2.ILW= 000100      TTICSR= 177560 G    T10NBA  042711      T5.2    026772
RWPACK  011100      S2.INR= 000020      TTIVEC= 000060 G    T10NIN  043137      T6      027506 G
SC    = 100000      S2.OUT= 000040      T$ARGC= 000003      T10NNB  042773      T6INT   030533
SCE   = 020000      S2.UND= 000003      T$CODE= 001130      T10PAC  042510      T6LOOP  027524
SCHERR  005274      TBLEND= 003056 G    T$ERRN= 002144      T10PKT  042552      T6NBA   030430
SCME    005007      TCOASC  006472      T$EXCP= 000000      T10RST  043346      T6NINT  030611
SDELAY  010650      TCOCOD  006672      T$FLAG= 000040      T10RT2  043420      T6PACK  030420
SELASC  020432      TEMP1   003112 G    T$GMAN= 000000      T10SSR  043050      T6SSR   030455
SELDAT= 000004      TEMP2   003114 G    T$HILI= 000776      T10.1   040410      T6TSBA  030671
SEL2  = 000002      TERCLS= 000016      T$LAST= 000001      T10.2   040774      T6.1    027524
SETMAP  017306      TESTNO= 000013      T$LOLI= 000000      T10.3   041366      T6.2    030050
SETU    022076      TEXASC  006431      T$LSYM= 010000      T10.4   042256      T7      030764 G
SFFMSG  012072 G    TFCASC  006533      T$LTNO= 000013      T11     043474 G    T7BFR   033146
SFHERR  003701      TIMEXP  015542 G    T$NEST= 177777      T11BFR  044672      T7DATA  033130
SFIERR  003646      TIMSGO  015570      T$NS0 = 000000      T11BF2  044742      T7INT   034101
SFIMSG  012024 G    TINERR  012011      T$NS1 = 000005      T11DAT  044660      T7LOOP  031002
SFPTBL  002164 G    TMPBFR  002630 G    T$NS2 = 000002      T11DTA  044730      T7NBA   033260
SIFLAG  003150 G    TNAM    016674      T$NS3 = 000003      T11LOO  043512      T7NINT  034010
SIMSG   011756      TRANST  002164 G    T$PTNU= 000000      T11NBA  044762      T7PACK  033120
SKIPT   003374      TSBA  = 000000 G    T$SAVL= 177777      T11NIN  045344      T7REST  034302
SOFINI  015764 G    TSBAH = 000001 G    T$SEGL= 177777      T11PAC  044650      T7SP    033140
SPACE   010462 G    TSBAM2  026214      T$SEKO= 010000      T11PK2  044720      T7SSR   033721
SPM1    045774      TSBAM3  026276      T$SUBN= 000004      T11RES  045524      T7TSBA  034170
SPM4    046024      TSDB  = 000000 G    T$TAGL= 177777      T11RT2  045566      T7.1    031002
SPM6    046054      TSDBH = 000001 G    T$TAGN= 010111      T11SR2  045270      T7.2    031330
SPM7    046104      TSFCOD  007232      T$TEMP= 000000      T11SSR  045214      T7.3    031602
SR0   = 177572      TSREJ = 000006      T$TEST= 000013      T11TSB  045422      T7.4    032046
SR1   = 177574      TSSDEF  006602      T$TSTM= 177777      T11.1   043512      T7.5    032274
SR2   = 177576      TSSR  = 000002 G    T$TSTS= 000001      T11.2   043754      T7.6    032540
SR3   = 172516      TSSRBI  003476 G    T$$AU = 010031      T11.3   044200      T72DAT  033166
SSR   = 000200      TSSRFO  006411      T$$AUT= 010033      T11.4   044420      T72DON= 033202
STATCO  012402      TSSRH = 000003 G    T$$CLE= 010034      T112RE  045022      T72NBA  033202
SVCGBL= 000000      TSSX    004014      T$$DU = 010032      T113RE  045103      T72REJ  033333
SVCINS= 000000      TSTBLK  002750 G    T$$HAR= 010107      T114RE  045133      T73REJ  033432
SVCSUB= 000001      TSTCNT  002212 G    T$$HW = 010000      T2      023656 G    T74REJ  033525
SVCTAG= 000000      TSTEND  016710      T$$INI= 010030      T2LOOP  023674      T75REJ  033623
SVCTST= 000001      TSTFLA  002312 G    T$$MSG= 010025      T2SSR   024252      T8      034352 G
S$LSYM= 010000      TSTLOO  016446 G    T$$PRO= 010027      T2TSBA  024140      T8BFR   035002
S0.IDB= 000010      TSTPTR  002314 G    T$$RPT= 010035      T2TSSR  024205      T8DATA  034770
S0.IFB= 000002      TSTSET  016500 G    T$$SEG= 010000      T23A    003140 G    T8LOOP  034370
S0.IFP= 000001      TST1ID  023634      T$$SOF= 010110      T23B    003142 G    T8NVCK  035057
S0.ILD= 000020      TST1OI  043317      T$$SRV= 010026      T3      024354 G    T8PACK  034760
S0.ION= 000040      TST1II  045472      T$$SUB= 010106      T3BFLG  003144 G    T8SSR   035150
S0.IRD= 000100      TST2ID  024324      T$$SW = 010001      T3LOOP  024372      T8VCK   035022
S0.IRW= 000004      TST3ID  025017      T$$TES= 010102      T3SSR   024746      T9      035256 G
S0.ISP= 000200      TST4ID  026356      T1      023436 G    T3TSBA  024636      T9BFR   037242
S1.ICE= 002000      TST5ID  027354      T1LOOP  023454      T3TSSR  024702      T9DATA  037230
```

```
T9INT   040117          UAM    = 000200 G      WRTCHR  010652 G      XSOONL= 000100          X2.EXT= 000200
T9LOOP  035300          UNITN    002200 G      WRTERR  005107        XSOPED= 000010          X2.OPM= 100000
T9NBA   037276          UNREC  = 000006        WRTMSG  005052        XSORLL= 010000          X2.RCE= 040000
T9NINT  040026          USI      004117        WSMBK   021260 G      XSORLS= 040000          X2.REV= 000077
T9PACK  037220          WAITF    016240 G      XFERAS  015730        XSOTMK= 100000          X2.SPA= 035400
T9REST  040316          WC.IFA= 000200         XNXM    016366        XSOVCK= 000020          X2.UNI= 000007
T9SSR   037737          WC.IFE= 000002         XORBFO  007670        XSOWLE= 004000          X2.WCF= 002000
T9TSBA  040206          WC.IGO= 000001         XORFOR  010006        XSOWLK= 000004          X3.DCK= 000010
T9.1    035300          WC.IRE= 000010         XST0   = 000006 G      XXCOMM  003116 G      X3.MBZ= 000006
T9.2    035546          WC.IRW= 000004         XST1   = 000010 G      X$ALWA= 000000          X3.MDE= 177400
T9.3    035760          WC.IOT= 000100         XST2   = 000012 G      X$FALS= 000040          X3.OPI= 000100
T9.4    036164          WC.I1T= 000040         XST3   = 000014 G      X$OFFS= 000400          X3.REV= 000040
T9.5    036352          WC.I5R= 000020         XST4   = 000016 G      X$TRUE= 000020          X3.RIB= 000001
T9.6    036556          WF.IED= 000010         XSOBOT= 000002         X1.COR= 020000          X3.SPA= 000200
T9.7    037010          WF.IER= 000004         XSOEOT= 000001         X1.DLT= 100000          X3.TRF= 000020
T92DAT  037262          WF.IHI= 000200         XSOIE  = 000040        X1.MBZ= 017375          X4.HSP= 100000
T92DON= 037276          WF.IRE= 000040         XSOILA= 000400         X1.RBP= 000400          X4.MBZ= 017400
T92REJ  037351          WF.IWF= 000020         XSOILC= 001000         X1.SPA= 040000          X4.RCE= 040000
T93REJ  037450          WF.IWR= 000100         XSOLET= 020000         X1.UNC= 000002          X4.TSM= 020000
T94REJ  037543          WF.I3R= 000002         XSOMOT= 000200         X2.BUF= 000100          X4.WRC= 000377
T95REJ  037641          WF.I4R= 000001         XSONEF= 002000

. ABS.  046404      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  28224 WORDS  ( 111 PAGES)
DYNAMIC MEMORY:  20346 WORDS  ( 78 PAGES)
ELAPSED TIME:  00:34:19
CVTSAA0,CVTSAA0/-SP=SVC/ML,TSV1A,TSV22A,TSV3A,TSV4,TSV5A,TSV6
```