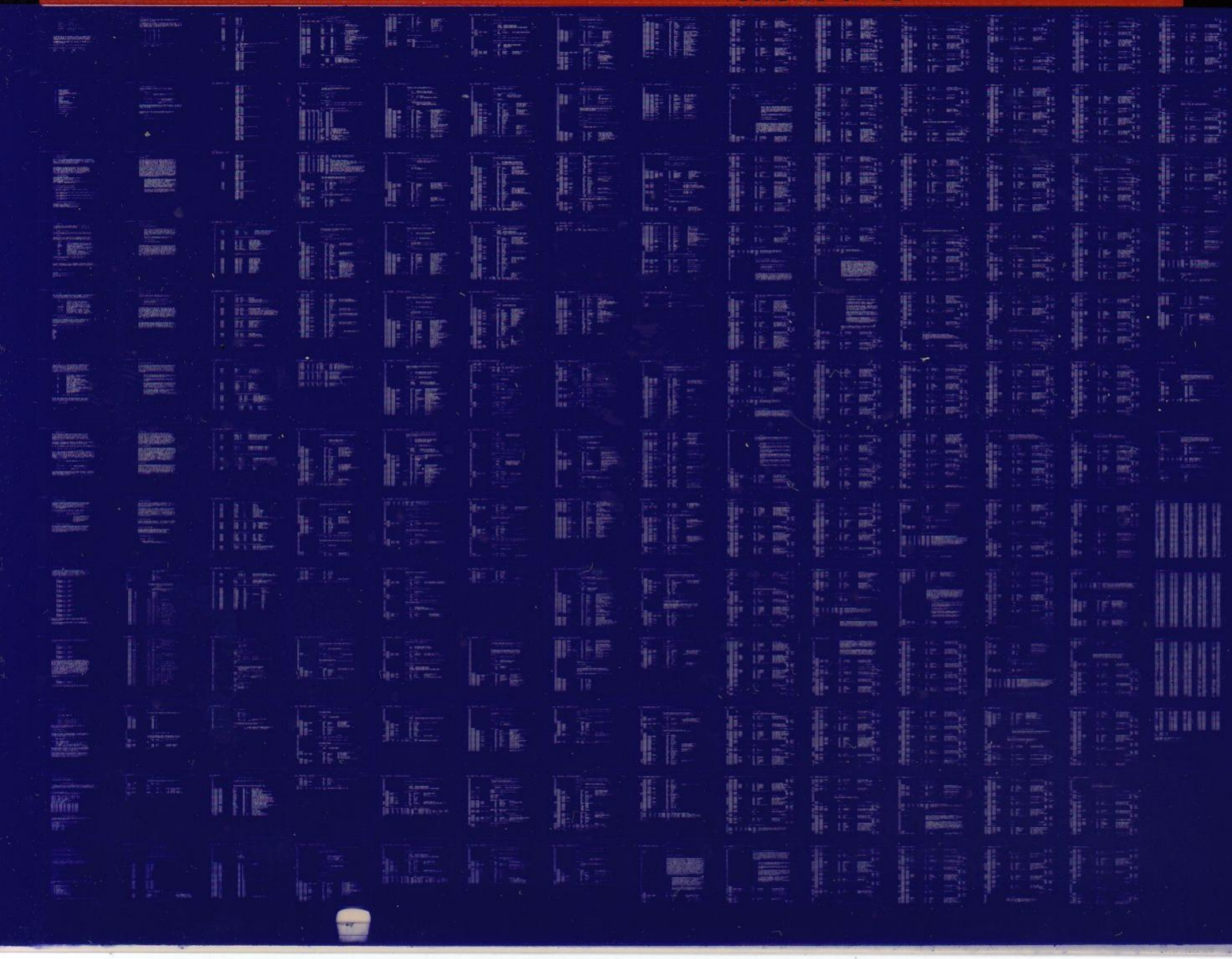
TSV05 TSV05 CTRL LT1
CVTSABO

COPYRIGHT (c) 1982-84
AH-T094B-MC
FICHE 01 OF 01

APR 1985
digital
Made In USA



.REM

IDENTIFICATION

PRODUCT ID: AC-T093B-MC

PRODUCT TITLE: CVTSABO TSV05 CTRL LT1

DECO/DEPO: 1.0

DEPARTMENT: COMPUTER SPECIAL SYSTEMS/PPG

DATE: JUNE 4, 1984

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

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

COPYRIGHT (C) 1982, 1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

TABLE OF CONTENTS

1.0 GENERAL INFORMATION
1.1 PROGRAM ABSTRACT
1.2 SYSTEM REQUIREMENTS
1.3 RELATED DOCUMENTS AND STANDARDS
1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
1.5 ASSUMPTIONS

2.0 OPERATING INSTRUC IONS
2.1 COMMANDS
2.2 SWITCHES
2.3 FLAGS
2.4 HARDWARE QUESTIONS
2.5 SOFTWARE QUESTIONS
2.6 EXTENDED P-TABLE DIALOGUE
2.7 QUICK STARTUP PROCEDURE

3.0 ERROR INFORMATION

4.0 PERFORMANCE AND PROGRESS REPORTS

5.0 DEVICE INFORMATION TABLES

6.0 TEST SUMMARIES

7.0 MAINTENANCE HISTORY

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

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

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

1.2 SYSTEM REQUIREMENTS

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

1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

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

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

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

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

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

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

1.5 ASSUMPTIONS

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

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

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

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

2.1.1 OPERATOR COMMANDS

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

BOOT THE DIAGNOSTIC MEDIA

```
.R VTSA??
DIAG. RUN-TIME SERVICES REV D. APR 79
CVTSA-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 LSI-11 AND UP TO 2 DRIVES PER CONTROLLER

2.5 SOFTWARE QUESTIONS

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

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

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

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

2.6 EXTENDED P-TABLE DIALOGUE

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

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

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

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

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

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

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

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

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

UNIT 7
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  6,7<CR>
Q-FACTOR (O) O ?  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) O ?  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 AN UNRECOVERABLE ERROR OCCURED WHICH INDICATES THAT THE CONTROLLER MAY BE DEFECTIVE

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

ERROR MESSAGE EXAMPLE 3

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

```
CVTS M7D ERR 00121 ON UNIT 00 TST 001 SUB 002 PC: 023306
MOT BIT (XSTO) NOT SET DURING REWIND (EXTENDED FEATURES MODE)
 EXPD  OL0312 RECV  000112 XOR  000200
```

4 0 PERFORMANCE AND PROGRESS REPORTS

AT THE END OF EACH PASS, THE PASS COUNT IS GIVEN ALONG WITH THE TOTAL NUMBER OF ERRORS REPORTED SINCE THE DIAGNOSTIC WAS STARTED. THE "EOP" SWITCH CAN BE USED TO CONTROL HOW OFTEN THE END OF PASS MESSAGE IS PRINTED SECTION 2 2 DESCRIBES SWITCHES

SUCCESSFUL RUN EXAMPLE (LSI 11)

```
DR>STA/FLA-PNT MOE

UNITS (D) ?  1

UNIT 0

DEVICE ADDRESS  (0) 172520 ? <CR>

VECTOR (0) 224 ? <CR>

CHANGE SW (L) ? N<CR>
```

THE ABOVE COMMAND WILL START THE DIAGNOSTIC. THE COMMAND HAS TWO SWITCHES ON WHICH ARE "PRINT EACH TEST NBR AS EXECUTED AND "HALT ON ERROR".

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

0 ERRORS
```

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

PROGRAM RUN TIMES

THE AVERAGE RUN TIMES OF THE PROGRAM ARE LISTED BELOW.THESE FIGURES ARE TO BE USED AS A GUIDE. THE TIMING WAS DONE ON A LSI-11 PROCESSOR WITH A LA34 CONSOLE.

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

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

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

Q.V.	1 MIN 57 SECONDS
DEFAULT	12 MINS

5.0 DEVICE INFORMATION TABLES

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

```
CHANGE HW (L) ?

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

UNIT 0

DEVICE ADDRESS  (0) 172520 ? <ENTER THE ADDRESS OF THE
                              TSBA/TSDB REGISTER>

VECTOR (0) 224 ?        <ENTER ADDRESS OF INTERRUPT
                         VECTOR>
```

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

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

```
CHANGE SW (L) ?

INHIBIT ITERATIONS (L) N ?
```

6.0 TEST SUMMARIES

TEST 1: BUS RESET TEST

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

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

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

TEST 2: WRAP DATA - HIGH BYTE

THIS TEST VERIFIES OPERATION OF:

1. PART OF THE LSI-11 BUS INTERFACE SECTION OF THE M7196 MODULE: PART OF THE INPUT FILE (TSDB HIGH BYTE), PART OF THE OUTPUT FILE (TSSR HIGH BYTE AND TSBA, BOTH BYTES), PART OF THE DC005 TRANSCEIVER CIRCUITS (ADDRESS DECODER, BDAL DRIVERS, HIGH BYTE OF INTERNAL DAL BUS DRIVERS), AND BASIC PROGRAMMED I/O CONTROL SEQUENCES AND LOGIC;

2. PART OF 2901 MICROPROCESSOR ELEMENTS (Q-REGISTER, REGISTER 0, ROTATE AND NEGATE FUNCTIONS

3. Y AND SOURCE BUSES;

4. BASIC MICROPROGRAM SEQUENCES.

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

TEST 3: WRAP DATA - LOW BYTE

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

1. LOW BYTE OF THE TSDB INPUT FILE REGISTER,
2. LOW BYTE OF INTERNAL DAL BUS DRIVERS ON THE DC005 TRANSCEIVER CIRCUITS,
3. BASIC FUNCTIONING OF PARTS OF THE RAM.

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

TEST 4: RAM TEST

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

TEST 5: SECOND INITIALIZATION TEST

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

1. EACH RAM LOCATION AND 2901 REGISTERS 10 AND 11 ARE SET TO ALL 1'S BY USING WRITES INTO THE TSDB REGISTER (LOW BYTE AND MAINTENANCE MODE WORD WRITES).

2. THE CONTROLLER IS INITIALIZED AND THE VARIOUS CHECKS ON THE TSSR DESCRIBED IN INITIALIZATION TEST #1 ARE PERFORMED.

3. #1'S (377 OCTAL) ARE WRITTEN INTO THE LOW BYTE OF TSDB, WHICH SHOULD CAUSE RAM LOCATION 0 TO BE WRITTEN TO ALL 1'S SINCE 2901 REGISTERS 10 AND 11, SPECIFYING THE RAM ADDRESS, SHOULD BE 0. RAM LOCATION 0 IS VERIFIED BY WRITING A WORD OF ZEROS INTO THE TSDB. THE RESULTING LOW BYTE OF TSBA SHOULD CONTAIN ALL 1'S.

4. THE ENTIRE RAM IS SCANNED. LOCATION 0 SHOULD CONTAIN ALL 1'S AND THE REMAINING LOCATIONS, EXCEPT FOR THE MESSAGE BUFFER IMAGE AREA, SHOULD CONTAIN 0. DISCREPANCIES ARE REPORTED. AN ERROR AT THIS POINT IS MOST LIKELY DUE TO A ROM, PIPELINE OR SEQUENCER PROBLEM OR A TIMING PROBLEM.

TEST 6: COMMAND REJECT

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

TEST 7: WRITE CHARACTERISTICS

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

TEST 8: VOLUME CHECK

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

TEST 9: COMPLETION INTERRUPT

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

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

TEST 10: BASIC PACKET PROTOCOL

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

TEST 11: NON-TAPE MOTION COMMANDS

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

7.0 MAINTENANCE HISTORY

REVISION A - MARCH 1982

REVISION B - JUNE 1984
MINOR CHANGES FOR THE ORION CPU (11/??).
ELIMINATED THE MESSAGE DESCRIBING THE CPU TYPE.

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

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

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


002000                                  POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
002000                                  HEADER  CVTSA,8,0,655.,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      102                         .ASCII  /B/
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  045624                          .WORD   L$HARD
002020                          L$SPCP::                ;POINTER TO S.W. QUES.
002020  045756                          .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$DISPATCH
002042                          L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
002042  000000                          .WORD   0
002044                          L$ENVI::                ;FLAGS DESCRIBE HOW IT WAS SETUP
002044  000000                          .WORD   0
002046                          L$EXP1::                ;EXPANSION WORD
002046  000000                          .WORD   0
002050                          L$MREV::                ;SVC REV AND EDIT #
002050     003                          .BYTE   C$REVISION
002051     003                          .BYTE   C$EDIT
002052                          L$EF::                  ;DIAG. EVENT FLAGS
002052  000000                          .WORD   0
002054  000000                          .WORD   0
002056                          L$SPC::
002056  000000                          .WORD   0
002060                          L$DEVP::                ; POINTER TO DEVICE TYPE LIST
002060  003400                          .WORD   L$DVTYP
002062                          L$REPP::                ;PTR. TO REPORT CODE
002062  022620                          .WORD   L$RPT
002064                          L$EXP4::
002064  000000                          .WORD   0
002066                          L$EXP5::
002066  000000                          .WORD   0
002070                          L$AUT::                 ;PTR. TO ADD UNIT CODE
002070  022306                          .WORD   L$AU
002072                          L$DUT::                 ;PTR. TO DROP UNIT CODE
002072  022404                          .WORD   L$DU
002074                          L$LUN::                 ;LUN FOR EXERCISERS TO FILL
002074  000000                          .WORD   0
002076                          L$DESP::                ;POINTER TO DIAG. DESCRIPTION
002076  003406                          .WORD   L$DESC
002100                          L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
002100  104035                          EMT     E$LOAD
002102                          L$ETP::                 ;POINTER TO ERRTBL
002102  000000                          .WORD   0
002104                          L$ICP::                 ;PTR. TO  .IT CODE
002104  021512                          .WORD   L$INIT
002106                          L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  022572                          .WORD   L$CLEAN
002110                          L$ACP::                 ;PTR. TO AUTO CODE
002110  022512                          .WORD   L$AUTO
002112                          L$PRT::                 ;PTR. TO PROTECT TABLE
002112  021502                          .WORD   L$PROT
002114                          L$TEST::                ;TEST NUMBER
002114  000000                          .WORD   0
002116                          L$DLY::                 ;DELAY COUNT
002116  000000                          .WORD   0
002120                          L$HIME::                ;PTR. TO HIGH MEM
002120  000000                          .WORD   0
```

```
                                        .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  023402                          .WORD   T1
002126  023622                          .WORD   T2
002130  024320                          .WORD   T3
002132  025012                          .WORD   T4
002134  026346                          .WORD   T5
002136  027452                          .WORD   T6
002140  030724                          .WORD   T7
002142  034312                          .WORD   T8
002144  035216                          .WORD   T9
002146  040342                          .WORD   T10
002150  043454                          .WORD   T11


                                        .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
33
34 002174                       KT11    ..                      ;DEFINE MEMORY MANAGEMENT REGISTERS
                        .SBTTL  MEMORY MANAGEMENT DEFINITIONS
                        ;*KT11 VECTOR ADDRESS
000250                  MMVEC=  250
                        ;*KT11 STATUS REGISTER ADDRESSES
177572                  SR0=    177572
177574                  SR1=    177574
177576                  SR2=    177576
172516                  SR3=    172516
                        .IF NB
                        ;*USER "I" PAGE DESCRIPTOR REGISTERS
                        UIPDR0= 177600
                        UIPDR1= 177602
                        UIPDR2= 177604
                        UIPDR3= 177606
                        UIPDR4= 177610
                        UIPDR5= 177612
                        UIPDR6= 177614
                        UIPDR7= 177616
                        .IF NB
                        ;*USER "D" PAGE DESCRIPTOR REGISTORS
                        UDPDR0= 177620
                        UDPDR1= 177622
                        UDPDR2= 177624
                        UDPDR3= 177626
                        UDPDR4= 177630
                        UDPDR5= 177632
                        UDPDR6= 177634
                        UDPDR7= 177636
                        .ENDC
```

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

```
                                ;+
                                ;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 DEFINTTIONS
                                ;
                                ;-

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
```

```
                                ;+
                                ;BSELO 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          ;ITADO  - 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
004000                          S1.IIDENT       = BIT11         ;                IIDENT H
002000                          S1.ICER         = BIT10         ;                ICER H
001000                          S1.IFMK         = BIT9          ;                IFMK H
000400                          S1.IHER         = BIT8          ;                IHER H
000200                          S0.ISPEED       = BIT7          ;WORD #8 BYTE 0 ISPEED H
000100                          S0.IRDY         = BIT6          ;                IRDY L
000040                          S0.IONL         = BIT5          ;                IONL L
000020                          S0.ILDP         = BIT4          ;                ILDP L
000010                          S0.IDBY         = BIT3          ;                IDBY L
000004                          S0.IRWD         = BIT2          ;                IRWD L
000002                          S0.IFBY         = BIT1          ;                IFBY L
000001                          S0.IFPT         = BIT0          ;                IFPT L
```

```
                        .SBTTL  SPECIAL MACROS AND OPDEFS.

                ;+
                ;SAVE GENERAL REGS 1 TO 5
                ;-

                        .MACRO  SAVREG
                        JSR     R5,REGSAV
                        .ENDM

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

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

```
                                ;+
                                ;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
```

N3


```
                                        .SBTTL  TSTBLK    TEST DATA TABLE

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

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  177757                          .WORD   ↑CBIT4
003026  177737                          .WORD   ↑CBIT5
003030  177677                          .WORD   ↑CBIT6
003032  177577                          .WORD   ↑CBIT7
003034  177377                          .WORD   ↑CBIT8
003036  176777                          .WORD   ↑CBIT9
003040  175777                          .WORD   ↑CBIT10
003042  173777                          .WORD   ↑CBIT11
003044  167777                          .WORD   ↑CBIT12
003046  157777                          .WORD   ↑CBIT13
003050  137777                          .WORD   ↑CBIT14
003052  077777                          .WORD   ↑CBIT15
003054  125252                          .WORD   125252          ;ALTERNATING ONES, ZEROS
003056  052525                          .WORD   052525          ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
        003060                  TBLEND::
```

```
                                        .SBTTL  GLOBAL ENVIRONMENT STORAGE
                                ;
                                ;STORAGE FOR DEVICE REGISTERS
                                ;
003060  000000  100000  000000  DUMMY:  0,100000,0,0     ;DUMMY DEVICE REGISTERS...
003070  000000  000000  000000          0,0,0,0,0,0,0,0
                                                         ;...FOR MULTI-UNIT CHECKOUT.

003110  000000                  DUFLG::         .WORD   0               ;"DROPPED UNIT" FLAG.
                                                                        ;INHIBITS CODE IN "CLEAN-UP".
003112  000000                  NODEV::         .WORD   0               ;FLAG TO SAY NO DEVICE.

003114  000000                  TEMP1::         .WORD   0               ;SOME TEMP LOCATIONS.
003116  000000                  TEMP2::         .WORD   0
003120  000000                  XXCOMM::        .WORD   0               ;XXDP+ COMM BLOCK POINTER.
003122  000000                  FREE::          .WORD   0               ;1ST FREE MEMORY ADDRESS...
003124  000000                  FRESIZ::        .WORD   0               ;...AND SIZE (IN WORDS).
003126  000000                  FREEHI: .WORD   0               ;LAST WORD IN FREE SPACE
003130  000000                  KTFLG::         .WORD   0               ;KT11, MEM AVAIL FLAG -
                                                                        ;- .WORD        0 = <24K OR NO KT
                                                                        ;- NZ = >24K AND KT.
003132  000000                  KTENABLE::      .WORD   0               ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
003134  000000                  NXMFLG::        .WORD   0               ;SET IF WE CAN TEST CLEARED OTHERWISE
003136  000000                  NXMLO::         .WORD   0               ;NXM LO ADDRESS BITS
003140  000000                  NXMHI::         .WORD   0               ;NXM HI ADDRESS BITS FOR DAL'S 16-21
003142  000000                  T23A::          .WORD   0               ;11/23A FLAG
003144  000000                  T23B::          .WORD   0               ;11/23B FLAG
003146  000000                  T3BFLG::        .WORD   0               ;TEST 3B FLAG ↑0
003150  002000                  PST32W::        .WORD   2000            ;32W BLOCK ADDRESS FOR 32K START
003152  000000                  SIFLAG::        .WORD   0               ;
003154  000000                  BADDAT::        .WORD   0               ;ACTUAL DATA
003156  000000                  GDDAT::         .WORD   0               ;EXPECTED DATA
003160  000000                  LOOPFL::        .WORD   0
003162                          CTAB::                                  ;CONFIGURATION TABLES.
003162  000000                  CTABM::         .WORD   0               ;CONFIG WORK.
003164  000000                                  .WORD   0
003166  000000                                  .WORD   0
003170  000000                                  .WORD   0
003172  177777                                  .WORD   -1              ;END OF MEM TABLE.
003174                          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
                                ;
003174                          ERTABL:         .BLKW   64.
003374  000000                  ERTABE:         .WORD   0

003376  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
                                ;-
003400                                  DEVTYP  <TSV05>
003400                          L$DVTYP::
003400    124     123     126           .ASCIZ  /TSV05/
                                        .EVEN

                                ;*
                                ;TEST DESCRIPTION
                                ;-
003406                                  DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****>
003406                          L$DESC::
003406    052     052     052           .ASCIZ  /**** TSV05 LOGIC DIAGNOSTIC - REPLACE M7196 IF ERROR ****/
                                        .EVEN

                                ;*
                                ;BIT TO ASCII CONVERSION FOR TSSR REGISTER
                                ;-
003500  003540  003543  003547  TSSRBIT::       .WORD   1$,2$,3$,4$,5$,6$,7$,8$
003520  003601  003605  003611                  .WORD   9$,10$,11$,12$,13$,14$,15$,16$
003540    123     103     000   1$:     .ASCIZ  'SC'
003543    102     111     105   2$:     .ASCIZ  'BIE'
003547    123     103     105   3$:     .ASCIZ  'SCE'
003553    122     115     122   4$:     .ASCIZ  'RMR'
003557    116     130     115   5$:     .ASCIZ  'NXM'
003563    116     102     101   6$:     .ASCIZ  'NBA'
003567    102     111     124   7$:     .ASCIZ  'BIT9'
003574    102     111     124   8$:     .ASCIZ  'BIT8'
003601    123     123     122   9$:     .ASCIZ  'SSR'
003605    117     106     114   10$:    .ASCIZ  'OFL'
003611    102     111     124   11$:    .ASCIZ  'BIT5'
003616    102     111     124   12$:    .ASCIZ  'BIT4'
003623    102     111     124   13$:    .ASCIZ  'BIT3'
003630    102     111     124   14$:    .ASCIZ  'BIT2'
003635    102     111     124   15$:    .ASCIZ  'BIT1'
003642    102     111     124   16$:    .ASCIZ  'BIT0'
                                        .EVEN
003650    124     123     123   SFIERR: .ASCIZ  'TSSR ERROR AFTER SOFT INIT'
003703    124     123     123   SFHERR: .ASCIZ  'TSSR ERROR AFTER BUS RESET'
003736    040     040     116   NXR:    .ASCIZ  /  NON-EXISTANT DEVICE REGISTER/
003775    045     101     040   NXRX:   .ASCIZ  /%A  ADDRESS: %O6/
004016    045     101     040   TSSX:   .ASCII  /%A  TSBA,TSSR EXP'D: %O6%A,%O6%N/
004056    045     101     040           .ASCIZ  /%A  TSBA,TSSR REC'D: %O6%A,%O6/
004115    045     116     045   FUSI:   .ASCII  /%N%A/
004121    040     040     125   USI:    .ASCIZ  /  UNEXPECTED INTERRUPT/
004150    040     040     111   NSI:    .ASCIZ  /  INTERRUPT EXPECTED, NOT RECEIVED/
004213    045     116     045   FNOINTR:        .ASCII  /%N%A/
004217    040     040     116   NOINTR: .ASCIZ  /  NO INTERRUPT WAS GENERATED/
004254    040     040     111   IFAULT: .ASCIZ  /  INTERRUPT FAULT/
004276    045     101     040   INTX:   .ASCIZ  /%A  CPU PC: %O6%A  TSBA: %O6/
```

```
633 004333   040   040   042  NOINIT: .ASCIZ  /  "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
634 004405   040   040   042  NSINIT: .ASCIZ  /  "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
635 004455   040   040   042  BRINIT: .ASCIZ  /  "BUS-RESET" DIDN'T INITIALIZE THE DPU/
636 004525   000                NUL:    .ASCIZ  //
637 004526   045   116   000  NULCR:  .ASCIZ  /%N/
638 004531   045   101   040  EXPGOT: .ASCIZ  /%A EXP'D: %06%A, REC'D: %06/
639 004565   045   116   045  EXPGT2: .ASCIZ  /%N%A EXP'D: %06%A, %06%N%A REC'D: %0%A, %06/
640 004641   045   101   040  DUAD12: .ASCIZ  /%A REG(W) WRITTEN TO: %06%A REG(R) READ; EXP'D: %06%A, REC'D: %06/
641 004743   122   101   115  PKTRAM::        .ASCIZ  'RAM Contents Do Not Match Packet Sent'
642 005011   040   040   103  SCME:   .ASCIZ  /  CONFIG DOESN'T MATCH MFG. MASTER/
643 005054   127   122   111  WRTMSG: .ASCIZ  'WRITE CHARACTERISTICS Failed'
644 005111   124   123   123  WRTERR: .ASCIZ  'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
645 005204   124   123   123  RDERR:  .ASCIZ  'TSSR Incorrect After READ Command, More Bits Set Than SSR'
646 005276   106   101   124  SCHERR: .ASCIZ  'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
647 005370   105   122   122  RETERR: .ASCIZ  'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
648 005456   045   116   045  NOMEM:  .ASCIZ  '%N%A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. ******%N'
649 005552   045   116   045  M8186:  .ASCIZ  '%N%A ****************** 11/23A SYSTEM ********************%N'
650 005643   045   116   045  M8189:  .ASCIZ  '%N%A ****************** 11/23B SYSTEM ********************%N'
651                                          .EVEN
652                                          .SBTTL  GLOBAL ERROR REPORT SECTION
653
654                                  ;+
655                                  ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
656                                  ; CALLS THAT ARE USED IN MORE THAN ONE TEST.
657                                  ; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
658                                  ;-
659 005734                                   BGNMSG  NXRERR          ;NON-EXISTANT DEVICE REGISTER.
    005734                           NXRERR::
660 005734                                   PRINTX  #NXRX,NODEV     ;NODEV = NEXM ADDRESS.
    005734  013746  003112                   MOV     NODEV,-(SP)
    005740  012746  003775                   MOV     #NXRX,-(SP)
    005744  012746  000002                   MOV     #2,-(SP)
    005750  010600                           MOV     SP,R0
    005752  104415                           TRAP    C$PNTX
    005754  062706  000006                   ADD     #6,SP
661 005760  004737  005766                   JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
662 005764                                   ENDMSG
    005764                           L10002:
    005764  104423                           TRAP    C$MSG
663                                  ;
664                                  ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
665                                  ; TO ANY OF THE ABOVE ERROR SIGNATURES.
666                                  ;
667 005766  005727                   EXTEND: TST     (PC)+
668 005770  000000                   EXTA:   0                       ; 0 = NO EXTENSION.
669 005772  001402                           BEQ     1$
670 005774  004777  177770                   JSR     PC,@EXTA        ; APPEND EXTENSION TEXT.
671 006000                           1$:     PRINTX  #NULCR          ; PRINT A BLANK LINE
    006000  012746  004526                   MOV     #NULCR,-(SP)
    006004  012746  000001                   MOV     #1,-(SP)
    006010  010600                           MOV     SP,R0
    006012  104415                           TRAP    C$PNTX
    006014  062706  000004                   ADD     #4,SP
672 006020  000207                           RTS     PC
```

```
                                        .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
                                ;
                                ;

006022                          PRITSSR:
006022                                  SAVREG                          ;SAVE GENERAL REGISTERS
006026  010104                          MOV     R1,R4                   ;SAVE THE TSSR CONTENTS
006030                                  PRINTB  #TSSRFOR,R4             ;PRINT THE CONTENTS OF TSSR
006030  010446                          MOV     R4,-(SP)
006032  012746  006413                  MOV     #TSSRFOR,-(SP)
006036  012746  000002                  MOV     #2,-(SP)
006042  010600                          MOV     SP,R0
006044  104414                          TRAP    C$PNTB
006046  062706  000006                  ADD     #6,SP
006052  010400                          MOV     R4,R0                   ;GET TSSR BACK FOR CHKAMB
006054  004737  016044                  JSR     PC,CHKAMB               ;ARE CONTENTS AMBIGUOUS ?
006060  103410                          BCS     5$                      ;BRANCH IF NOT
006062                                  PRINTX  #AMBTSSR                ;SHOW CONTENTS ARE AMBIGUOUS
006062  012746  006633                  MOV     #AMBTSSR,-(SP)
006066  012746  000001                  MOV     #1,-(SP)
006072  010600                          MOV     SP,R0
006074  104415                          TRAP    C$PNTX
006076  062706  000004                  ADD     #4,SP
006102  010403                  5$:     MOV     R4,R3                   ;CONTENTS OF TSSR
006104  042703  001476                  BIC     #HIADDR!FATERR!TERCLS,R3        ;CLEAR ALL MULTIPLE BIT FIELDS
006110  001434                          BEQ     20$                     ;NO BITS ARE SET
006112  012702  002630                  MOV     #TMPBFR,R2              ;TEMPORARY ASCII BUFFER
006116  012701  003500                  MOV     #TSSRBIT,R1             ;ASCII EQUIVALENT OF BITS
006122  005703                  10$:    TST     R3                      ;REMAINING BITS TO CONVERT
006124  001413                          BEQ     15$                     ;BRANCH WHEN ALL ARE DONE
006126  000241                          CLC                             ;CLEAR CARRY FOR SHIFT
006130  006103                          ROL     R3                      ;SHIFT NEXT BIT TO CARRY
006132  103006                          BCC     13$                     ;BRANCH IF BIT NOT SET
006134  011100                          MOV     (R1),R0                 ;POINTER TO BIT DEFINITION
006136  112022                  1$:     MOVB    (R0)+,(R2)+             ;MOVE ASCIZ TO BUFFER
006140  001376                          BNE     11$                     ;MOVE ALL BITS
006142  112762  000054  177777          MOVB    #',,-1(R2)              ;INSERT A COMMA TO TERMINATE
006150  005721                  13$:    TST     (R1)+                   ;POINT TO NEXT DESCRIPTION
006152  000763                          BR      10$                     ;GET THE REMAINING BITS
006154  105042                  15$:    CLRB    -(R2)                   ;TERMINATE THE LINE
006156                                  PRINTX  #TSSDEF,#TMPBFR         ;PRINT THE BIT DEFINITIONS
006156  012746  002630                  MOV     #TMPBFR,-(SP)
006162  012746  006604                  MOV     #TSSDEF,-(SP)
```

```
        006166  012746  000002                  MOV     #2,-(SP)
        006172  010600                          MOV     SP,R0
        006174  104415                          TRAP    C$PNTX
        006176  062706  000006                  ADD     #6,SP
    718
    719 006202  010403                  20$:    MOV     R4,R3           ;GET THE TSSR CONTENTS
    720 006204  042703  177761                  BIC     #^CTERCLS,R3    ;CLEAR ALL BUT TERMINATION
    721 006210  016303  006674                  MOV     TCOCOD(R3),R3   ;GET THE TERMINATION CODE MEANING
    722 006214                                  PRINTX  #TCOASC,R3      ;PRINT THE TERMINATION CODE
        006214  010346                          MOV     R3,-(SP)
        006216  012746  006474                  MOV     #TCOASC,-(SP)
        006222  012746  000002                  MOV     #2,-(SP)
        006226  010600                          MOV     SP,R0
        006230  104415                          TRAP    C$PNTX
        006232  062706  000006                  ADD     #6,SP
    723 006236  010403                          MOV     R4,R3           ;TSSR CONTENTS AGAIN
    724 006240  042703  177717                  BIC     #^CFATERR,R3    ;CLEAR ALL BUT FATAL TERMINATION
    725 006244  001416                          BEQ     25$             ;DON'T PRINT IF ZERO
    726 006246  006203                          ASR     R3
    727 006250  006203                          ASR     R3
    728 006252  006203                          ASR     R3              ;ALINE TERMINATION CODE FOR INDEX
    729 006254  016303  007234                  MOV     TSFCOD(R3),R3   ;GET THE FATAL TERMINATION CODE
    730 006260                                  PRINTX  #TFCASC,R3      ;PRINT THE FATAL TERMINATION CODE
        006260  010346                          MOV     R3,-(SP)
        006262  012746  006535                  MOV     #TFCASC,-(SP)
        006266  012746  000002                  MOV     #2,-(SP)
        006272  010600                          MOV     SP,R0
        006274  104415                          TRAP    C$PNTX
        006276  062706  000006                  ADD     #6,SP
    731 006302  042704  176377          25$:    BIC     #^CHIADDR,R4    ;CLEAR ALL BUT EXTENDED ADDRESS
    732 006306  001411                          BEQ     30$             ;DON'T PRINT IF ZERO
    733 006310                                  PRINTX  #TEXASC,R4      ;PRINT THE EXTENDED ADDRESS BITS
        006310  010446                          MOV     R4,-(SP)
        006312  012746  006433                  MOV     #TEXASC,-(SP)
        006316  012746  000002                  MOV     #2,-(SP)
        006322  010600                          MOV     SP,R0
        006324  104415                          TRAP    C$PNTX
        006326  062706  000006                  ADD     #6,SP
    734 006332  013703  002176          30$:    MOV     EPRTSW,R3               ;PRINT MEAASGE BUFFER ADDRESS
    735 006336                                  PRINTX  R3                      ;PRINT PROPER MESSAGE
        006336  010346                          MOV     R3,-(SP)
        006340  012746  000001                  MOV     #1,-(SP)
        006344  010600                          MOV     SP,R0
        006346  104415                          TRAP    C$PNTX
        006350  062706  000004                  ADD     #4,SP
    736 006354  000207                          RTS     PC              ;RETURN TO CALLER
```

```
006356                          EPRT2:
006356     045     116     045  EPRT1:   .ASCIZ  '¢N¢A *****REPLACE M7196*****'
006413     045     116     045  TSSRFOR:         .ASCIZ  '¢N¢A TSSR = ¢06'
006433     045     116     045  TEXASC:          .ASCIZ  '¢N¢A Extended Address Bits = ¢06'
006474     045     116     045  TCOASC:          .ASCIZ  '¢N¢A Termination Class Code = ¢T'
006535     045     116     045  TFCASC:          .ASCIZ  '¢N¢A Fatal Termination Class Code = ¢T
006604     045     116     045  TSSDEF:          .ASCIZ  '¢N¢A TSSR Bits Set: ¢T'
006633     045     116     045  AMBTSSR:         .ASCIZ  '¢N¢A TSSR Contents Are Ambiguous'
                                        .EVEN
006674  006714  006737  006765  TCOCOD: .WORD   1$,2$,3$,4$,5$,6$,7$,8$
006714     116     157     162  1$:     .ASCIZ  'Normal Termination'
006737     124     145     162  2$:     .ASCIZ  'Termination Condition'
006765     124     141     160  3$:     .ASCIZ  'Tape Status Alert'
007007     106     165     156  4$:     .ASCIZ  'Function Reject'
007027     122     145     143  5$:     .ASCIZ  'Recoverable Error - Tape Position One Record Down'
007111     122     145     143  6$:     .ASCIZ  'Recoverable Error - Tape Was Not Moved'
007160     125     156     162  7$:     .ASCIZ  'Unrecoverable Error'
007204     106     141     164  8$:     .ASCIZ  'Fatal Controller Error'
                                        .EVEN

007234  007244  007300  007311  TSFCOD: .WORD   1$,2$,3$,4$
007244     111     156     164  1$:     .ASCIZ  'Internal Diagnostic Failure'
007300     122     145     163  2$:     .ASCIZ  'Reserved'
007311     102     165     163  3$:     .ASCIZ  'Bus Interface or Sanity Check Error'
007355     122     145     163  4$:     .ASCIZ  'Reserved'
                                        .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.
                                ;-

007366                          PRIPKT::
007366                                  SAVREG                          ;SAVE THE REGISTERS
007372  010005                          MOV     R0,R5                   ;SAVE NO. OF WORDS IN PACKET
007374  005737  003132                  TST     KTENABLE                ;ABOVE 28K UNDER TEST?
007400  001001                          BNE     10$                     ;BR IF YES
007402  005003                          CLR     R3                      ;SET HIGH ORDER ADDRESS TO 0
007404  010301                  10$:    MOV     R3,R1                   ;COPY HIGH ORDER ADDRESS
007406  010400                          MOV     R4,R0                   ;GET LOWER ADDRESS
007410  006100                          ROL     R0                      ;SHIFT BIT 15 INTO C BIT
007412  006101                          ROL     R1                      ;AND INTO HIGH ORDER.
007414                                  PRINTB  #PKTADD,R1,R4           ;PRINT PACKET ADDRESS
007414  010446                          MOV     R4,-(SP)
007416  010146                          MOV     R1,-(SP)
007420  012746  007552                  MOV     #PKTADD,-(SP)
007424  012746  000003                  MOV     #3,-(SP)
007430  010600                          MOV     SP,R0
007432  104414                          TRAP    C$PNTB
007434  062706  000010                  ADD     #10,SP
007440  010300                  15$:    MOV     R3,R0                   ;GET HIGH ORDER ADDRESS
007442  001404                          BEQ     20$                     ;BR IF NOT ABOVE 28K.
007444  010401                          MOV     R4,R1                   ;GET LOW ORDER ADDRESS
007446  004737  017316                  JSR     PC,SETMAP               ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
007452  010004                          MOV     R0,R4                   ;GET RETURNED PAR6 ADDRESS BIAS
007454  005001                  20$:    CLR     R1                      ;SAVE WORD NUMBER
007456  012402                  25$:    MOV     (R4)+,R2                ;GET PACKET CONTENTS
007460                                  PRINTB  #PKTFRM,R1,R2           ;PRINT THE DATA
007460  010246                          MOV     R2,-(SP)
007462  010146                          MOV     R1,-(SP)
007464  012746  007514                  MOV     #PKTFRM,-(SP)
007470  012746  000003                  MOV     #3,-(SP)
007474  010600                          MOV     SP,R0
007476  104414                          TRAP    C$PNTB
007500  062706  000010                  ADD     #10,SP
007504  005201                          INC     R1                      ;NEXT WORD NUMBER
007506  020105                          CMP     R1,R5                   ;DONE ALL PACKET WORDS?
007510  002762                          BLT     25$                     ;LOOP TILL ALL DONE
007512  000207                          RTS     PC                      ;RETURN

007514     045     116     045  PKTFRM: .ASCIZ  '<N><A Packet Word #<D1><A = <06
007552     045     116     045  PKTADD: .ASCIZ  '<N><A Packet Address = <01><05'
                                        .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
                                ;
007610                          PRIBXOR::
007610                                  SAVREG                          ;SAVE THE REGISTERS
007614  010203                          MOV     R2,R3                   ;EXPECTED DATA
007616                                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007626  012700  177400                  MOV     #^C<377>,R0             ;BYTE MASK
007632  040001                          BIC     R0,R1                   ;SAVE LOW BYTE RECV
007634  040002                          BIC     R0,R2                   ;SAVE LOW BYTE EXPD
007636  040003                          BIC     R0,R3                   ;SAVE LOW BYTE XOR
007640                                  PRINTB  #XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
007640  010346                          MOV     R3,-(SP)
007642  010146                          MOV     R1,-(SP)
007644  010246                          MOV     R2,-(SP)
007646  012746  007672                  MOV     #XORBFOR,-(SP)
007652  012746  000004                  MOV     #4,-(SP)
007656  010600                          MOV     SP,R0
007660  104414                          TRAP    C$PNTB
007662  062706  000012                  ADD     #12,SP
007666  010300                          MOV     R3,R0                   ;R0 HAS XOR ON RETURN
007670  000207                          RTS     PC                      ;RETURN TO CALLER

007672     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
                                ;-
007740                          PRIXOR::
007740                                  SAVREG                          ;SAVE THE REGISTERS
007744  010203                          MOV     R2,R3                   ;EXPECTED DATA
007746                                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007756                                  PRINTB  #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
```

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

```
                                        .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
                                ;
                                ;-

010056                          PRIEQU:
010056                                  SAVREG                          ;SAVE THE REGISTERS
010062  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
                                ;
                                ;-
010064                          PRIRAM:
010064                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
010070                                  PRINTB  #RAMFOR,R4              ;PRINT RAM ADDRESS IN ERROR
010070  010446                          MOV     R4,-(SP)
010072  012746  010114                  MOV     #RAMFOR,-(SP)
010076  012746  000002                  MOV     #2,-(SP)
010102  010600                          MOV     SP,R0
010104  104414                          TRAP    C$PNTB
010106  062706  000006                  ADD     #6,SP
010112  000207                          RTS     PC                      ;RETURN

010114     045     116     045  RAMFOR: .ASCIZ  '%N%A CONTROLLER RAM ADDRESS = %O6'
                                        .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
                                ;
                                ;
010156                          PRIADD:
010156                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
010162  013700  002234                  MOV     ERRHI,R0                ;GET HIGH ADDRESSS
010166  013701  002236                  MOV     ERRLO,R1                ;GET LOW ADDRESS
010172  010102                          MOV     R1,R2                   ;COPY LOW ADDRESS
010174  006101                          ROL     R1                      ;SHIFT BIT 15 TO C BIT
010176  006100                          ROL     R0                      ;SHIFT INTO HIGH ORDER
010200                                  PRINTB  #PRIA0,R0,R2            ;PRINT MEMORY ADDRESS IN ERROR
010200  010246                          MOV     R2,-(SP)
010202  010046                          MOV     R0,-(SP)
010204  012746  010226                  MOV     #PRIA0,-(SP)
010210  012746  000003                  MOV     #3,-(SP)
010214  010600                          MOV     SP,R0
010216  104414                          TRAP    C$PNTB
010220  062706  000010                  ADD     #10,SP
010224  000207                          RTS     PC                      ;RETURN

010226     045     116     045  PRIA0:  .ASCIZ  '%N%A MEMORY ERROR ADDRESS = %O1%O5'
                                        .EVEN

                                        .SBTTL  PRITADD - PRINT MEMORY TEST ADDRESS
                                ;+
                                ;
                                ;PRINT MEMORY ADDRESS
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                                ;
                                ; IMPLICIT INPUTS
                                ;
                                ;       ERRHI   - HIGH ORDER ADDRESS
                                ;       ERRLO   - LOW ORDER ADDRESS
                                ;
                                ;-
010272                          PRITADD:
010272                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
010276  013702  002234                  MOV     ERRHI,R2                ;GET HIGH ADDRESSS
010302  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
010306                                  PRINTB  #PRIT0,R1               ;PRINT MEMORY ADDRESS LOW IN ERROR
010306  010146                          MOV     R1,-(SP)
010310  012746  010354                  MOV     #PRIT0,-(SP)
010314  012746  000002                  MOV     #2,-(SP)
010320  010600                          MOV     SP,R0
010322  104414                          TRAP    C$PNTB
```

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

```
                                        .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:
                                ;
                                ;
                                ;

010464                          SPACE::
010454                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
010470  012737  000764  010660          MOV     #500.,SDELAY            ;SET UP DELAY
010476  012737  140010  010650          MOV     #140010,80$             ;SET UP COMMAND, SPACE FORWARD
010504  005703                          TST     R3                      ;CHECK FOR DIRECTION
010506  100403                          BMI     5$                      ;BR, IF REVERSE INDICATED
010510  010337  010652                  MOV     R3,90$                  ;LOAD UP NUMBER OF RECORDS TO SPACE
010514  000407                          BR      10$                     ;GO DO COMMAND
010516  042703  100000          5$:     BIC     #BIT15,R3               ;CLEAR DIRECTION BIT
010522  010337  010652                  MOV     R3,90$                  ;LOAD UP NUMBER OF RECORDS TO SPACE
010526  052737  000400  010650          BIS     #BIT8,80$               ;SET REVERSE BIT IN COMMAND PACKET
010534  012704  010050          10$:    MOV     #80$,R4                 ;SET UP R4 WITH PACKET ADDRESS
010540  010465  000000                  MOV     R4,TSDB(R5)             ;SEND OUT COMMAND
010544  004737  016250          15$:    JSR     PC,WAITF                ;WAIT FOR SSR
010550  103420                          BCS     20$                     ;BR, IF SSR IS SET AND OK
010552                                  DELAY   250             ;DELAY ABOUT .25 SECONDS
010552  012727  000250                  MOV     #250,(PC)+
010556  000000                          .WORD   0
010560  013727  002116                  MOV     L$DLY,(PC)+
010564  000000                          .WORD   0
010566  005367  177772                  DEC     -6(PC)
010572  001375                          BNE     . 4
```

```
        010574  005367  177756                  DEC     -22(PC)
        010600  001367                          BNE     .-20
   1018 010602  005337  010660                  DEC     SDELAY                  ;BUMP DELAY COUNTER DOWN
   1019 010606  001356                          BNE     15$                     ;BR, IF MORE DELAY
   1020 010610  000411                          BR      60$                     ;BR IF TROUBLE CARRY = CLEAR
   1021 010612  016501  000002          20$:    MOV     TSSR(R5),R1             ;READ TSSR
   1022 010616  012702  000200                  MOV     #SSR,R2                 ;SET UP EXPECTED
   1023 010622  020201                  25$:    CMP     R2,R1                   ;ARE THEY OK
   1024 010624  001401                          BEQ     40$                     ;BR, IF EQUAL = OK
   1025 010626  000402                          BR      60$                     ;TROUBLE EXIT
   1026 010630  000261                  40$:    SEC                             ;SET CARRY NO TROUBLE
   1027 010632  000401                          BR      70$                     ;EXIT
   1028 010634  000241                  60$:    CLC                             ;CARRY CLEAR = ERROR
   1029 010636                          70$:
   1030 010636  010400                          MOV     R4,R0                   ;PASS PACKET ADDRESS
   1031 010640  000207                          RTS     PC                      ;RETURN
   1032                                 ;
   1033                                 ;
   1034                                 ;
   1035                                 ;PACKET FOR SPACE COMMAND
   1036                                 ;
   1038         010650                          .=<.+10>&177770
   1040                                 ;
   1041                                 ;COMMAND WORD
   1042 010650  000000                  80$:    .WORD
   1043                                 ;NUMBER OF RECORDS TO BE SPACED OVER WORD
   1044 010652  000000                  90$:    .WORD
   1045 010654  000000                          .WORD
   1046 010656  000000                          .WORD
   1047 010660  000000                  SDELAY: .WORD   0                       ;DELAY COUNTER
   1048                                         .EVEN
   1049                                         .SBTTL  WRTCHR  - WRITE CHARACTERISTICS COMMAND
```

```
                                        ;+
                                        ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
                                        ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
                                        ;
                                        ;INPUT:
                                        ;       R4      ADDRESS OF PACKET FROM TEST
                                        ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                        ;       REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
                                        ;
                                        ;OUTPUT:
                                        ;       R0      TSSR CONTENTS
                                        ;       CARRY   SET - WRITE CHARACTERISTICS COMMAND OK
                                        ;               CLR - WRITE CHARACTERISTICS FAILED
                                        ;
                                        ;IMPLICIT OUTPUT:
                                        ;
                                        ;       MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
                                        ;       SOFTWARE SWITCHES SET AS FOLLOWS:
                                        ;               EXTFEA = EXTENDED FEATURES PRESENT
                                        ;               BENBSW = BUFFER ENABLE SWITCH ON OR OFF
                                        ;
                                        ;SIDE EFFECTS:
                                        ;-
010662                                  WRTCHR::
010662                                          SAVREG                          ;SAVE THE GENERAL REGISTERS
010666  005037  002226                          CLR     BENBSW          ;CLEAR BUFFER ENABLE SWITCH
010672  005037  002224                          CLR     EXTFEA                  ;CLEAR EXTENDED FEATURES SW SWITCH
010676  010465  000000                  10$:    MOV     R4,TSDB(R5)             ;SEND OUT COMMAND
010702  004737  016336                          JSR     PC,CHKTSSR              ;WAIT FOR SSR
010706  103401                                  BCS     20$                     ;BR, IF SSR IS SET AND OK
010710  000435                                  BR      60$                     ;BR IF TROUBLE CARRY = CLEAR
010712  016501  000002                  20$:    MOV     TSSR(R5),R1             ;READ TSSR
010716  012702  000200                          MOV     #SSR,R2                 ;SET UP EXPECTED
010722  032701  000100                          BIT     #OFL,R1                 ;WAS OFF LINE SET IN TSSR
010726  001402                                  BEQ     25$                     ;BR, IF NO OFL SET
010730  052702  000100                          BIS     #OFL,R2                 ;MAKE THEM LOOK ALIKE
010734  020201                          25$:    CMP     R2,R1                   ;ARE THEY OK
010736  001401                                  BEQ     40$                     ;BR, IF EQUAL = OK
010740  000421                                  BR      60$                     ;TROUBLE EXIT
010742  062704  000010                  40$:    ADD     #8.,R4                  ;POINT TO WRT CHARA DATA PACKET
010746  011403                                  MOV     (R4),R3                 ;GET ADDRESS OF MESSAGE BUFFER
010750  032763  000200  000012                  BIT     #X2.EXTF,XST2(R3)       ;EXTENDED FEATURES BIT SET?
010756  001402                                  BEQ     45$                     ;BR IF NO
010760  005237  002224                          INC     EXTFEA                  ;SET EXTENDED FEATURES SW SWITCH
010764                                  45$:
010764  032763  000100  000012                  BIT     #X2.BUFE,XST2(R3)       ;BUFFER ENABLE SWITCH SET
010772  001402                                  BEQ     50$                     ;BR, IF SWITCH NOT SET
010774  005237  002226                          INC     BENBSW                  ;SET SOFTWARE SWITCH FOR ENABLED
011000                                  50$:
011000  000261                                  SEC                             ;SET CARRY NO TROUBLE
011002  000401                                  BR      70$                     ;EXIT
011004  000241                          60$:    CLC                             ;CARRY CLEAR = ERROR
011006  016500  000002                  70$:    MOV     TSSR(R5),R0             ;RETURN TSSR CONTENTS
011012  000207                                  RTS     PC                      ;RETURN
```

```
                                        .SBTTL  REWIND  - POSITION TAPE (REWIND) COMMAND
                                ;+
                                ;
                                ;THIS ROUTINE WILL REWIND THE SELECTED TAPE.
                                ;
                                ;       CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
                                ;                TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
                                ;                SSR TO SET IN THE TSSR
                                ;
                                ;
                                ;CALLING SEQUENCE:
                                ;
                                ;       DO A SOFT INIT
                                ;       DO A WRITE CHARACTERISTICS
                                ;       JSR     PC,REWIND
                                ;
                                ;INPUT:
                                ;
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;
                                ;
                                ;OUTPUT
                                ;
                                ;       R0      THE CONTENTS OF R4 IS PASSED TO R0
                                ;
                                ;
                                ;-
011014                          REWIND::
011014                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011020  012704  011110                  MOV     #RWPACK,R4              ;GET PACKET ADDRESS
011024  010465  000000                  MOV     R4,TSDB(R5)             ;SEND PACKET ADDRESS TO EXECUTE
011030  012703  000550                  MOV     #360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
011034  004737  016250          10$:    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
011040  103417                          BCS     20$                     ;LEAVE WHEN SSR IS SET
011042                                  DELAY   250.                    ;WAIT FOR .25 SECONDS
011042  012727  000372                  MOV     #250.,(PC)+
011046  000000                          .WORD   0
011050  013727  002116                  MOV     L$DLY,(PC)+
011054  000000                          .WORD   0
011056  005367  177772                  DEC     -6(PC)
011062  001375                          BNE     .-4
011064  005367  177756                  DEC     -22(PC)
011070  001367                          BNE     .-20
011072  005303                          DEC     R3                      ;BUMP COUNTER DOWN
011074  001357                          BNE     10$                     ;KEEP GOING
011076  000241                          CLC                             ;CLEAR CARRY TO SET ERROR
011100  010400                  20$:    MOV     R4,R0                   ;PASS THE PACKET ADDRESS
011102  000207                          RTS     PC                      ;RETURN

        011110                          .=<.+10>&177770
011110                          RWPACK:
011110  102010                          .WORD   102010                  ;POSTION COMMAND (REWIND)
011112  000000                          .WORD   0                       ;NOT USED
```

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

011114                          CKRAM::
011114                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
011120  012701  002240                  MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
011124  012702  000201                  MOV     #RMPKTBEG,R2            ;BYTE ADDRESS OF FIRST RAM DATA
011130  005003                          CLR     R3                      ;CLEAR THE ERROR FLAG
011132  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
011136  112765  000000  000000          MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
011144  004737  016336          10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011150  010265  000000                  MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
011154  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011160  116511  000000                  MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
011164  122124                          CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
011166  001401                          BEQ     20$                     ;BRANCH IF OK
011170  005203                          INC     R3                      ;SET ERROR FLAG
011172  005202          20$:            INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
011174  020227  000210                  CMP     R2,#RMPKTEND            ;REACHED END YET ?
011200  003761                          BLE     10$                     ;BRANCH TILL ALL READ
011202  005703                          TST     R3                      ;WAS AN ERROR FOUND ?
011204  001402                          BEQ     30$                     ;BRANCH IF NOT
011206  000241                          CLC                             ;CLEAR CARRY TO SHOW ERROR
011210  000401                          BR      50$                     ;AND EXIT
011212  000261          30$:            SEC                             ;SHOW GOOD COMPARE
011214  012737  000010  002300  50$:    MOV     #8.,RAMSIZ              ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
011222  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
                                ;-
011224                          CKRAM2::
011224                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
011230  012701  002240                  MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
011234  012702  000167                  MOV     #RMCHBEG,R2             ;BYTE ADDRESS OF FIRST RAM DATA
011240  005003                          CLR     R3                      ;CLEAR THE ERROR FLAG
011242  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
011246  112765  000000  000000          MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
011254  004737  016336          10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011260  010265  000000                  MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
011264  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
011270  116511  000000                  MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
011274  122124                          CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
011276  001401                          BEQ     20$                     ;BRANCH IF OK
011300  005203                          INC     R3                      ;SET ERROR FLAG
011302  005202          20$:            INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
011304  012737  000010  002300          MOV     #8.,RAMSIZ              ;ASSUME EXTFEA NOT SET
011312  005737  002224                  TST     EXTFEA                  ;IS THE SOFTWARE EXTENDED FEATURES SET
011316  001407                          BEQ     25$                     ;BR. IF NOT SET
011320  012737  000012  002300          MOV     #10.,RAMSIZ             ;SET RAMSIZ FOR EXTEND FEATURES
011326  020227  000200                  CMP     R2,#RMCHEND             ;AT END OF EXTENDED BUFFER
011332  003750                          BLE     10$                     ;BR. IF NOT AT END YET
011334  000403                          BR      27$                     ;AT END BRANCH
011336  020227  000176          25$:    CMP     R2,#RMCHEND-2           ;REACHED END YET ?
011342  003744                          BLE     10$                     ;BRANCH TILL ALL READ
011344  005703                  27$:    TST     R3                      ;WAS AN ERROR FOUND ?
011346  001402                          BEQ     30$                     ;BRANCH IF NOT
011350  000241                          CLC                             ;CLEAR CARRY TO SHOW ERROR
011352  000401                          BR      50$                     ;AND EXIT
011354  000261                  30$:    SEC                             ;SHOW GOOD COMPARE
011356  000207                  50$:    RTS     PC                      ;RETURN
```

```
                                        .SBTTL  CKMSG   - COMPARE WRITE CHAR. MESSAGE BUFFERS
                                ;+
                                ;
                                ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
                                ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
                                ;ERROR PRINT ROUTINES.
                                ;
                                ;INPUT:
                                ;
                                ;       R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
                                ;       R2      EXPD MESSAGE BUFFER ADDRESS
                                ;OUTPUT:
                                ;
                                ;       CARRY   SET - MESSAGE BUFFERS MATCH
                                ;               CLR -MESSAGE BUFFERS DON'T MATCH
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       EXPMSG          BUFFER IS SET TO EXPD DATA
                                ;       RECMSG          BUFFER IS SET TO RECV DATA
                                ;       RCVHIADD        SET TO HIGH ORDER ADDRESS OF RECV
                                ;       RCVLOADD        SET TO LOW ORDER ADDRESS OF RECV
                                ;
                                ;-
011360                          CKMSG::
011360                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011364  010037  002302                  MOV     R0,RCVHIADD             ;SAVE RECV HIGH ADDRESS
011370  010137  002304                  MOV     R1,RCVLOAD              ;SAVE RECV LOW ADDRESS
011374  005737  003132                  TST     KTENABLE                ;TESTING ABOVE 28K?
011400  001403                          BEQ     10$                     ;BR IF NO
011402  004737  017316                  JSR     PC SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
011406  010001                          MOV     R0,R1                   ;GET RETURNED ADDRESS BIASED TO PAR6
011410  005004                  10$:    CLR     R4                      ;WORD IN BUFFER
011412  005003                          CLR     R3                      ;CLEAR ERROR SEEN FLAG
011414  010205                          MOV     R2,R5                   ;GET EXPD BUFFER ADDRESS
011416  011264  002320          15$:    MOV     (R2),EXPMSG(R4)         ;SAVE EXPD FOR ERROR REPORT
011422  011164  002464                  MOV     (R1),RECMSG(R4)         ;SAVE RECV FOR ERROR REPORT
011426  022221                          CMP     (R2)+,(R1)+             ;EXPD EQUAL RECV?
011430  001401                          BEQ     25$                     ;BR IF YES
011432  005203                          INC     R3                      ;SET ERROR SEEN FLAG
011434  062704  000002          25$:    ADD     #2,R4                   ;POINT TO NEXT WORD ADDRESS
011440  020427  000014                  CMP     R4,#14                  ;DONE FIRST 7 WORDS?
011444  003764                          BLE     15$                     ;BR IF NO
011446  032765  000200  000012          BIT     #X2.EXTF,XST2(R5);IS EXTENDED FEATURES SET IN EXPD?
011454  001403                          BEQ     50$                     ;BR IF NO
011456  020427  000016                  CMP     R4,#16                  ;DONE EXTENDED FEATURES WORD?
011462  003755                          BLE     15$                     ;BR IF NO
011464  005703                  50$:    TST     R3                      ;ANY ERRORS SEEN?
011466  001402                          BEQ     55$                     ;BR IF NO
011470  000241                          CLC                             ;SET FAILURE
011472  000401                          BR      60$                     ;
011474  000261                  55$:    SEC                             ;SET SUCCESS
011476  000207                  60$:    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
                                ;-
011500                          CKMSG2::
011500                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011504  020327  000144                  CMP     R3,#RECMSG-EXPMSG;BGD IS COUNT ABOVE MAX ALLOWED?
011510  003412                          BLE     5$                      ;BGD BR IF NO
011512  012703  000144                  MOV     #RECMSG-EXPMSG,R3;BGD
011516                                  PRINTF  #DEBUGMSG               ;BGD
011516  012746  011632                  MOV     #DEBUGMSG,-(SP)
011522  012746  000001                  MOV     #1,-(SP)
011526  010600                          MOV     SP,R0
011530  104417                          TRAP    C$PNTF
011532  062706  000004                  ADD     #4,SP
011536  010037  002302          5$:     MOV     R0,RCVHIADD             ;SAVE RECV HIGH ADDRESS
011542  010137  002304                  MOV     R1,RCVLOAD              ;SAVE RECV LOW ADDRESS
011546  005737  003132                  TST     KTENABLE                ;TESTING ABOVE 28K?
011552  001403                          BEQ     10$                     ;BR IF NO
011554  004737  017316                  JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
011560  010001                          MOV     R0,R1                   ;GET RETURNED ADDRESS BIASED TO PAR6
011562  005004                  10$:    CLR     R4                      ;WORD IN BUFFER
011564  005005                          CLR     R5                      ;CLEAR ERROR SEEN FLAG
011566  111264  002320          15$:    MOVB    (R2),EXPMSG(R4)         ;SAVE EXPD FOR ERROR REPORT
011572  111164  002464                  MOVB    (R1),RECMSG(R4)         ;SAVE RECV FOR ERROR REPORT
011576  122221                          CMPB    (R2)+,(R1)+             ;EXPD EQUAL RECV?
011600  001401                          BEQ     25$                     ;BR IF YES
011602  005205                          INC     R5                      ;SET ERROR SEEN FLAG
011604  062704  000001          25$:    ADD     #1,R4                   ;POINT TO NEXT BYTE
011610  020403                          CMP     R4,R3                   ;DONE ALL BYTES?
011612  002001                          BGE     50$                     ;BR IF YES
011614  000764                          BR      15$                     ;DO NEXT BYTE
011616  005705                  50$:    TST     R5                      ;ANY ERRORS SEEN?
011620  001402                          BEQ     55$                     ;BR IF NO
011622  000241                          CLC                             ;SET FAILURE
011624  000401                          BR      60$                     ;
011626  000261                  55$:    SEC                             ;SET SUCCESS
011630  000207                  60$:    RTS     PC                      ;RETURN
```

```
011632     120     122     117  DEBUGMSG: .ASCIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED- ;@@D
011722     045     116     045  FERCM:   .ASCII  /%N%A  ***/
011733     040     040     124  ERCM:    .ASCIZ  /  TSSR ERROR CODE REC'D = /
011766     056     056     056  SIMSG:   .ASCIZ  /.... AFTER DOING SOFT INIT/
012021     124     105     123  TINERR:  .ASCIZ  /TEST: .../
                                        .EVEN
                                ;+
                                ;
                                ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
                                ;
                                ;INPUT:
                                ;
                                ;       R1      CONTENTS OF TSSR AT ERROR
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;       EXECUTES DROP UNIT TO CEASE TESTING
                                ;
                                ;-

012034                                  BGNMSG  SFIMSG
012034                          SFIMSG::
012034  004737  006022                  JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
012040  004737  017202                  JSR     PC,CKDROP       ;DROP UNIT, IF ALLOWED
012044                                  ENDMSG
012044                          L10003:
012044  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
                                ;
                                ;-

012046                                  BGNMSG  PKTSSR
012046                          PKTSSR::
012046  004737  006022                  JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
012052  012700  000004                  MOV     #4,R0           ;NO. OF WORDS IN PACKET
012056  004737  007366                  JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
012062                                  ENDMSG
012062                          L10004:
012062  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
                                        ;-
012064                                          BGNMSG  PKTGETS
012064                                  PKTGETS::
012064  004737  006022                          JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
012070  012700  000002                          MOV     #2,R0           ;NO. OF WORDS IN GET STATUS PACKET
012074  004737  007366                          JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
012100                                          ENDMSG
012100                                  L10005:
012100  104423                                  TRAP    C$MSG
                                        ;+
                                        ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
                                        ;
                                        ;INPUTS:
                                        ;       R1      TSSR CONTENTS
                                        ;       R4      ADDRESS OF COMMAND PACKET
                                        ;-
012102                                          BGNMSG  SFFMSG
012102                                  SFFMSG::
012102  004737  006022                          JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
012106                                          ENDMSG
012106                                  L10006:
012106  104423                                  TRAP    C$MSG
                                                .SBTTL  PKTMES  - PRINT TSSR AND MESSAGE BUFFER
                                        ;+
                                        ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
                                        ;BUFFER FOR ERROR REPORTS
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      CONTENTS OF TSSR
                                        ;       R2      LOW ORDER MESSAGE BUFFER
                                        ;       R3      HIGH ORDER MESSAGE BUFFER ADDRESS
                                        ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
                                        ;
012110                                          BGNMSG  PKTMES
012110                                  PKTMES::
012110  004737  006022                          JSR     PC,PRITSSR              ;PRINT CONTENTS OF TSSR
012114  010200                                  MOV     R2,R0                   ;LOW ORDER ADDRESS
012116  010301                                  MOV     R3,R1                   ;HIGH ORDER ADDRESS
012120  004737  014242                          JSR     PC,PRMESS               ;PRINT THE MESSAGE BUFFER
012124                                          ENDMSG
012124                                  L10007:
012124  104423                                  TRAP    C$MSG
```

```
                                        .SBTTL  ADDSSR  - PRINT TEST ADDRESS AND TSSR
                                ;+
                                ;PRINT ROUTINE TO PRINT THE CONTENTS OF
                                ;TSSR AND A MEMORY TEST ADDRESS
                                ;
                                ;INPUTS:
                                ;
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;       ERRHI   HIGH ORDER MEMORY TEST ADDRESS
                                ;       ERRLO   LOW ORDER MEMORY TEST ADDRESS
                                ;
                                        BGNMSG  ADDSSR
012126                          ADDSSR::
012126                                  JSR     PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
012126  004737  010272                  MOV     TSSR(R5),R1     ;GET CURRENT TSSR
012132  016501  000002                  JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
012136  004737  006022                  ENDMSG
012142
012142                          L10010:
012142  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
                                ;-
012144                                  BGNMSG  MSGEXP
012144                          MSGEXP::
012144  012700  000007                  MOV     #7,R0           ;ASSUME NO EXT FEATURES
012150  005737  002224                  TST     EXTFEA          ;EXT FEATURES SET?
012154  001402                          BEQ     5$              ;BR IF NO
012156  012700  000010                  MOV     #8.,R0          ;EXT FEATURE BUFFER IS 8 WORDS
012162  004737  014552          5$:     JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012166                                  ENDMSG
012166                          L10011:
012166  104423                          TRAP    C$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
012170                          FIFEXP::
012170                                  PRINTX  #FIF1MSG,R1     ;PRINT BYTES TRANSFERRED
012170  010146                          MOV     R1,-(SP)
012172  012746  012242                  MOV     #FIF1MSG,-(SP)
012176  012746  000002                  MOV     #2,-(SP)
012202  010600                          MOV     SP,R0
012204  104415                          TRAP    C$PNTX
012206  062706  000006                  ADD     #6,SP
012212                                  PRINTX  #FIF2MSG        ;PRINT HEADER MSG
012212  012746  012311                  MOV     #FIF2MSG,-(SP)
012216  012746  000001                  MOV     #1,-(SP)
012222  010600                          MOV     SP,R0
012224  104415                          TRAP    C$PNTX
012226  062706  000004                  ADD     #4,SP
012232  010100                          MOV     R1,R0           ;GET BYTE COUNT
012234  004737  015122                  JSR     PC,PRBYTEXP     ;PRINT FIFO BYTES IN ERROR
012240                                  ENDMSG
012240                          L10012:
012240  104423                          TRAP    C$MSG
012242     045     116     045  FIF1MSG:        .ASCIZ  '%N%A NUMBER OF BYTES TRANSFERRED = %D2
012311     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
                                ;-
012350                                  BGNMSG  MSGSTAT
012350                          MSGSTAT::
012350  012701  012412                  MOV     #STATCOD,R1     ;ASCII ADDRESS TABLE
012354  012100                  10$:    MOV     (R1)+,R0        ;DONE ALL MSG LINES?
012356  001410                          BEQ     20$             ;BR IF YES
012360                                  PRINTX  R0              ;PRINT STATUS BIT NAMES
012360  010046                          MOV     R0,-(SP)
012362  012746  000001                  MOV     #1,-(SP)
012366  010600                          MOV     SP,R0
012370  104415                          TRAP    C$PNTX
012372  062706  000004                  ADD     #4,SP
012376  000766                          BR      10$             ;DO ANOTHER MSG LINE
012400  012700  000012          20$:    MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
012404  004737  014552                  JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012410                                  ENDMSG
012410                          L10013:
012410  104423                          TRAP    C$MSG

012412  012430  012472  012563  STATCOD:        .WORD   1$,2$,3$,4$,5$,6$,0
012430     045     116     045  1$:.ASCIZ  '#N#A Tape Bus Signals in Word #8:'
012472     045     116     045  2$:.ASCIZ  '#N#A        PARERR<15>  IEOT  <12>  IFMK  <9>  IRDY<6>  IRWD<2>'
012563     045     116     045  3$:.ASCIZ  '#N#A        IRESV2<14>  IIDENT<11>  IHER  <8>  IONL<5>  IFBY<1>'
012654     045     116     045  4$:.ASCIZ  '#N#A        IRESV1<13>  ICER  <10>  ISPEED<7>  ILDP<4>  IFPT<0>'
012745     045     116     045  5$:.ASCIZ  '#N#A Tape Bus Signals in Word #9:'
013007     045     116     045  6$:.ASCIZ  '#N#A        DATMIS<7>  ILW<6>  OUTRDY<5>  INRDY<4>
                                        .EVEN
```

N5


```
                                        .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
                                ;
                                        BGNMSG  MSGLOOP
013064                          MSGLOOP:
013064  012701  013126                  MOV     #LOOPCOD,R1     ;ASCII ADDRESS TABLE
013070  012100                  10$:    MOV     (R1)+,R0        ;DONE ALL MSG LINES?
013072  001410                          BEQ     20$             ;BR IF YES
013074                                  PRINTX  R0              ;PRINT STATUS BIT NAMES
013074  010046                          MOV     R0,-(SP)
013076  012746  000001                  MOV     #1,-(SP)
013102  010600                          MOV     SP,R0
013104  104415                          TRAP    C$PNTX
013106  062706  000004                  ADD     #4,SP
013112  000766                          BR      10$             ;DO ANOTHER MSG LINE
013114  012700  000012          20$:    MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
013120  004737  014552                  JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
013124                                  ENDMSG
013124                          L10014:
013124  104423                          TRAP    C$MSG

013126  013146  013221  013320  LOOPCOD:        .WORD   1$,2$,3$,4$,5$,6$,7$,0
013146     045     116     045  1$:.ASCIZ  '<NL>A Tape Bus Loopback Signals in Word #8:'
013221     045     116     045  2$:.ASCIZ  '<NL>A       PARERR<15>          IRESV2<14>          IRESV1<13>'
013320     045     116     045  3$:.ASCIZ  '<NL>A IHISP=>IEOT<12>    IWRT=>IIDENT<11>    IREV  =>ICER  <10>'
013417     045     116     045  4$:.ASCIZ  '<NL>A IWFM =>IFMK<09>    IEDIT=>IHER <08>    IFAD  =>ISPEED<07>'
013516     045     116     045  5$:.ASCIZ  '<NL>A ITADO=>IRDY<06>    ITAD1=>IONL <05>    IERASE=>ILDP  <04>'
013615     045     116     045  6$:.ASCIZ  '<NL>A IREW =>IDBY<03>    IRWU =>IRWD <02>    IFEN  =>IFBY  <01>'
013714     045     116     045  7$:.ASCIZ  '<NL>A IGO  =>IFPT<00>'
                                        .EVEN
```

```
                                        .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
013742                          MSGSUB::
013742                                  MOV     #10.,R0         ;SIZE OF WRITE SUBSYSTEM BUFFER
013742  012700  000012                  JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
013746  004737  014552                  ENDMSG
013752                          L10015:
013752                                  TRAP    C$MSG
013752  104423

                                        .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
013754                          MEMADD::
013754  004737  010156                  JSR     PC,PRIADD       ;PRINT MEMORY ADDRESS IN ERROR
013760  013701  002230                  MOV     EXPD,R1         ;GET EXPD DATA
013764  013702  002232                  MOV     RECV,R2         ;GET RECEIVED DATA
013770  004737  007740                  JSR     PC,PRIXOR       ;PRINT EXPD/RECV
013774                                  ENDMSG
013774                          L10016:
013774  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
                                ;-
013776                          PRAMPKT:
013776                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
014002  012701  002240                  MOV     #RAMDATA,R1             ;DATA FROM THE RAM
014006  005002                          CLR     R2                      ;INIT BYTE NUMBER
014010  122124                  5$:     CMPB    (R1)+,(R4)+             ;COMPARE EXPECTED, RECEIVED
014012  001005                          BNE     7$                      ;BR IF NO MATCH
014014                                  FORCERROR       7$,NOTSSR
014024  000436                          BR      10$                     ;BRD
014026  116105  177777          7$:     MOVB    -1(R1),R5               ;GET RECV RAM DATA
014032  116403  177777                  MOVB    -1(R4),R3               ;GET EXPD PACKET DATA
014036                                  XOR     R5,R3                   ;XOR EXPD/RECV
014046  042703  177400                  BIC     #177400,R3              ;LOW BYTE ONLY
014052  116137  177777  002232          MOVB    -1(R1),RECV             ;GET RECEIVED RAM DATA
014060  116437  177777  002230          MOVB    -1(R4),EXPD             ;GET EXPECTED RAM DATA
014066                                  PRINTB  #RAMASC,R2,RECV,EXPD,R3
014066  010346                          MOV     R3,-(SP)
014070  013746  002230                  MOV     EXPD,-(SP)
014074  013746  002232                  MOV     RECV,-(SP)
014100  010246                          MOV     R2,-(SP)
014102  012746  014156                  MOV     #RAMASC,-(SP)
014106  012746  000005                  MOV     #5,-(SP)
014112  010600                          MOV     SP,R0
014114  104414                          TRAP    C$PNTB
014116  062706  000014                  ADD     #14,SP
014122  005202                  10$:    INC     R2                      ;UPDATE BYTE COUNT
014124  005737  002300                  TST     RAMSIZ                  ;DEFAULT TO 8.?
014130  001404                          BEQ     15$                     ;BR IF YES
014132  020237  002300                  CMP     R2,RAMSIZ               ;DONE ALL BYTES?
014136  003724                          BLE     5$                      ;BR IF NO
014140  000403                          BR      25$                     ;
014142  020227  000010          15$:    CMP     R2,#8.                  ;DONE DEFAULT NUMBER OF BYTES?
014146  002720                  20$:    BLT     5$                      ;BR IF NO
014150  005037  002300          25$:    CLR     RAMSIZ                  ;SET DEFAULT RAMSIZ
014154  000207                          RTS     PC                      ;RETURN

014156     045     116     045  RAMASC: .ASCIZ  '%N%A  BYTE: %D2%A RAM: %O3%A Packet: %O3%A XOR:%O3'
                                        .EVEN
```

```
                                        .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
                                ;-
014242                          PRMESS: SAVREG                          ;SAVE THE REGISTERS
014246  010005                          MOV     R0,R5                   ;SAVE LOW ORDER ADDRESS
014250  005737  003132                  TST     KTENABLE                ;ADDRESS ABOVE 28K?
014254  001001                          BNE     10$                     ;BR IF YES
014256  005001                          CLR     R1                      ;SET HIGH ORDER ADDRESS TO 0
014260  010103                  10$:    MOV     R1,R3                   ;SAVE HIGH ORDER ADDRESS
014262  006100                          ROL     R0                      ;SHIFT BIT15 TO C BIT
014264  006101                          ROL     R1                      ;SHIFT TO HIGH ORDER FOR PRINTOUT
014266                                  PRINTX  #PROASC,R1,R5           ;PRINT MESSAGE BUFFER ADDRESS
014266  010546                          MOV     R5,-(SP)
014270  010146                          MOV     R1,-(SP)
014272  012746  014420                  MOV     #PROASC,-(SP)
014276  012746  000003                  MOV     #3,-(SP)
014302  010600                          MOV     SP,R0
014304  104415                          TRAP    C$PNTX
014306  062706  000010                  ADD     #10,SP
014312                                  PRINTX  #PR1ASC                 ;PRINT HEADER FOR CONTENTS
014312  012746  014465                  MOV     #PR1ASC,-(SP)
014316  012746  000001                  MOV     #1,-(SP)
014322  010600                          MOV     SP,R0
014324  104415                          TRAP    C$PNTX
014326  062706  000004                  ADD     #4,SP
014332  005004                          CLR     R4                      ;NUMBER OF THE NEXT WORD
014334  010501                          MOV     R5,R1                   ;COPY LOW ORDER ADDRESS
014336  010300                          MOV     R3,R0                   ;COPY HIGH ORDER ADDRESS
014340  001403                          BEQ     20$                     ;BR IF NOT ABOVE 28K
014342  004737  017316                  JSR     PC,SETMAP               ;SETUP PAR ADDRESS IN R0
014346  010005                          MOV     R0,R5                   ;GET PAR FORMAT ADDRESS ABOVE 28K
014350                          20$:    PRINTX  #PRASC,R4,(R5)+         ;PRINT THE CONTENTS OF MEMORY BUFFER
014350  012546                          MOV     (R5)+,-(SP)
014352  010446                          MOV     R4,-(SP)
014354  012746  014523                  MOV     #PRASC,-(SP)
014360  012746  000003                  MOV     #3,-(SP)
014364  010600                          MOV     SP,R0
014366  104415                          TRAP    C$PNTX
014370  062706  000010                  ADD     #10,SP
014374  005204                          INC     R4                      ;NUMBER OF THE NEXT
014376  020427  000007                  CMP     R4,#7                   ;DONE ALL YET ?
014402  003005                          BGT     50$                     ;BRANCH IF ALL DONE
014404  002761                          BLT     20$                     ;PRINT FIRST 7 WORDS
014406  032763  000200  000012          BIT     #X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
014414  001355                          BNE     20$                     ;PRINT EXTENDED STATUS WORD
014416  000207                  50$:    RTS     PC                      ;RETURN
014420     045     116     045  PROASC: .ASCIZ  '%N%A Message Buffer Address = %O1%O5'
014465     045     116     045  PR1ASC: .ASCIZ  '%N%A Message Buffer Contents:'
014523     045     116     045  PRASC:  .ASCIZ  '%N%A    Word%D1%A: %O'
```

```
                                        .EVEN
                                        .SBTTL  PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
                                ;+
                                ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
                                ;       R0      - NUMBER OF WORDS IN BUFFER
                                ;INPUT INITIAL INPUTS:
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
014552                          PRMSGEXP::
014552                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
014556  010005                          MOV     R0,R5                   ;SAVE NUMBER OF WORDS
014560  013700  002304                  MOV     RCVLOADD,R0             ;GET RECV LOW ADDRESS
014564  010004                          MOV     R0,R4                   ;COPY LOW ADDRESS
014566  013701  002302                  MOV     RCVHIADD,R1             ;GET RECV HIGH ADDRESS
014572  006100                          ROL     R0                      ;SHIFT BIT15 TO C BIT
014574  006101                          ROL     R1                      ;SHIFT TO HIGH ORDER FOR PRINTOUT
014576                                  PRINTX  #PRMSG0,R1,R4           ;PRINT MESSAGE BUFFER ADDRESS
014576  010446                          MOV     R4,-(SP)
014600  010146                          MOV     R1,-(SP)
014602  012746  014732                  MOV     #PRMSG0,-(SP)
014606  012746  000003                  MOV     #3,-(SP)
014612  010600                          MOV     SP,R0
014614  104415                          TRAP    C$PNTX
014616  062706  000010                  ADD     #10,SP
014622                                  PRINTX  #PRMSG1                 ;PRINT HEADER FOR CONTENTS
014622  012746  014777                  MOV     #PRMSG1,-(SP)
014626  012746  000001                  MOV     #1,-(SP)
014632  010600                          MOV     SP,R0
014634  104415                          TRAP    C$PNTX
014636  062706  000004                  ADD     #4,SP
014642  005004                          CLR     R4                      ;NUMBER OF THE CURRENT WORD
014644  012701  002320                  MOV     #EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
014650  012702  002464                  MOV     #RECMSG,R2              ;GET RECV BUFFER ADDRESS
014654  011100                  20$:    MOV     (R1),R0                 ;GET EXPD
014656  011203                          MOV     (R2),R3                 ;GET RECV
014660                                  XOR     R0,R3                   ;XOR EXPD/RECV
014670                                  PRINTX  #PRMSG2,R4,(R1)+,(R2)+,R3
014670  010346                          MOV     R3,-(SP)
014672  012246                          MOV     (R2)+,-(SP)
014674  012146                          MOV     (R1)+,-(SP)
014676  010446                          MOV     R4,-(SP)
014700  012746  015035                  MOV     #PRMSG2,-(SP)
014704  012746  000005                  MOV     #5,-(SP)
014710  010600                          MOV     SP,R0
014712  104415                          TRAP    C$PNTX
014714  062706  000014                  ADD     #14,SP
014720  005204                          INC     R4                      ;NUMBER OF THE NEXT
014722  020405                          CMP     R4,R5                   ;DONE ALL YET?
014724  002001                          BGE     50$                     ;BR IF YES
014726  000752                          BR      20$                     ;DO ANOTHER
014730  000207                  50$:    RTS     PC                      ;RETURN
014732     045     116     045  PRMSG0: .ASCIZ  '%N%A Message Buffer Address = %01%05'
014777     045     116     045  PRMSG1: .ASCIZ  '%N%A Message Buffer Contents:'
015035     045     116     045  PRMSG2: .ASCIZ  '%N%A   WORD #%D2%A  EXPD: %06%A  RECV: %06%A  XOR: %06'
```

```
                                        .EVEN
                                        .SBTTL  PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
                                ;+
                                ;
                                ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
                                ;       ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
                                ;
                                ;       R0      - NUMBER OF BYTES IN BUFFER
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;-
015122                          PRBYTEXP::
015122                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
015126  010005                          MOV     R0,R5                   ;SAVE NUMBER OF BYTES
015130  005037  002316                  CLR     PRMNO                   ;INIT ERROR COUNT
015134  005004                          CLR     R4                      ;NUMBER OF THE CURRENT BYTE
015136  012701  002320                  MOV     #EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
015142  012702  002464                  MOV     #RECMSG,R2              ;GET RECV BUFFER ADDRESS
015146  111100                  20$:    MOVB    (R1),R0                 ;GET EXPD BYTE
015150  042700  177400                  BIC     #^C<377>,R0             ;CLEAR UPPER BYTE
015154  110037  015470                  MOVB    R0,PRBEXP               ;SAVE FOR ERROR REPORT
015160  111203                          MOVB    (R2),R3                 ;GET RECV BYTE
015162  042703  177400                  BIC     #^C<377>,R3             ;CLEAR UPPER BYTE
015166  110337  015472                  MOVB    R3,PRBREC               ;FOR ERROR REPORT
015172                                  XOR     R0,R3                   ;XOR EXPD/RECV
015202  122122                          CMPB    (R1)+,(R2)+             ;EXPD = RECV?
015204  001431                          BEQ     30$                     ;BR IF YES
015206  005237  002316                  INC     PRMNO                   ;UPDATE ERROR COUNT
015212  023727  002316  000010          CMP     PRMNO,#8.               ;PRINTED 8?
015220  101023                          BHI     30$                     ;BR IF YES
015222                          27$:    PRINTX  #PRBMSG,R4,PRBEXP,PRBREC,R3
015222  010346                          MOV     R3,-(SP)
015224  013746  015472                  MOV     PRBREC,-(SP)
015230  013746  015470                  MOV     PRBEXP,-(SP)
015234  010446                          MOV     R4,-(SP)
015236  012746  015336                  MOV     #PRBMSG,-(SP)
015242  012746  000005                  MOV     #5,-(SP)
015246  010600                          MOV     SP,R0
015250  104415                          TRAP    C$PNTX
015252  062706  000014                  ADD     #14,SP
015256                                  FORCEXIT        50$             ;BBD
015266  000404                          BR      35$                     ;BBD
015270                          30$:
015270                                  FORCERROR  27$,NOTSSR           ;BBD
015300                          35$:                                    ;BBD
015300  005204                          INC     R4                      ;NUMBER OF THE NEXT
015302  020405                          CMP     R4,R5                   ;DONE ALL YET?
015304  002001                          BGE     50$                     ;BR IF YES
015306  000717                          BR      20$                     ;DO ANOTHER
015310                          50$:    PRINTX  #PRBTOT,PRMNO           ;PRINT TOTAL ERROR COUNT
015310  013746  002316                  MOV     PRMNO,-(SP)
015314  012746  015423                  MOV     #PRBTOT,-(SP)
015320  012746  000002                  MOV     #2,-(SP)
015324  010600                          MOV     SP,R0
```

```
     015326  104415                          TRAP    C$PNTX
     015330  062706  000006                  ADD     #6,SP
1783 015334  000207                          RTS     PC              ;RETURN
1784
1785 015336     045     116     045  PRBMSG: .ASCIZ  '%N%A   BYTE #%D2%A  EXPD: %O3%A  RECV: %O3%A  XOR: %O3'
1786 015423     045     116     045  PRBTOT: .ASCIZ  '%N%A NUMBER OF BYTES IN ERROR = %D2'
1787                                         .EVEN
1788 015470  000000                  PRBEXP: .WORD   0               ;EXPD
1789 015472  000000                  PRBREC: .WORD   0               ;RECV
1790                                         .SBTTL  EXPREC  - PRINT EXPD/RECV WORD DATA
1791                                 ;+
1792                                 ;
1793                                 ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1794                                 ;
1795                                 ;INPUTS:
1796                                 ;
1797                                 ;       R1      RECEIVED DATA
1798                                 ;       R2      EXPECTED DATA
1799                                 ;
1800                                 ;
1801
1802 015474                                  BGNMSG  EXPREC
     015474                          EXPREC::
1803 015474  004737  007740                  JSR     PC,PRIXOR               ;PRINT THE DATA
1804 015500                                  ENDMSG
     015500                          L10017:
     015500  104423                          TRAP    C$MSG
1805                                         .SBTTL  EXPBREC - PRINT EXPD/RECV BYTE DATA
1806                                 ;+
1807                                 ;
1808                                 ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
1809                                 ;
1810                                 ;
1811                                 ;INPUTS:
1812                                 ;
1813                                 ;       R1      RECEIVED DATA BYTE
1814                                 ;       R2      EXPECTED DATA BYTE
1815                                 ;
1816                                 ;-
1817
1818 015502                                  BGNMSG  EXPBREC
     015502                          EXPBREC::
1819 015502  004737  007610                  JSR     PC,PRIBXOR              ;PRINT THE DATA
1820 015506                                  ENDMSG
     015506                          L10020:
     015506  104423                          TRAP    C$MSG
1821
1822
1823                                         .SBTTL  RAMERR  - PRINT RAM AND PACKET DATA
1824                                 ;+
1825                                 ;
1826                                 ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
1827                                 ;
1828                                 ;INPUTS:
1829                                 ;
1830                                 ;       R4      POINTER TO COMMAND PACKET
1831                                 ;
```

```
                                ;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
     015510                     RAMERR::
     015510                             JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
     015510  004737  013776             ENDMSG
     015514                     L10021:
     015514                             TRAP    C$MSG
     015514  104423

                                        .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
     015516                     RAMTADD::
     015516                             JSR     PC,PRITADD      ;PRINT TEST ADDRESS
     015516  004737  010272             JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
     015522  004737  013776             ENDMSG
     015526                     L10022:
     015526                             TRAP    C$MSG
     015526  104423

                                        .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
015530                                  RAMEXP::
015530  042701  177400                          BIC     #↑C<377>,R1             ;SAVE EXPD RAM DATA BYTE
015534  042702  177400                          BIC     #↑C<377>,R2             ;SAVE EXPD RAM DATA BYTE
015540  004737  010064                          JSR     PC,PRIRAM               ;PRINT THE RAM ADDRESS
015544  004737  007740                          JSR     PC,PRIXOR               ;PRINT THE DATA
015550                                          ENDMSG
015550                                  L10023:
015550  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
                                        ;-

015552                                          BGNMSG  TIMEXP
015552                                  TIMEXP::
015552                                          PRINTX  #TIMSGO                 ;PRINT HEADER
015552  012746  015600                          MOV     #TIMSGO,-(SP)
015556  012746  000001                          MOV     #1,-(SP)
015562  010600                                  MOV     SP,R0
015564  104415                                  TRAP    C$PNTX
015566  062706  000004                          ADD     #4,SP
015572  004737  007740                          JSR     PC,PRIXOR               ;PRINT THE DATA
015576                                          ENDMSG
015576                                  L10024:
015576  104423                                  TRAP    C$MSG

015600     045     116     045  TIMSGO: .ASCIZ  '%N%A TIMER A STATUS IS IN BIT 3%N%A TIMER B STATUS IS IN BIT 2'
                                        .EVEN
                                                .SBTTL  BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

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

015700                                          BGNMSG  BADSSR
015700                                  BADSSR::
015700  010246                                  MOV     R2,-(SP)                ;SAVE DATA TRANSFERRED
015702  042702  177400                          BIC     #177400,R2              ;GET JUST ONE BYTE
```

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

```
                                        .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
                                        ;
                                        ;-

015774                                  SOFINIT::
015774                                          SAVREG                  ; SAVE THE REGISTERS
016000  012765  000000  000002                  MOV     #0,TSSR(R5)     ; DO THE INIT.
016006  004737  016250                          JSR     PC,WAITF        ; WAIT FOR SSR
016012  016500  000002                          MOV     TSSR(R5),R0     ;GET THE TSSR REGISTER
016016  010004                                  MOV     R0,R4           ;TSSR CONTENTS
016020  042704  176277                          BIC     #^C<HIADDR!OFL>,R4
016024  052704  002200                          BIS     #SSR!NBA,R4     ;R4 HAS EXPECTED CONTENTS
016030  020400                                  CMP     R4,R0           ;ONLY EXPECTED BITS SET ?
016032  001402                                  BEQ     5$              ;BRANCH IF OKAY
016034  000241                                  CLC                     ;CLEAR THE CARRY FOR ERROR
016036  000401                                  BR      10$             ;GO TO EXIT
016040  000261                          5$:     SEC                     ;SET THE CARRY BIT
016042  000207                          10$:    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
                                        ;
                                        ;-

016044                                  CHKAMB:
016044                                          SAVREG                          ;SAVE THE GENERAL REGISTERS
016050  010004                                  MOV     R0,R4                   ;CONTENTS OF TSSR
016052  032700  100000                          BIT     #SC,R0                  ;IS BIT 15 SET ?
016056  001004                                  BNE     5$                      ;BRANCH IF YES
016060  032700  174077                          BIT     #^C<NBA!OFL!SSR!HIADDR>,R0      ;ANY OTHER BITS SET ?
016064  001023                                  BNE     40$                     ;MUST BE AN ERROR
016066  000424                                  BR      45$                     ;RETURN WITH SUCCESS
016070  032700  000200                  5$:     BIT     #SSR,R0                 ;IS READY BIT SET ?
016074  001011                                  BNE     10$                     ;BRANCH IF READY BIT IS SET.
016076  032700  000040                          BIT     #BIT5,R0                ;IS FATAL ERROR BIT SET ?
016102  001414                                  BEQ     40$                      ERROR IF NOT
016104  042704  177761                          BIC     #^CTERCLS,R4            ;CLEAR ALL BUT TERMINATION CODE
016110  020427  000016                          CMP     R4,#16                  ;ALL THREE BITS MUST BE SET
016114  001007                                  BNE     40$                     ;ERROR IF NOT SET
016116  000410                                  BR      45$                     ;OK IF ALL ARE SET
016120  032700  000040                  10$:    BIT     #BIT5,R0                ;IS FATAL ERROR BIT SET ?
016124  001405                                  BEQ     45$                     ;ERROR IF BIT IS SET WITH SSR
016126  032700  000006                          BIT     #BIT2!BIT1,R0           ;IS THIS A FUNCTION REJECT
016132  001002                                  BNE     45$                     ;BR. IF TSSR IS OK
016134  000241                          40$:    CLC                             ;AMBIGUOUS CONTENTS
016136  000401                                  BR      50$
016140  000261                          45$:    SEC                             ;SHOW SUCCESS - NO AMBIGUITY
016142  000207                          50$:    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
016144     000                          INTMASK:        .BYTE   0
                                        ;INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
016145     000                          INTFLAG:        .BYTE   0

                                        ;SAVED INTERRUPT VECTOR:
016146  000000                          INTVEC: .WORD   0
                                        ;SAVE CPU PC
016150  000000                          INTCPC: .WORD   0

                                        ;SUBROUTINE TO ENABLE INTERRUPTS:
016152  010046                          ENAINT: MOV     R0,-(SP)        ;SAVE R0
016154  013700  002206                          MOV     IVEC,R0         ;GET POINTER TO VECTORS
016160  012720  016216                          MOV     #INTR,(R0)+     ;SET UP INTERRUPT VECTOR
016164  012720  000340                          MOV     #PRI07,(R0)+
016170  012600                                  MOV     (SP)+,R0        ;RESTORE R0
016172  011646                                  MOV     (SP),-(SP)
016174  012766  000000  000002                  MOV     #0,2(SP)        ;SET CPU TO LEVEL 0
016202  000002                                  RTI

                                        ;SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
016204  011646                          DSBINT: MOV     (SP),-(SP)
016206  012766  000340  000002                  MOV     #PRI07,2(SP)
016214  000002                                  RTI
                                                .SBTTL  INTR    - INTERRUPT HANDLERS

016216                                          BGNSRV  INTR            ;DEFINE INTERRUPT ENTRY
016216                                  INTR::
016216  012737  000001  002222                  MOV     #1,INTRECV      ;SET FLAG TO SHOW INTERRUPT RECEIVED
016224  105037  016145                          CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
016230  132737  000001  016144                  BITB    #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
016236  001003                                  BNE     1$              ;BR IF YES
016240  152737  000001  016145                  BISB    #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.

                                        ;SAVE REGISTERS, MSG BUFFER, ETC.
016246                                  1$:
016246                                          ENDSRV
016246                                  L10026:
016246  000002                                  RTI
```

NF,


```
                                        .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
                                ;
016250  000401                  WAITF:: BR      1$              ;NOP WHEN SUPER FIXED
016252                                  BREAK                   ; DO A SUPVSR BREAK FIRST.
016252  104422                          TRAP    C$BRK
016254  012746  011000          1$      MOV     #11000,-(SP)    ;25-APRIL-83 REV B    1100 MSEC TIMER
016260  016500  000002          2$:     MOV     TSSR(R5),R0     ;READ THE TSSR REGISTER
016264  105700                          TSTB    R0              ;TEST FOR READY BIT SET

016266  100420                          BMI     3$              ; EXIT ON STOP FLAG.
016270                                  DELAY   1               ; WAIT 100 USEC
016270  012727  000001                  MOV     #1,(PC)+
016274  000000                          .WORD   0
016276  013727  002116                  MOV     L$DLY,(PC)+
016302  000000                          .WORD   0
016304  005367  177772                  DEC     -6(PC)
016310  001375                          BNE     . 4
016312  005367  177756                  DEC     -22(PC)
016316  001367                          BNE     . 20
016320  005316                          DEC     (SP)            ;REDUCE DELAY COUNT
016322  001356                          BNE     2$              ;RETRY UNTIL TIMER EXPIRES
016324  000241                          CLC                     ; C = 0, CONTROLLER STILL RUNNING...
016326  000401                          BR      4$              ;...OR HUNG-UP AFTER 300 MSEC.
016330  000261                  3$:     SEC                     ; C = 1, CONTROLLER IS STOPPED.
016332  005326                  4$:     DEC     (SP)+           ;RESTORE STACK WITHOUT CHANGING CARRY BIT
016334  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
                                        ;
016336                                  CHKTSSR:
016336  004737  016250                          JSR     PC,WAITF                ;WAIT FOR READY
016342  103014                                  BCC     20$                     ;BRANCH IF TIME OUT
016344  004737  016044                          JSR     PC,CHKAMB               ;TSSR AMBIGOUS?
016350  103006                                  BCC     10$                     ;BR IF YES
016352  032700  100000                          BIT     #SC,R0                  ;SPECIAL CONDITION SET?
016356  001405                                  BEQ     15$                     ;BR IF NO
016360  032700  074000                          BIT     #<SCE!BIE!RMR!NXM>,R0   ;ANY ERROR BITS SET?
016364  001402                                  BEQ     15$                     ;BR IF NO
016366  000241                          10$:    CLC                             ;SET FAILURE
016370  000401                                  BR      20$                     ;
016372  000261                          15$:    SEC                             ;SET SUCCESS
016374  000207                          20$:    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.
016376  012737  016430  000004          XNXM:   MOV     #2$,B#4                 ; SET BUSERR VECTOR.
016404  012737  000200  000006                  MOV     #PRI04,B#6
016412  005003                                  CLR     R3                      ;FLAG.
016414  005711                          1$:     TST     (R1)                    ;TEST THE ADDRESS(ES).
                                                                                ;IF ANY TRAP, CONTINUE AT 2$.
016416  020102                                  CMP     R1,R2                   ;OTHERWISE, CONTINUE HERE.
016420  001407                                  BEQ     3$                      ;BR IF FINISHED (NO NEXM'S).
016422  062701  000002                          ADD     #2,R1                   ;SET NEXT ADDRESS...
016426  000772                                  BR      1$                      ;...AND CONTINUE.
016430  005103                          2$:     COM     R3                      ;GOT ONE, SET FLAG...
016432  012716  016440                          MOV     #3$,(SP)
016436  000002                                  RTI                             ;...AND DISMISS INTERRUPT...
016440                                  3$:     CLRVEC  #4                      ;...AND GIVE BACK THE VECTOR.
        016440  012700  000004                  MOV     #4,R0
        016444  104436                          TRAP    C$CVEC
016446  005703                                  TST     R3                      ;DID WE CATCH ONE ??
016450  001401                                  BEQ     .+4                     ;NO, "C" = 0, SKIP NEXT.
016452  000261                                  SEC                             ;YES, "C" = 1, (R1) = NEXM ADDR.
016454  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
                                ;
016456                          TSTLOOP::
016456  005737  002166                  TST     NOITS           ; ITERATIONS INHIBITED?
016462  001006                          BNE     1$              ; YES.
016464  005737  002202                  TST     QVP             ; NO.
016470  100403                          BMI     1$              ;LOOPS DISALLOWED IN QUICK PASS.
016472  005337  002214                  DEC     LOOPCNT         ; BUMP LOOP COUNTER.
016476  001002                          BNE     2$
016500  000241                  1$:     CLC                     ;LOOP DISALLOWED, OR DONE.
016502  000401                          BR      3$
016504  000261                  2$:     SEC                     ;LOOP ENABLED.
016506  000207                  3$:     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
                                ;
                                ;-

016510                          TSTSETUP::
016510  010046                          MOV     R0,-(SP)        ;SAVE THE TEST ID MESSAGE
016512  005037  003152                  CLR     SIFLAG          ; CLEAR "SOFT INIT" FLAG
016516  005037  016756                  CLR     ERRK            ; CLEAR LOCAL ERROR COUNTER.
016522  005037  005770                  CLR     EXTA            ; CLEAR ERROR EXTENSION FLAG.
016526  105037  016144                  CLRB    INTMASK         ; CLEAR INTERRUPT MASK (CHECK ERROR)
016532  013700  002200                  MOV     UNITN,R0        ; GET THE UNIT NUMBER.
016536  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
016540  005737  003112                  TST     NODEV           ; DID STARTUP FIND THE DEVICE?
016544  001430                          BEQ     4$              ; BR IF YES
016546  100010                          BPL     3$              ; BR IF NOT IDLE
```

```
016550  052760  160000  003174          BIS     #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
016556                                  ERRDF   1,NXR,NXRERR     ; NO DEVICE HERE    PRINT IT
016556  104455                          TRAP    C$ERDF
016560  000001                          .WORD   1
016562  003736                          .WORD   NXR
016564  005734                          .WORD   NXRERR
016566  000407                          BR      2$
016570  052760  160001  003174  3$:     BIS     #160001,ERTABL(R0)       ; FLAG ERROR IN THE ERROR TABLE
016576                                  ERRDF   2,NOINIT         ; DEVICE NOT IDLE
016576  104455                          TRAP    C$ERDF
016600  000002                          .WORD   2
016602  004333                          .WORD   NOINIT
016604  000000                          .WORD   0
016606  012737  177777  003110  2$:     MOV     #-1,DUFLG        ; DROP THE UNIT
016614                                  DODU    UNITN
016614  013700  002200                  MOV     UNITN,R0
016620  104451                          TRAP    C$DODU
016622                                  DOCLN                    ; ABORT THE PASS
016622  104444                          TRAP    C$DCLN
016624  000423                          BR      5$

016626                          4$:     RFLAGS  R0               ; GET THE OPERATOR FLAGS.
016626  104421                          TRAP    C$RFLA
016630  032700  001000                  BIT     #PNT,R0          ; PRINT THE TEST NUMBERS?
016634  001412                          BEQ     1$               ; BR IF NO
016636  011600                          MOV     (SP),R0          ;GET THE ID MESSAGE
016640                                  PRINTF  #TNAM,R0                 ;DISPLAY THE TEST ID
016640  010046                          MOV     R0,-(SP)
016642  012746  016704                  MOV     #TNAM,-(SP)
016646  012746  000002                  MOV     #2,-(SP)
016652  010600                          MOV     SP,R0
016654  104417                          TRAP    C$PNTF
016656  062706  000006                  ADD     #6,SP
016662  005237  002212          1$:     INC     TSTCNT           ; BUMP TEST COUNTER.
016666                                  SETPRI  IPRI             ;PRIORITY THAT OF DEVICE
016666  013700  002210                  MOV     IPRI,R0
016672  104441                          TRAP    C$SPRI
016674  005726                  5$:     TST     (SP)+            ;FIX UP THE STACK
016676  013705  002204                  MOV     CSRADDR,R5       ; ADDRESS OF TSV REGISTERS ON UNIBUS
016702  000207                          RTS     PC
016704     045     123     045  TNAM:   .ASCIZ  '%S%T%A Test'
                                        .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).
                                ;
016720                          TSTEND: RFLAGS  R0
016720  104421                          TRAP    C$RFLA
016722  030027  020000                  BIT     R0,#IER
016726  001412                          BEQ     1$               ; BR IF "IER" NOT SET.
016730                                  PRINTF  #ESUM,ERRK       ; PRINT ERROR COUNT.
016730  013746  016756                  MOV     ERRK,-(SP)
016734  012746  016760                  MOV     #ESUM,-(SP)
016740  012746  000002                  MOV     #2,-(SP)
016744  010600                          MOV     SP,R0
016746  104417                          TRAP    C$PNTF
```

```
        016750  062706  000006                  ADD     #6,SP
   2249 016754  000207                  1$:     RTS     PC
   2250
   2251 016756  000000                  ERRK:   0                       ; LOCAL ERROR COUNT.
   2252 016760     045     101     040  ESUM:   .ASCIZ  /%A %D%A ERRORS/
   2253 016777     105     122     122  EMAXDU: .ASCIZ  /ERROR LIMIT REACHED  - DROPPING UNIT/
   2254                                         .EVEN
```

```
                                        .SBTTL  INCERK  - INCREMENT LOCAL ERROR COUNT
                                ;+
                                ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
                                ;-
017044  005237  016756          INCERK: INC     ERRK            ; INCREMENT LOCAL ERROR COUNT
017050  010046                          MOV     R0,-(SP)        ; SAVE R0
017052  013700  002200                  MOV     UNITN,R0        ; GET UNIT NUMBER,
017056  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
017060  062700  003174                  ADD     #ERTABL,R0      ; R0 GETS ADDRESS OF ERROR TABLE ENTRY.
017064  005210                          INC     (R0)            ; INCREMENT THE DEVICE ERROR COUNT
017066  032710  007777                  BIT     #7777,(R0)      ; DID WE OVERFLOW THE FIELD?
017072  001001                          BNE     1$              ; BR IF NO.
017074  005310                          DEC     (R0)            ; YES -- BACK IT UP TO 7777.
017076  012600                  1$:     MOV     (SP)+,R0        ; RESTORE R0
017100  000207                          RTS     PC              ; RETURN TO CALLER

017102  010046                  CKEMAX: MOV     R0,-(SP)        ; SAVE R0
017104  013700  002200                  MOV     UNITN,R0        ; GET UNIT NUMBER
017110  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET
017112  016000  003174                  MOV     ERTABL(R0),R0   ; GET ERROR TABLE ENTRY
017116  042700  170000                  BIC     #170000,R0      ; EXTRACT ERROR COUNT FIELD
017122  020037  002172                  CMP     R0,GERRMAX      ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
017126  103004                          BHIS    1$              ; BR IF YES
017130  023737  016756  002170          CMP     ERRK,LERRMAX    ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
017136  103417                          BLO     2$              ; BR IF NO
017140                          1$:     RFLAGS  R0              ; GET OPERATOR FLAGS
017140  104421                          TRAP    C$RFLA
017142  032700  000040                  BIT     #IDU,R0         ; IS DROPPING INHIBITED?
017146  001013                          BNE     2$              ; BR IF YES.
017150  012737  177777  003110          MOV     #-1,DUFLG       ; NO -- DROP THE UNIT
017156                                  ERRDF   4,EMAXDU
017156  104455                          TRAP    C$ERDF
017160  000004                          .WORD   4
017162  016777                          .WORD   EMAXDU
017164  000000                          .WORD   0
017166                                  DODU    UNITN
017166  013700  002200                  MOV     UNITN,R0
017172  104451                          TRAP    C$DODU
017174                                  DOCLN
017174  104444                          TRAP    C$DCLN
017176  012600                  2$:     MOV     (SP)+,R0        ; RESTORE R0
017200  000207                          RTS     PC              ; RETURN TO CALLER
```

```
                                        .SBTTL  CKDROP  - CHECK IF UNIT SHOULD BE DROPPED
                                ;+
                                ; CHECK IF UNIT SHOULD BE DROPPED
                                ;-
017202  010046                  CKDROP: MOV     R0,-(SP)
017204                                  FORCERROR       1$,NOTSSR
017214                                  RFLAGS  R0
017214  104421                          TRAP    C$RFLA
017216  032700  000040                  BIT     #IDU,R0
017222  001010                          BNE     1$
017224  011600                          MOV     (SP),R0
017226  012737  177777  003110          MOV     #-1,DUFLG
017234                                  DODU    UNITN
017234  013700  002200                  MOV     UNITN,R0
017240  104451                          TRAP    C$DODU
017242                                  DOCLN                   ;ABORT THE PASS
017242  104444                          TRAP    C$DCLN
017244  012600                  1$:     MOV     (SP)+,R0
017246  000207                          RTS     PC


                                        .SBTTL  CONFIG  - DETERMINE CONFIGURATION OF SYSTEM
                                ;
                                ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
                                ;
017250                          CONFIG:
017250  004737  015774                  JSR     PC,SOFINIT
017254  000207                          RTS     PC
                                        .SBTTL  KTON,KTOFF      - ENABLE/DISABLE MEMORY MANAGEMENT
                                ;
                                ; SUBROUTINE - ENABLE MEM MGT.
                                ;
017256  005737  003130          KTON:   TST     KTFLG           ; GOT KT?
017262  001403                          BEQ     1$              ; NO.
017264  012737  000001  177572          MOV     #1,SR0          ; YES. ENABLE KT11.
017272  000207                  1$:     RTS     PC

                                ;
                                ; SUBROUTINE - DISABLE MEM MGT.
                                ;
017274  005737  003130          KTOFF:  TST     KTFLG           ; GOT KT11?
017300  001405                          BEQ     1$              ; NO.
017302  000240                          NOP
017304  000240                          NOP
017306  012737  000000  177572          MOV     #0,SR0          ; DISABLE KT.
017314  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
                                ;-
017316                          SETMAP:
017316                                  SAVREG                          ;SAVE R1-R4 UNTIL NEXT RETURN
017322  005737  003130                  TST     KTFLG                   ;SYSTEM HAVE ABOVE 28K?
017326  001433                          BEQ     10$                     ;BR IF NO
017330  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
017362  042701  000177                  BIC     #177,R1                 ;ALINE FOR LOWER 4K BOUNDARY
017366  020137  003130                  CMP     R1,KTFLG                ;HIGHER THAN EXISTING MEMORY?
017372  103011                          BHIS    10$                     ;BR IF YES
017374  010137  172354                  MOV     R1,@#KIPAR6             ;SETUP MAPPING REGISTER PAR6
017400  042702  160000                  BIC     #160000,R2              ;SETUP DISPLACEMENT IN PAGE
017404  062702  140000                  ADD     #140000,R2              ;ADD IN PAR6 BIAS
017410  010200                          MOV     R2,R0                   ;RETURN IN R0
017412  000261                          SEC                             ;SET SUCCESS
017414  000401                          BR      15$                     ;
017416  000241                  10$:    CLC                             ;SET FAILURE
017420  000207                  15$:    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
                                ;-
                                ;
017422                          FILLMEM:
017422                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
017426  004737  017274                  JSR     PC,KTOFF        ;DISABLE KT.
```

```
017432  010003                          MOV     R0,R3           ;COPY TEST PATTERN
017434  013701  003122                  MOV     FREE,R1         ;GET FIRST FREE LOCATION
017440  013702  003124                  MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
017444  010321                  10$:    MOV     R3,(R1)+        ;STORE A BACKGROUND WORD
017446  005302                          DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
017450  003375                          BGT     10$             ;BR IF NO
017452  005737  003130                  TST     KTFLG           ; GOT KT?
017456  001477                          BEQ     55$             ; NO. GET OUT.
017460  004737  017256                  JSR     PC,KTON         ; YES. ENABLE KT.
017464  005000                          CLR     R0              ;HIGH ORDER ADDRESS START
017466  013701  003150                  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
017536  004737  017316                  JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
017542  010320                  30$:    MOV     R3,(R0)+        ;STORE TEST PATTERN IN >28K ADDRESS
017544  020027  160000                  CMP     R0,#160000      ;END OF PAR6 MAPPING AREA?
017550  103774                          BLO     30$             ;BR IF NO
017552  162700  020000                  SUB     #20000,R0       ;BACKUP INTO PAR6 MAPPING BEGIN
017556  062737  000200  172354          ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
017564  023737  172354  003130          CMP     @#KIPAR6,KTFLG  ;END OF MEMORY?
017572  001427                          BEQ     50$             ;BR IF YES
017574  005737  003142                  TST     T23A            ;11/23A?
017600  001407                          BEQ     35$             ;NO KEEP GOING
017602  013704  177572                  MOV     SR0,R4          ;GET SR0 CONTENTS
017606  042704  177761                  BIC     #177761,R4      ;CLEAR ALL BUT PAGE NUMBER
017612  022704  000016                  CMP     #16,R4          ;SEE IF PAGE 7
017616  001415                          BEQ     50$             ;EXIT IF THERE
017620  005737  003144          35$:    TST     T23B            ;11/23B?
017624  001410                          BEQ     45$             ;NO KEEP GOING
017626  023727  172354  007600          CMP     @#KIPAR6,#7600  ;REACHED 18 BITS?
017634  103001                          BHIS    40$             ;YES
017636  000403                          BR      45$             ;NO KEEP GOING
017640  012737  000020  172516  40$:    MOV     #20,SR3         ;SET 22 BIT RELOCATION
017646  000137  017542          45$:    JMP     30$             ;KEEP GOING ON ETC.
017652  004737  017274          50$:    JSR     PC,KTOFF        ; DISABLE KT.
017656  000207                  55$:    RTS     PC
```

```
                                        .SBTTL  CMPMEM    COMPARE MEMORY TO BACKGROUND PATTERN
                                ;+
                                ; COMPARE MEMORY WITH A BACKGROUND PATTERN
                                ;
                                ; INPUTS:
                                ;
                                ;       R0 = BACKGROUND PATTERN
                                ;       FREE    = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
                                ;       KTFLG   = SET TO HIGHEST MEMORY LOCATION IF > 28K.
                                ;
                                ; OUTPUTS:
                                ;
                                ;       CARRY   - SET IF NO ERROR
                                ;       CARRY   - CLR IF ERROR
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       ERRHI   - ERROR HIGH ADDRESS
                                ;       ERRLO   - ERROR LOW ADDRESS
                                ;       EXPD    - EXPECTED DATA
                                ;       RECV    - RECEIVED DATA
                                ;-
017660                          CMPMEM:
017660                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
017664  010003                          MOV     R0,R3           ;COPY TEST PATTERN
017666  004737  017274                  JSR     PC,KTOFF        ;DISABLE KT.
017672  013701  003122                  MOV     FREE,R1         ;GET FIRST FREE LOCATION
017676  013702  003124                  MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
017702  020311                  10$:    CMP     R3,(R1)         ;FREE SPACE LOCATION EQUAL TO EXPD?
017704  001411                          BEQ     15$             ;BR IF YES
017706  010137  002236                  MOV     R1,ERRLO        ;SAVE ADDRESS IN ERROR
017712  005037  002234                  CLR     ERRHI           ;NO HIGH ADDRESS
017716  010337  002230                  MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
017722  011137  002232                  MOV     (R1),RECV       ;SAVE RECV FOR ERPOR REPORT
017726  000474                          BR      50$             ;
017730  005721                  15$:    TST     (R1)+           ;POINT TO NEXT ADDRESS
017732  005302                          DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
017734  003362                          BGT     10$             ;BR IF NO
017736  005737  003130                  TST     KTFLG           ; GOT KT?
017742  001472                          BEQ     55$             ; NO. GET OUT.
017744  004737  017256                  JSR     PC,KTON         ; YES. ENABLE KT.
017750  005000                          CLR     R0              ;HIGH ORDER ADDRESS START
017752  013701  003150                  MOV     PST32W,R1       ;GET >28K START ADDRESS (IN 32W BLOCKS)
        000006                          .REPT   6
                                        ROL     R1              ;CONVERT BLOCKS TO WORDS
                                        ROL     R0              ;MAKE IT DOUBLE PRECISION
                                        .ENDR
020006  042701  000177                  BIC     #177,R1         ;ALINE 4K BOUNDARY
020012  010046                          MOV     R0,-(SP)        ;SAVE HIGH ORDER
020014  010146                          MOV     R1,-(SP)        ;SAVE LOW ORDER
020016  004737  017316                  JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
020022  010004                          MOV     R0,R4           ;COPY ADDRESS BIASED TO PAR6
020024  012601                          MOV     (SP)+,R1        ;RESTORE LOW ORDER IN NON PAR6 FORMAT
020026  012600                          MOV     (SP)+,R0        ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
020030  020314                  30$:    CMP     R3,(R4)         ;ABOVE 28K LOCATION EQUAL EXPD?
020032  001411                          BEQ     32$             ;BR IF YES
020034  010037  002234                  MOV     R0,ERRHI        ;SAVE HIGH ORDER IN ERROR
```

```
020040  010137  002236                  MOV     R1,ERRLO          ;SAVE LOW ORDER IN ERROR
020044  010337  002230                  MOV     R3,EXPD           ;SAVE EXPD FOR ERROR REPORT
020050  011437  002232                  MOV     (R4),RECV         ;SAVE RECV FOR ERROR REPORT
020054  000421                          BR      50$               ;
020056  062701  000002          32$:    ADD     #2,R1             ;UPDATE NON PAR6 ADDRESS
020062  005500                          ADC     R0                ;MAKE IT DOUBLE PRECISION ADD
020064  062704  000002                  ADD     #2,R4             ;UPDATE PAR FORMAT ADDRESS
020070  020427  160000                  CMP     R4,#160000        ;END OF PAR6 MAPPING AREA?
020074  103755                          BLO     30$               ;BR IF NO
020076  162704  020000                  SUB     #20000,R4         ;BACKUP INTO PAR6 MAPPING BEGIN
020102  062737  000200  172354          ADD     #200,@#KIPAR6     ;POINT TO NEXT 4K BLOCK >28K.
020110  023737  172354  003130          CMP     @#KIPAR6,KTFLG    ;END OF MEMORY?
020116  101744                          BLOS    30$               ;BR IF NO
020120  004737  017274          50$:    JSR     PC,KTOFF          ;TURN OFF MEMORY MAPPING
020124  000241                          CLC                       ;SET FAILURE
020126  000403                          BR      60$               ;
020130  004737  017274          55$:    JSR     PC,KTOFF          ;TURN OFF MEMORY MAPPING
020134  000261                          SEC                       ;SET SUCCESS
020136  000207                  60$:    RTS     PC
                                        .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
                                ;
                                ;-

020140                          REGSAV:
020140  010446                          MOV     R4,-(SP)
020142  010346                          MOV     R3,-(SP)
020144  010246                          MOV     R2,-(SP)
020146  010146                          MOV     R1,-(SP)
020150  010546                          MOV     R5,-(SP)
020152  016605  000012                  MOV     10.(SP),R5
020156  004736                          JSR     PC,@(SP)+
020160  012601                          MOV     (SP)+,R1
020162  012602                          MOV     (SP)+,R2
020164  012603                          MOV     (SP)+,R3
020166  012604                          MOV     (SP)+,R4
020170  012605                          MOV     (SP)+,R5
020172  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
                                ;-
020174                          GETPAT::
020174                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
020200                          1$:     GMANID  DATASC,PATDAT,0,377,0,377,NO
020200  104443                          TRAP    C$GMAN
020202  000406                          BR      10000$
020204  020230                          .WORD   PATDAT
020206  000022                          .WORD   T$CODE
020210  020232                          .WORD   DATASC
020212  000377                          .WORD   377
020214  000000                          .WORD   T$LOLIM
020216  000377                          .WORD   T$HILIM
020220                          10000$:
020220                                  BNCOMPLETE      1$              ;RETRY IF ERROR
020220  103367                          BCC     1$
020222  013700  020230                  MOV     PATDAT,R0               ;DATA PATTERN FROM OPERATOR
020226  000207                          RTS     PC                      ;RETURN TO CALLER

                                ;+
                                ;LOCAL DATA AREA
                                ;-

020230  000000                  PATDAT: .WORD   0                       ;TEMPORARY STORAGE FOR DATA
020232     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
                                ;-
020256                          GETSEL::
020256                                  SAVREG                          ;SAVE GENERAL REGISTERS
020262  010002                          MOV     R0,R2                   ;SAVE THE MENU ADDRESS
020264  010203                  1$:     MOV     R2,R3                   ;START OF MENU STRING
020266  005713                  2$:     TST     (R3)                    ;END OF ASCII ?
020270  001412                          BEQ     3$                      ;BRANCH IF ALL LINES DISPLAYED
020272                                  PRINTF  #SELASC,(R3)+           ;DISPLAY THE MENU
020272  012346                          MOV     (R3)+,-(SP)
020274  012746  020442                  MOV     #SELASC,-(SP)
020300  012746  000002                  MOV     #2,-(SP)
020304  010600                          MOV     SP,R0
020306  104417                          TRAP    C$PNTF
020310  062706  000006                  ADD     #6,SP
020314  000764                          BR      2$
020316                          3$:     GMANID  MENASC,MENRES,D,-1,0,-1,NO
020316  104443                          TRAP    C$GMAN
020320  000406                          BR      10001$
020322  020476                          .WORD   MENRES
020324  000042                          .WORD   T$CODE
020326  020447                          .WORD   MENASC
020330  177777                          .WORD   -1
020332  000000                          .WORD   T$LOLIM
020334  177777                          .WORD   T$HILIM
020336                          10001$:
020336                                  BNCOMPLETE      1$              ;RETRY IF ERROR
020336  103352                          BCC     1$
020340  013700  020476                  MOV     MENRES,R0               ;GET THE OPERATOR'S REPLY
020344  020001                          CMP     R0,R1                   ;COMPARE TO MAXIMUM ALLOWED
020346  101411                          BLOS    5$                      ;BRANCH IF OK
020350                                  PRINTF  #MENERR                 ;DISPLAY ERROR MESSAGE
020350  012746  020374                  MOV     #MENERR,-(SP)
020354  012746  000001                  MOV     #1,-(SP)
020360  010600                          MOV     SP,R0
020362  104417                          TRAP    C$PNTF
020364  062706  000004                  ADD     #4,SP
020370  000735                          BR      1$                      ;RETRY
020372  000207                  5$:     RTS     PC                      ;RETURN TO CALLER
020374     045     116     045  MENERR: .ASCIZ  '%N%A *** Menu Selection Too Large ***
020442     045     116     045  SELASC: .ASCIZ  '%N%T'
020447     105     156     164  MENASC: .ASCIZ  'Enter Menu Selection:
                                        .EVEN
020476  000000                  MENRES:         .WORD   0
```

```
                                        .SBTTL  CHKMAN  - CHECK MANUAL INTERVENTION LEGALITY
                                ;+
                                ;
                                ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
                                ;
                                ;INPUT:
                                ;
                                ;       NONE.
                                ;
                                ;OUTPUT:
                                ;
                                ;       CARRY   0       MANUAL INTERVENTION NOT ALLOWED
                                ;               1       MANUAL INTERVENTION IS OK
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;       A MESSAGE IS DISPLAYED WARNING THAT TEST IS
                                ;       NOT EXECUTED IF MANUAL INTERVENTION IS NOT
                                ;       ALLOWED.
                                ;
                                ;
020500                          CHKMAN::
                                        SAVREG                  ;SAVE THE REGISTERS
020500                                  MANUAL                  ;SEE IF MANUAL INTERVENTION OK
020504  104450                          TRAP    C$MANI
020506                                  BCOMPLETE 1$            ;BRANCH IF ALLOWED
020506  103411                          BCS     1$
020510                                  PRINTF  #NOMAN          ;PRINT THE WARNING MESSAGE
020510  012746  020534                  MOV     #NOMAN,-(SP)
020514  012746  000001                  MOV     #1,-(SP)
020520  010600                          MOV     SP,R0
020522  104417                          TRAP    C$PNTF
020524  062706  000004                  ADD     #4,SP
020530  000241                          CLC                     ;CLEAR CARRY FOR ERROR
020532  000207                  1$:     RTS     PC              ;RETURN

020534     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.
                                        ;
020630                                  ENVIRN: MEMORY  R0
020630  104431                                  TRAP    C$MEM
020632  010037  003122                          MOV     R0,FREE         ; GET 1ST FREE ADDRESS...
020636  062737  000002  003122                  ADD     #2,FREE
020644  011037  003124                          MOV     (R0),FRESIZ     ;...AND WORD COUNT.
020650  162737  000004  003124                  SUB     #4,FRESIZ
020656  013702  002012                          MOV     L$UNIT,R2       ; GET NUMBER OF UNITS
020662  162737  000007  003124          10$:    SUB     #7,FRESIZ       ; TAKE AWAY 7 WORDS PER UNIT
020670  005302                                  DEC     R2
020672  001373                                  BNE     10$
020674  013700  003122                          MOV     FREE,R0         ;GET FIRST FREE ADDRESS
020700  063700  003124                          ADD     FRESIZ,R0       ;POINT TO LAST FREE ADDRESS
020704  162700  000002                          SUB     #2,R0           ;BACKUP 1 WORD
020710  010037  003126                          MOV     R0,FREEHI       ;STORE LAST FREE ADDRESS
020714  000240                                  NOP                     ;*****************************************
020716  012701  177520                          MOV     #BDVPCR,R1      ;GET BDV11 PCR ADDRESS
020722  010102                                  MOV     R1,R2           ;COPY TO R2
020724  062702  000002                          ADD     #2,R2           ;SET THE RANGE
020730  004737  016376                          JSR     PC,XNXM         ;SEE IF WE HAVE ONE
020734  103001                                  BCC     15$             ;OK TO SET FLAGS
020736  000423                                  BR      40$             ;RETURN WITH FLAGS CLEAR
020740  013701  177520          15$:            MOV     BDVPCR,R1       ;SAVE PCR CONTENTS
020744  062701  000001                          ADD     #1,R1           ;ADD ONE TO IT
020750  012702  177520                          MOV     #BDVPCR,R2      ;GET BDV11 PCR ADDRESS
020754  005212                                  INC     (R2)            ;TRY TO WRITE TO IT
020756  013703  177520                          MOV     BDVPCR,R3       ;GET RESULTS
020762  020103                                  CMP     R1,R3           ;DID IT CHANGE?
020764  001006                                  BNE     20$             ;NO, MUST BE 11/23B
020766  005237  003142                          INC     T23A            ;SET THE FLAG
020772  042737  170000  002120                  BIC     #170000,L$HIME  ;SUPERVISOR COULD BE WRONG
                                        ;       NOP                     ;BR 40$ FOR RELEASE
                                        ;       PRINTF  #M8186          ;TELL THE SYSTEM TYPE
021000  000402                                  BR      40$             ;RETURN
021002  005237  003144          20$:            INC     T23B            ;SET THE FLAG
                                        ;       NOP                     ;BR 40$ FOR RELEASE
                                        ;       PRINTF  #M8189          ;TELL THE SYSTEM TYPE
021006                          40$:
021006  000207                                  RTS     PC              ;RETURN
```

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

021010                          KTINIT:
021010  005037  003130                  CLR     KTFLG           ; INIT >28K MEMORY FLAG
021014  005037  003132                  CLR     KTENABLE        ; INIT TEST >28K FLAG
021020  023727  002120  001577          CMP     L$HIME,#1577    ; GOT ENOUGH MEMORY (>28K)?
021026  101444                          BLOS    9$              ; NO.
021030  013700  000004                  MOV     @#ERRVEC,R0     ; SAVE OLD ERR VEC PTR.
021034  012737  021126  000004          MOV     #2$,@#ERRVEC    ; SET ERR VEC PTR.
021042  005737  177572                  TST     @#SR0           ; GOT KT11?
021046  000240                          NOP                     ;  (TRAP IF NO).
021050  013737  002120  003130          MOV     L$HIME,KTFLG    ; YES. SET KT FLAG.
021056  042737  000177  003130          BIC     #177,KTFLG      ;
021064  010037  000004                  MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
02107C  005000                          CLR     R0              ; R0 = AR DATA.
021072  012701  172340                  MOV     #KIPAR0,R1      ; R1 = KI REGS PTR.
021076  012761  077406  177740  1$:     MOV     #77406,-40(R1)  ; SET DESCRIPTOR REG.
021104  010021                          MOV     R0,(R1)+        ; SET KIPAR REG.
021106  062700  000200                  ADD     #200,R0         ; BUMP AR DATA BY "4K".
021112  020027  002000                  CMP     R0,#2000        ; AT "I/O"?
021116  001367                          BNE     1$              ; NO.
021120  012741  177600                  MOV     #177600,-(R1)   ; YES. SET KTPAR7 FOR I/O.
021124  000405                          BR      9$

021126  012716  021134          2$:     MOV     #6$,(SP)        ; SET UP RETURN
021132  000002                          RTI                     ; RTI TO NEXT LOCATION

021134  010037  000004          6$:     MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.

021140  000207                  9$:     RTS     PC
```

```
                                        ;+
                                        ;       SUBROUTINE TO SET EXTENDED FEATURES SWITCH
                                        ;
                                        ;       Requires that SOFINIT and WRTCHR have been done previous to call.
                                        ;
                                        ;
                                        ;INPUTS:
                                        ;       R5      CURRENT UNIT NUMBER
                                        ;OUTPUTS:
                                        ;       The Extended Features Switch is set.
                                        ;
                                        ;-

021142                                  INVERT::

021142  005737  002224                          TST     EXTFEA                  ; IS SWITCH SET?
021146  001020                                  BNE     1$                      ; YES,EXIT STAGE RIGHT!(or the next one outa town!)
021150  012737  100206  021214                  MOV     #100206,CMDPKT          ; WRT SUB-SYS MEM CMD
021156  012737  021224  021216                  MOV     #WSMBK,CMDPKT+2         ; MSG BUF ADDR
021164  012737  000006  021222                  MOV     #6,CMDPKT+6             ; BYTE COUNT
021172  012737  100010  021224                  MOV     #100010,WSMBK           ; INVERT THE SWITCH
021200  012704  021214                          MOV     #CMDPKT,R4              ; SET CMDPKT INTO R4
021204  004737  010662                          JSR     PC,WRTCHR               ; DO IT
021210  000207                          1$:     RTS     PC                      ; RETURN

                                        ;       COMMAND PACKET.

        021214                          .       .       <.+3>&177774    ;MUST BE ON MOD 4 BOUNDRY.

021214  000000                          CMDPKT:: 0                      ;1ST WORD IS TS05 COMMAND.
021216  000000                                   0                      ;2ND WORD IS THE BUFFER LOW ADDRESS.
021220  000000                                   0                      ;3RD WORD IS THE BUFFER HIGH ADDRESS.
021222  000000                                   0                      ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.

                                        ;       WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.

021224  000000                          WSMBK:: 0                       ;1ST WORD:: SEL 0
021226  000000                                  0                       ;2ND WORD:: SEL 2
021230  000000                                  0                       ;3RD WORD:: SEL 4
                                                .EVEN
                                        ;+
                                        ;       SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
                                        ;
                                        ;
                                        ;INPUTS:
                                        ;OUTPUTS:
                                        ;       The NXMFLG is set if we can test.
                                        ;       The NXMLO and NXMHI addresses are setup.
                                        ;-

021232                                  MEMCK::

021232                                          SAVREG                          ;SAVE THE REGISTERS
021236  005037  003134                          CLR     NXMFLG                  ;CLEAR THE FLAG
021242  005037  003136                          CLR     NXMLO                   ;CLEAR THE TEST ADDRESS LO
021246  005037  003140                          CLR     NXMHI                   ;CLEAR THE TEST ADDRESS HI
021252  005737  003144                          TST     T23B                    ;IS IT A 11/23B?
```

```
021256  001407                          BEQ     1$                      ;NO
021260  023727  002120  007777          CMP     L$HIME,#7777            ; GREATER THAN 128K
021266  103406                          BLO     2$                      ; NO
021270  004737  021406                  JSR     PC,NXMTST               ;SETUP THE ADDRESS
021274  000427                          BR      13$                     ;SET THE FLAG AND EXIT
021276  005737  003142          1$:     TST     T23A                    ;IS IT A 11/23A?
021302  001413                          BEQ     4$                      ;NO
021304  023727  002120  005777  2$:     CMP     L$HIME,#5777            ;GREATER THAN 96K
021312  101023                          BHI     14$                     ;YES,23A/23B WITH 128K MEMORY
021314  023727  002120  003777          CMP     L$HIME,#3777            ;GREATER THAN 64K BUT LESS THAN 92K?
021322  103403                          BLO     4$                      ;NO, CHECK 24K
021324  004737  021406                  JSR     PC,NXMTST               ;SETUP THE ADDRESS
021330  000411                          BR      13$                     ;SET THE FLAG AND EXIT
021332  023727  002120  001577  4$:     CMP     L$HIME,#1577            ;GREATER THAN 24K BUT LESS THAN 64K?
021340  103410                          BLO     14$                     ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
021342  004737  021406                  JSR     PC,NXMTST               ;SETUP THE ADDRESS
021346  062737  000077  003140          ADD     #77,NXMHI               ;FOOL THE 11/02 & 11/03
021354  005237  003134          13$:    INC     NXMFLG                  ;SET THE FLAG
021360  000411                          BR      15$                     ;EXIT
021362  000410                  14$:    BR      15$                     ;NOP FOR PRINTOUT
021364                                  PRINTF  @NOMEM                  ;TELL THEM & EXIT ***NO PRINT****
021364  012746  005456                  MOV     @NOMEM,-(SP)
021370  012746  000001                  MOV     #1,-(SP)
021374  010600                          MOV     SP,R0
021376  104417                          TRAP    C$PNTF
021400  062706  000004                  ADD     #4,SP
021404  000207                  15$:    RTS     PC                      ;RETURN

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

021406  013701  002120          NXMTST: MOV     L$HIME,R1               ;GET TOP OF MEMORY
021412  062701  000200                  ADD     #200,R1                 ;MAKE IT I/O BLOCK OR OTHER NXM
021416  042701  000177                  BIC     #177,R1
021422  010102                          MOV     R1,R2                   ;RESAVE RESULTS
        000006                          .REPT   6
                                        ASL     R1                      ;PUT IN PLACE FOR XFER
                                        .ENDR
021440  010137  003136                  MOV     R1,NXMLO                ;SAVE TEST ADDRESS LOW
        000012                          .REPT   10.
                                        ASR     R2                      ;PUT IN PLACE FOR XFER
                                        .ENDR
021470  042702  177700                  BIC     #177700,R2              ;DON'T WANT ILA!
021474  010237  003140                  MOV     R2,NXMHI                ;SAVE TEST ADDRESS HIGH
021500  000207                          RTS     PC                      ;RETURN

021502                                  ENDMOD
```

F 8


```
                                                .TITLE  TSV4   MISCELLANEOUS SECTIONS

 021502                                         BGNMOD  TSV4
 021502                                 TSV4::
 021502
 021502
 021502
 021502
 021502                                         .SBTTL  PROTECTION TABLE
 021502                                         BGNPROT
 021502                                 L$PROT::
 021502  177777  177777  177777                 .WORD   -1, -1,  1,  1          ;NO DEVICE PROTECTION REQUIRED.
 021512                                         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
                                ;-
021512                                  BGNINIT
021512                          L$INIT::
021512  005037  002224          40$:    CLR     EXTFEA
021516  005037  003134                  CLR     NXMFLG
021522  012737  006356  002176          MOV     #EPRT1,EPRTSW           ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
021530  005037  003152                  CLR     SIFLAG                  ;CLEAR "SOFT INIT" FLAG
021534  005037  003132                  CLR     KTENABLE                ;CLEAR TEST ABOVE 28K FLAG
021540  005037  002300                  CLR     RAMSIZ                  ;CLEAR RAM SIZE FOR RAMERR ROUTINE
021544                                  READEF  #EF.CONTINUE
021544  012700  000036                  MOV     #EF.CONTINUE,R0
021550  104447                          TRAP    C$REFG
021552                                  BNCOMPLETE 1$
021552  103023                          BCC     1$
021554  023737  002200  002012          CMP     UNITN,L$UNIT            ;UNIT IN RANGE?
021562  103070                          BHIS    4$                      ;BR IF NO.
021564  005737  003110                  TST     DUFLG                   ;DROPPED UNIT?
021570  100472                          BMI     NXTU                    ;BR IF YES
021572  013701  002200                  MOV     UNITN,R1
021576  006301                          ASL     R1
021600  005761  003174                  TST     ERTABL(R1)
021604  001516                          BEQ     SETU
021606  032761  040000  003174          BIT     #BIT14,ERTABL(R1)       ;DROPPED?
021614  001060                          BNE     NXTU
021616                                  EXIT    INIT                    ;DO NOTHING IF "CONTINUE".
021616  104432                          TRAP    C$EXIT
021620  000416                          .WORD   L10030-.
021622                          1$:     READEF  #EF.NEW
021622  012700  000035                  MOV     #EF.NEW,R0
021626  104447                          TRAP    C$REFG
021630                                  BNCOMPLETE NXTU                 ;TAKE NEXT UNIT IF NOT NEW PASS.
021630  103052                          BCC     NXTU
021632                                  READEF  #EF.START
021632  012700  000040                  MOV     #EF.START,R0
021636  104447                          TRAP    C$REFG
021640                                  BCOMPLETE 2$
021640  103404                          BCS     2$
021642                                  READEF  #EF.RESTART
021642  012700  000037                  MOV     #EF.RESTART,R0
021646  104447                          TRAP    C$REFG
021650                                  BNCOMPLETE 31$
021650  103031                          BCC     31$
021652                          2$:                                     ;1ST PASS, BUS-INIT...
021652                                  BRESET                          ;BUS RESET.
021652  104433                          TRAP    C$RESET
```

```
021654  005037  002212                  CLR     TSTCNT                  ;NUMBER OF TESTS RUN IN PASS
021660  005037  002220                  CLR     FATFLG                  ;CLEAR FATAL ERROR COUNT
021664  005037  003142                  CLR     T23A                    ;CLEAR 11/23A FLAG
021670  005037  003144                  CLR     T23B                    ;CLEAR 11/23B FLAG
                                ;       MOV     #340,-(SP)
                                ;       MOV     #20$,-(SP)              ;RETURN TO DEBUGGER
                                ;       JMP     O.ODT           ;;@@ENTER THE DEBUGGER
021674  005037  003376                  CLR     SKIPT           ;CLEAR THE SUBTEST "SKIPPER"
021700                          20$:
021700  012737  177777  002202          MOV     #-1,QVP                 ;...QUICK VERIFY...
021706  004737  020630                  JSR     PC,ENVIRN               ;SET ENVIRONMENT.
021712  004737  021010                  JSR     PC,KTINIT               ;INITIALIZE KT MEMORY MANAGEMENT
021716  012700  003174                  MOV     #ERTABL,R0
021722  005020                  30$:    CLR     (R0)+                   ;CLEAR THE ERROR TABLE
021724  020027  003374                  CMP     R0,#ERTABE
021730  103774                          BLO     30$
021732  000404                          BR      4$
021734  005037  002202          31$:    CLR     QVP
021740  000137  022010                  JMP     PASRPT                  ;GO REPORT THE STATUS

021744                          4$:
021744  012737  177777  002200  NEWPAS: MOV     #-1,UNITN               ;INIT UNIT NUMBER...
021752  005037  002216                  CLR     DEVCNT                  ;CLEAR COUNT OF DEVICES RUNNING
021756                          NXTU:   BREAK
021756  104422                          TRAP    C$BRK
021760  005237  002200                  INC     UNITN                   ;...AND SET NEXT UNIT NUMBER.
021764  023737  002200  002012          CMP     UNITN,L$UNIT
021772  103423                          BLO     SETU
021774  012737  177777  003110          MOV     #-1,DUFLG
022002  000401                          BR      11$
022004                                  DOCLN                           ;ABORT, NO MORE UNITS.
022004  104444                          TRAP    C$DCLN
022006  000240                  11$:    NOP
022010                          PASRPT:
022010  023727  002012  000001          CMP     L$UNIT,#1               ;HOW MANY UNITS SELECTED?
022016  101752                          BLOS    NEWPAS                  ;BR IF ONLY 1
022020  005737  002216                  TST     DEVCNT                  ;ARE ANY STILL RUNNING?
022024  001747                          BEQ     NEWPAS                  ;BR IF NO
022026                                  RFLAGS  R0
022026  104421                          TRAP    C$RFLA
022030  032700  000100                  BIT     #ISR,R0                 ;SHOULD WE PRINT STATISTICS
022034  001343                          BNE     NEWPAS                  ;BR IF NO

022036                                  DORPT
022036  104424                          TRAP    C$DRPT
022040  000741                          BR      NEWPAS
022042                          10$:

022042                          SETU:   GPHARD  UNITN,R0                ;GET UNIT N P TABLE POINTER.
022042  013700  002200                  MOV     UNITN,R0
022046  104442                          TRAP    C$GPHRD
022050                                  BNCOMPLETE NXTU                 ;BR IF UNIT NOT AVAILABLE.
022050  103342                          BCC     NXTU
022052  005037  003110                  CLR     DUFLG                   ;CLEAR "DROPPED" FLAG.
022056  005237  002216                  INC     DEVCNT
022062  012001                          MOV     (R0)+,R1                ;GET 1ST REGISTER ADDRESS.
022064  010137  002204                  MOV     R1,CSRADDR              ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
```

```
022070  012001                          MOV     (R0)+,R1                ;GET VECTOR ADDRESS.
                                        ;MOV    (R0),R2                 ;GET INTERRUPT PRIORITY
                                        ;MOV    R2,IPRI                 ;SET INTERRUPT PRIORITY.
022072  010137  002206                  MOV     R1,IVEC                 ;SET INTERRUPT VECTOR POINTER...
022076  012721  016216                  MOV     #INTR,(R1)+             ;...VECTOR...
022102  013721  002210                  MOV     IPRI,(R1)+              ;...AND PRIORITY.

022106                          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.
                                ;
022106  013701  002200                  MOV     UNITN,R1
022112  006301                          ASL     R1
022114  052761  100000  003174          BIS     #BIT15,ERTABL(R1)       ;SAY DEVICE RUNNING
022122  005037  005770                  CLR     EXTA                    ;CLEAR ERROR EXTENSION FLAG.
022126  023727  002012  000001          CMP     L$UNIT,#1               ;ARE WE TESTING MULTIPLE UNITS?
022134  101416                          BLOS    10$                     ;BR IF NO.
022136                                  RFLAGS  R0                      ;YES -- GET OPERATOR FLAGS.
022136  104421                          TRAP    C$RFLA
022140  032700  001000                  BIT     #PNT,R0                 ;SHOULD WE PRINT UNIT #?
022144  001412                          BEQ     10$                     ;BR IF NOT.
022146                                  PRINTF  #PUNIT,UNITN            ;PRINT THE UNIT #
022146  013746  002200                  MOV     UNITN,-(SP)
022152  012746  022240                  MOV     #PUNIT,-(SP)
022156  012746  000002                  MOV     #2,-(SP)
022162  010600                          MOV     SP,R0
022164  104417                          TRAP    C$PNTF
022166  062706  000006                  ADD     #6,SP
022172                          10$:
022172  005037  003112                  CLR     NODEV
022176  013701  002204                  MOV     CSRADDR,R1              ;ADDRESS OF FIRST REGISTER
022202  010102                          MOV     R1,R2                   ;START OF REGISTERS
022204  062702  000002                  ADD     #TSSR,R2                ;ADDRESS OF TSSR REGISTER
022210  004737  016376                  JSR     PC,XNXM                 ;TEST BOTH CONTROLLER REGISTERS...
022214  103005                          BCC     2$                      ;...AND BR IF ALL OK.
022216  010137  003112                  MOV     R1,NODEV                ;FLAG DEVICE AS NON-EXISTENT
022222  012737  177777  003110          MOV     #-1,DUFLG               ;DROP THIS UNIT.
022230                          2$:
                                ;
                                ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
                                ;
022230                          5$:     SETPRI  #PRI00          ;ENABLE INTERRUPTS.
022230  012700  000000                  MOV     #PRI00,R0
022234  104441                          TRAP    C$SPRI
022236                                  ENDINIT
022236                          L10030:
022236  104411                          TRAP    C$INIT

022240     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.
                                ;--
022306                                  BGNAU
022306                          L$AU::
022306  010001                          MOV     R0,R1                   ; GET UNIT TO BE ADDED (R0)
022310  006301                          ASL     R1                      ; MAKE IT A WORD INDEX
022312  052761  100000  003174          BIS     #100000,ERTABL(R1)      ; SET THE "ACTIVE" BIT
022320  042761  040000  003174          BIC     #40000,ERTABL(R1)       ; CLEAR THE "DROPPED" BIT
022326                                  PRINTF  #1$,R0
022326  010046                          MOV     R0,-(SP)
022330  012746  022354                  MOV     #1$,-(SP)
022334  012746  000002                  MOV     #2,-(SP)
022340  010600                          MOV     SP,R0
022342  104417                          TRAP    C$PNTF
022344  062706  000006                  ADD     #6,SP
022350                                  EXIT    AU
022350  000167                          .WORD   J$JMP
022352  000026                          .WORD   L10031-2-.
022354     045     116     045  1$:    .ASCIZ  /<N><A  UNIT <D><A ADDED/
                                        .EVEN

022402                                  ENDAU           ; UNUSED.
022402                          L10031:
022402  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.

022404                                  BGNDU
022404                          L$DU::
022404  012737  177777  003110          MOV     #-1,DUFLG
022412  010001                          MOV     R0,R1
022414  006301                          ASL     R1
022416  052761  140000  003174          BIS     #140000,ERTABL(R1)      ; SAY DROPPED
022424  000240  000240  000240          240,240,240                     ; ?????????
022432                                  PRINTF  #1$,R0
022432  010046                          MOV     R0,-(SP)
022434  012746  022460                  MOV     #1$,-(SP)
022440  012746  000002                  MOV     #2,-(SP)
022444  010600                          MOV     SP,R0
022446  104417                          TRAP    C$PNTF
022450  062706  000006                  ADD     #6,SP
022454                                  EXIT    DU
022454  000167                          .WORD   J$JMP
022456  000030                          .WORD   L10032-2 .
```

```
022460     045     116     045  1$:     .ASCIZ  /%N%A  UNIT %D%A DROPPED/
                                        .EVEN
022510                                  ENDDU
022510                          L10032:
022510  104453                          TRAP    C$DU
                                ;++
                                ; AUTO-DROP CODE SECTION.
                                ;
022512                                  BGNAUTO
022512                          L$AUTO::
022512  013705  002204                  MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
022516  012703  000550                  MOV     #360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
022522  004737  016250          10$:    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
022526  103420                          BCS     20$                     ;LEAVE WHEN SSR IS SET
022530                                  DELAY   250.                    ;WAIT FOR .25 SECONDS
022530  012727  000372                  MOV     #250.,(PC)+
022534  000000                          .WORD   0
022536  013727  002116                  MOV     L$DLY,(PC)+
022542  000000                          .WORD   0
022544  005367  177772                  DEC     -6(PC)
022550  001375                          BNE     .-4
022552  005367  177756                  DEC     -22(PC)
022556  001367                          BNE     .-20
022560  005303                          DEC     R3                      ;BUMP COUNTER DOWN
022562  001357                          BNE     10$                     ;KEEP GOING
022564  004737  017202                  JSR     PC,CKDROP               ;TRY AND DROP UNIT
022570                          20$:
022570                                  ENDAUTO                 ; UNUSED.
022570                          L10033:
022570  104461                          TRAP    C$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.
                                ;
022572                                  BGNCLN
022572                          L$CLEAN::
022572  013705  002204                  MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
022576  005737  003110                  TST     DUFLG           ;"DROPPED" FLAG IS SET ON...
022602  100405                          BMI     1$              ;...AND GROSS CONTROLLER FAULT...
                                                                ;...DON'T TRY TO XCT CLEANUP CODE.

022604  012765  000000  000002          MOV     #0,TSSR(R5)     ;DO SOFT INIT
022612  004737  016250                  JSR     PC,WAITF
022616                          1$:
022616                          2$:     ENDCLN
022616                          L10034:
022616  104412                          TRAP    C$CLEAN
                                ;**
                                ; THE REPORT CODING SECTION CONTAINS THE
                                ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;--
022620                                  BGNRPT
022620                          L$RPT::
022620                                  PRINTS  #DEVSUM
022620  012746  023062                  MOV     #DEVSUM,-(SP)
022624  012746  000001                  MOV     #1,-(SP)
022630  010600                          MOV     SP,R0
022632  104416                          TRAP    C$PNTS
022634  062706  000004                  ADD     #4,SP
022640  010246                          MOV     R2,-(SP)
022642  010346                          MOV     R3,-(SP)
022644  010446                          MOV     R4,-(SP)
022646  012704  003174                  MOV     #ERTABL,R4      ; GET START OF ERROR TABLE.
022652  005003                          CLR     R3              ; CLEAR UNIT NUMBER
022654  011402                  1$:     MOV     (R4),R2         ; GET ERROR TABLE ENTRY & TEST IT.
022656  001467                          BEQ     4$              ; ZERO IF UNIT NOT RUN
022660  100066                          BPL     4$
022662  032702  040000                  BIT     #BIT14,R2       ; WAS UNIT DROPPED?
022666  001015                          BNE     2$              ; BR IF YES
022670  042702  170000                  BIC     #^C7777,R2      ; GET ERROR COUNT FIELD
022674                                  PRINTS  #DEVONL,R3,R2   ; PRINT
022674  010246                          MOV     R2,-(SP)
022676  010346                          MOV     R3,-(SP)
022700  012746  023117                  MOV     #DEVONL,-(SP)
022704  012746  000003                  MOV     #3,-(SP)
022710  010600                          MOV     SP,R0
022712  104416                          TRAP    C$PNTS
022714  062706  000010                  ADD     #10,SP
022720  000446                          BR      4$
022722  020227  160000          2$:     CMP     R2,#160000      ; WAS UNIT NON-EXISTENT?
022726  001012                          BNE     3$              ; BR IF NO
022730                                  PRINTS  #DEVNXR,R3
022730  010346                          MOV     R3,-(SP)
022732  012746  023167                  MOV     #DEVNXR,-(SP)
```

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

```
                                        .TITLE  TSV5A   HARDWARE TESTS

023402                                  BGNMOD  TSV5
023402                          TSV5::

                                        .SBTTL  TEST  1: BUS RESET TEST

                                ;        THIS TEST VERIFIES THAT THE M7196 MODULE'S DEVICE REGISTERS ARE
                                ;        ACCESSIBLE ON THE BUS (SUBTEST 1) AND THEN CHECKS THAT THE
                                ;        BUILT-IN INITIALIZATION SELF-TEST MICRODIAGNOSTIC DID NOT FIND
                                ;        ANY BASIC PROBLEMS IN THE MODULE.  AREAS OF LOGIC TESTED BY THE
                                ;        SELF-TEST SEQUENCE ARE AS FOLLOWS:  ROM AND PIPELINE REGISTER,
                                ;        SEQUENCER, INTERNAL BUSES, 2901 MICROPROCESSOR, AND, RAM.  THIS
                                ;        TEST INITIALIZES THE CONTROLLER BY ISSUING THE BUS INIT SIGNAL
                                ;        VIA A RESET INSTRUCTION, OR BY WRITING INTO THE TSSR REGISTER,
                                ;        WAITS A PERIOD OF TIME (TO ALLOW THE CONTROLLER'S INITIALIZATION
                                ;        MICRODIAGNOSTIC SEQUENCE TO BE COMPLETED), AND THEN CHECKS THE
                                ;        CONTENTS OF THE TSSR REGISTER.  SUCCESSFUL INITIALIZATION IS
                                ;        INDICATED BY SUBSYSTEM READY (SSR) AND NEED BUFFER ADDRESS (NBA)
                                ;        BITS BEING SET (1) AND ALL OTHER BITS (EXCEPT A17 AND A16 AND
                                ;        OFL, WHICH ARE IGNORED FOR THIS TEST) BEING CLEAR (0).  IF THE
                                ;        CONTENTS OF TSSR ARE NOT AS EXPECTED, AN ERROR REPORT IS ISSUED
                                ;        LISTING THE EXPECTED DATA, ACTUAL DATA, AND THE DISCREPANCIES.
                                ;        THE ERROR REPORT ANALYZES THE TSSR CONTENTS AND DISCERNS AND
                                ;        REPORTS ONE OF THREE POSSIBILITIES:
                                ;
                                ;
                                ;             1.  TSSR CONTENTS ARE AMBIGUOUS (ANY OF BITS 11-14 ARE SET,
                                ;                 OR STATES OF SSR AND SC BITS DO NOT CORRESPOND TO THE
                                ;                 APPARENT ERROR CODE IN BITS 0-5):  INDICATES THAT THE
                                ;                 TSSR CONTENT CANNOT BE TRUSTED.  INDICATES A
                                ;                 CATASTROPHIC CONTROLLER MALFUNCTION.  THIS IS A FATAL
                                ;                 ERROR (EXECUTION IS ABORTED).  FIELD ACTION WOULD BE TO
                                ;                 REPLACE THE M7196.  IF THE M7196 ITSELF IS BEING
                                ;                 DEBUGGED, THE PROGRAM SHOULD BE RESTARTED WITH LOOP ON
                                ;                 ERROR ENABLED IN ORDER TO PROBE FOR THE PROBLEM.
                                ;
                                ;             2.  SSR = 0, SC = 0 AND THE ERROR CODE IN BITS 0-5 IS IN
                                ;                 THE RANGE 17-13:  THIS IS A FATAL ERROR.  THE ERROR
                                ;                 CODE IS DECODED AND THE APPROPRIATE DESCRIPTION GIVEN.
                                ;                 INDICATES THAT A SERIOUS PROBLEM EXISTS.
                                ;

023402                                  BGNTST
023402                                                                          T1::
023402  012700  023600                  MOV     #TST1ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
023406  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
023412  012737  000024  002214          MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
023420                          T1LOOP:
023420  005003                          CLR     R3                      ;USE R3 AS FATAL ERROR FLAG

023422                                  BGNSUB                          ;///////////// BEGIN SUBTEST ////////////
023422                                                                          T1.1:
023422  104402                                                                          TRAP    C$BSUB
```

```
 76
 77 023424                          BRESET                          ;ISSUE A BUS RESET
    023424  104433                                                                          TRAP    C$RESET
 78 023426  004737  016250          JSR     PC,WAITF                ;WAIT FOR READY
 79 023432  016501  000002          MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 80 023436  010102                  MOV     R1,R2                   ;CONTENTS OF TSSR
 81 023440  042702  176277          BIC     #^C<HIADDR!OFL>,R2      ;THESE BITS MAY BE SET
 82 023444  052702  002200          BIS     #SSR!NBA,R2             ;READY AND NEW DATA SHOULD BE SET
 83 023450  020102                  CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
 84 023452  001405                  BEQ     10$                     ;BRANCH IF COMPARE
 88 023454                          ERRDF   ERRNO,SFHERR,SFFMSG     ;REPORT A FATAL ERROR
    023454  104455                                                                          TRAP    C$ERDF
    023456  000145                                                                          .WORD   101
    023460  003703                                                                          .WORD   SFHERR
    023462  012102                                                                          .WORD   SFFMSG
 89 023464  005203                  INC     R3                      ;SET THE FATAL ERROR FLAG
 90 023466                  10$:
 91 023466                          ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
    023466                                                                          L10037:
    023466  104403                                                                          TRAP    C$ESUB
 92
 93 023470  005703                  TST     R3                      ;DID WE HAVE FATAL ERROR ?
 94 023472  001402                  BEQ     20$                     ;BRANCH IF NOT
 95 023474  004737  017202          JSR     PC,CKDROP               ;GO DROP THIS UNIT, IF ALLOWED
 96 023500  005003          20$:    CLR     R3                      ;RESET FATAL ERROR FLAG
 97
 98
 99 023502                          BGNSUB                          ;///////////// BEGIN SUBTEST ///////////
    023502                                                                          T1.2:
    023502  104402                                                                          TRAP    C$BSUB
100
101 023504  005065  000002          CLR     TSSR(R5)                ;WRITE TO ISSUE A SOFT RESET
102 023510  004737  016250          JSR     PC,WAITF                ;WAIT FOR READY TO SET
103 023514  016501  000002          MOV     TSSR(R5),R1             ;GET REGISTER TSSR DATA
104 023520  010102                  MOV     R1,R2                   ;CONTENTS OF TSSR
105 023522  042702  176277          BIC     #^C<HIADDR!OFL>,R2      ;THESE BITS MAY BE SET
106 023526  052702  002200          BIS     #SSR!NBA,R2             ;READY AND NEW DATA SHOULD BE SET
107 023532  020102                  CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
108 023534  001405                  BEQ     10$                     ;BRANCH IF COMPARE
112 023536                          ERRDF   ERRNO,SFIERR,SFFMSG     ;REPORT A FATAL ERROR
    023536  104455                                                                          TRAP    C$ERDF
    023540  000146                                                                          .WORD   102
    023542  003650                                                                          .WORD   SFIERR
    023544  012102                                                                          .WORD   SFFMSG
113 023546  005203                  INC     R3                      ;SET THE ERROR FLAG
114 023550                  10$:
115 023550                          ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
    023550                                                                          L10040:
    023550  104403                                                                          TRAP    C$ESUB
116
117
118 023552  005703                  TST     R3                      ;FATAL ERROR DETECTED ?
119 023554  001402                  BEQ     20$                     ;BRANCH IF NOT
120 023556  004737  017202          JSR     PC,CKDROP               ;SEE IF TIME TO DROP UNIT
121 023562  004737  016456  20$:    JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
122 023566  103002                  BCC     40$                     ;BRANCH IF NOT
123 023570  000137  023420          JMP     T1LOOP                  ;LOOP UNTIL COUNT EXPIRED
```

```
023574                          40$:    EXIT    TST                     ;ALL DONE THIS TEST
023574  104432                                                                          TRAP    C$EXIT
023576  000022                                                                          .WORD   L10036 .

                                ;+
                                ;LOCAL TEXT MESSAGES FOR TEST
                                ;-

023600     111     156     151  TST1ID: .ASCIZ  'Initialization'
                                        .EVEN
023620                                  ENDTST
023620                                                                  L10036:
023620  104401                                                                          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).
                                ;
023622                                  BGNTST
023622                                                                  T2::
023622  012700  024270                  MOV     #TST2ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
023626  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
023632  012737  000024  002214          MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
023640  005004                  T2LOOP: CLR     R4                      ;STARTING DATA PATTERN
023642  012703  177777                  MOV     #-1,R3                  ;DO INIT ON FIRST TIME THROUGH
023646  005703                  5$:     TST     R3                      ;DO WE NEED SOFT INIT
```

```
023650  001412                          BEQ     10$                     ;BRANCH IF NOT
023652  005003                           LR     R3                      ;DON'T NEED INIT NEXT TIME THRU
023654  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
023660  103406                          BCS     10$                     ;BR IF SOFT INIT = OK
023662  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
023664                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
023664  104455                                                                                  TRAP    C$ERDF
023666  000311                                                                                  .WORD   201
023670  003650                                                                                  .WORD   SFIERR
023672  012034                                                                                  .WORD   SFIMSG
023674  005203                          INC     R3                      ;FORCE SOFT INIT ON NEXT PASS
023676  005037  002220          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG

023702                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
023702  104404                                                                                  TRAP    C$BSEG

023704  110465  000001                  MOVB    R4,TSDBH(R5)            ;SET MAINT MODE + WRITE DATA
023710  004737  016250                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
023714  103411                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
023716  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
023720  010402                          MOV     R4,R2                   ;DATA THAT WAS WRITTEN
023722                                  ERRDF   ERRNO,T2SSR,EXPREC      ;DEVICE FATAL SSR FAILED TO SET
023722  104455                                                                                  TRAP    C$ERDF
023724  000312                                                                                  .WORD   202
023726  024216                                                                                  .WORD   T2SSR
023730  015474                                                                                  .WORD   EXPREC
023732  005203                          INC     R3                      ;FORCE SOFT INIT ON NEXT PASS
023734  005237  002220                  INC     FATFLG                  ;SET FATAL ERROR FLAG
023740                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
023740  104406                                                                                  TRAP    C$CLP1
023742  005737  002220                  TST     FATFLG                  ;WAS FATAL ERROR RECEIVED ?
023746  001402                          BEQ     20$                     ;BRANCH IF NOT
023750  004737  017202                  JSR     PC,CKDROP               ;SEE IF TIME TO DROP UNIT
023754  010402                  20$:    MOV     R4,R2                   ;DATA PATTERN WRITTEN
023756  042702  177774                  BIC     #^C<BIT0!BIT1>,R2       ;CLEAR ALL BUT LOW 2 BITS
023762  000302                          SWAB    R2                      ;BITS 8 AND 9 HAVE LOW DATA BITS
023764  052702  002200                  BIS     #SSR!NBA,R2             ;THESE BITS MUST BE SET ALSO
023770  016501  000002                  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
023774  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
024000  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
024002  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
024006  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
024010  001405                          BEQ     30$                     ;OKAY IF MATCH
024012                                  ERRHRD  ERRNO,T2TSSR,EXPREC     ;TSSR WASN'T CORRECT
024012  104456                                                                                  TRAP    C$ERHRD
024014  000313                                                                                  .WORD   203
024016  024151                                                                                  .WORD   T2TSSR
024020  015474                                                                                  .WORD   EXPREC
024022  005203                          INC     R3                      ;FORCE SOFT INIT ON NEXT PASS
024024                          30$:    CKLOOP                          ;LOOP ON ERROR ?
024024  104406                                                                                  TRAP    C$CLP1
024026  016501  000000                  MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
024032  005002                          CLR     R2                      ;
024034  150402                          BISB    R4,R2                   ;DATA PATTERN WRITTEN
024036  000302                          SWAB    R2                      ;MOVE INTO TOP BYTE
024040  150402                          BISB    R4,R2                   ;BOTH HALVES SHOULD BE SAME
024042  020102                          CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
```

```
024044  001405                          BEQ     35$                     ;ERROR IF NOT EQUAL
024046                                  ERRHRD  ERRNO,T2TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
024046  104456                                                                          TRAP    C$ERHRD
024050  000314                                                                          .WORD   204
024052  024104                                                                          .WORD   T2TSBA
024054  015474                                                                          .WORD   EXPREC
024056  005203                          INC     R3                      ;FORCE SOFT INIT ON NEXT PASS

024060                          35$:    ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
024060                                                                                  10000$:
024060  104405                                                                          TRAP    C$ESEG

024062  105204                          INCB    R4                      ;NEXT DATA PATTERN TO TEST
024064  001270                          BNE     5$                      ;BRANCH TILL BACK TO ZERO
024066  004737  016456                  JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
024072  103002                          BCC     40$                     ;BRANCH IF NOT
024074  000137  023640                  JMP     T2LOOP                  ;LOOP UNTIL COUNT EXPIRED
024100                          40$:    EXIT    TST                     ;ALL DONE THIS TEST
024100  104432                                                                          TRAP    C$EXIT
024102  000214                                                                          .WORD   L10041 .

                                ;+
                                ;LOCAL TEXT MESSAGES FOR TEST
                                ;-

024104     124     123     102  T2TSBA: .ASCIZ  'TSBA Incorrect After TSDB High Write'
024151     124     123     123  T2TSSR: .ASCIZ  'TSSR Incorrect After TSDB High Write'
024216     116     157     040  T2SSR:  .ASCIZ  'No Sub-System Ready After TSDB High Write
024270     127     162     141  TST2ID: .ASCIZ  'Wrap Data - High Byte'
                                        .EVEN
024316                                  ENDTST
024316                                                                          L10041:
024316  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).
                                        ;
 024320                                         BGNTST
 024320                                                                                          T3::
 024320  012700  024763                         MOV     #TST3ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
 024324  004737  016510                         JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
 024330  012737  000024  002214                 MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
 024336  005004                          T3LOOP: CLR     R4                      ;STARTING DATA PATTERN
 024340  012703  000001                         MOV     #1,R3                   ;FORCE SOFT INIT ON FIRST PASS
 024344  005703                          5$:    TST     R3                      ;SET IF INIT IS REQUIRED
 024346  001412                                 BEQ     10$                     ;BRANCH IF NO INIT NEEDED
 024350  005003                                 CLR     R3                      ;ASSUME NO INIT NEEDED ON NEXT LOOP
 024352  004737  015774                         JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 024356  103406                                 BCS     10$                     ;BR IF SOFT INIT - OK
 024360  010001                                 MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 024362                                         ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
 024362  104455                                                                                  TRAP    C$ERDF
 024364  000455                                                                                  .WORD   301
 024366  003650                                                                                  .WORD   SFIERR
 024370  012034                                                                                  .WORD   SFIMSG
 024372  005203                                 INC     R3                      ;FORCE INIT ON NEXT PASS
 024374  005037  002220                  10$:   CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG

 024400                                         BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
 024400  104404                                                                                  TRAP    C$BSEG

 024402  110465  000000                         MOVB    R4,TSDB(R5)             ;SET MAINT MODE + WRITE DATA
 024406  004737  016250                         JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 024412  103411                                 BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 024414  010001                                 MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 024416  010402                                 MOV     R4,R2                   ;DATA THAT WAS WRITTEN
 024420                                         ERRDF   ERRNO,T3SSR,EXPREC      ;DEVICE FATAL SSR FAILED TO SET
 024420  104455                                                                                  TRAP    C$ERDF
 024422  000456                                                                                  .WORD   302
 024424  024712                                                                                  .WORD   T3SSR
 024426  015474                                                                                  .WORD   EXPREC
 024430  005203                                 INC     R3                      ;FORCE INIT ON NEXT PASS
 024432  005237  002220                         INC     FATFLG                  ;SET THE FATAL ERROR FLAG
 024436                                  15$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
 024436  104406                                                                                  TRAP    C$CLP1
 024440  005737  002220                         TST     FATFLG                  ;DID WE GET FATAL ERROR ?
 024444  001402                                 BEQ     20$                     ;BRANCH IF NOT
 024446  004737  017202                         JSR     PC,CKDROP               ;DROP UNIT, IF DROP ALLOWED
 024452  010402                          20$:   MOV     R4,R2                   ;DATA PATTERN WRITTEN
 024454  042702  177774                         BIC     #^C<BIT0!BIT1>,R2       ;CLEAR ALL BUT LOW 2 BITS
 024460  000302                                 SWAB    R2                      ;BITS 8 AND 9 HAVE LOW DATA BITS
 024462  052702  002200                         BIS     #SSR!NBA,R2             ;THESE BITS MUST BE SET ALSO
 024466  016501  000002                         MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 024472  032701  000100                         BIT     #OFL,R1                 ;IS DRIVE OFF-LINE
 024476  001402                                 BEQ     25$                     ;BRANCH IF NOT OFFLINE
 024500  052702  000100                         BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED
 024504  020201                          25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 024506  001405                                 BEQ     30$                     ;OKAY IF MATCH
 024510                                         ERRHRD  ERRNO,T3TSSR,EXPREC     ;TSSR WASN'T CORRECT
 024510  104456                                                                                  TRAP    C$ERHRD
```

```
     024512  000457                                                                    .WORD   303
     024514  024646                                                                    .WORD   T3TSSR
     024516  015474                                                                    .WORD   EXPREC
339  024520  005203                  INC     R3                      ;FORCE INIT ON NEXT PASS
340  024522                  30$:    CKLOOP                          ;LOOP ON ERROR ?
     024522  104406                                                                    TRAP    C$CLP1
341  024524  016501  000000          MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
342  024530  005002                  CLR     R2                      ;
343  024532  150402                  BISB    R4,R2                   ;DATA PATTERN WRITTEN
344  024534  000302                  SWAB    R2                      ;MOVE INTO TOP BYTE
345  024536  150402                  BISB    R4,R2                   ;BOTH HALVES SHOULD BE SAME
346  024540  020102                  CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
347  024542  001405                  BEQ     35$                     ;BR IF ERROR NOT EQUAL
351  024544                          ERRHRD  ERRNO,T3TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     024544  104456                                                                    TRAP    C$ERHRD
     024546  000460                                                                    .WORD   304
     024550  024602                                                                    .WORD   T3TSBA
     024552  015474                                                                    .WORD   EXPREC
352  024554  005203                  INC     R3                      ;FORCE INIT ON NEXT PASS
353
354  024556                  35$:    ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     024556                                                                          10000$:
     024556  104405                                                                    TRAP    C$ESEG
355
356  024560  105204                  INCB    R4                      ;NEXT DATA PATTERN TO TEST
357  024562  001270                  BNE     5$                      ;ALL DONE WHEN BACK TO ZERO
358  024564  004737  016456          JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
359  024570  103002                  BCC     40$                     ;BRANCH IF NOT
360  024572  000137  024336          JMP     T3LOOP                  ;LOOP UNTIL COUNT EXPIRED
361  024576                  40$:    EXIT    TST                     ;ALL DONE THIS TEST
     024576  104432                                                                    TRAP    C$EXIT
     024600  000210                                                                    .WORD   L10042-.
362
363                                  ;+
364                                  ;LOCAL TEXT MESSAGES FOR TEST
365                                  ;-
366
367  024602     124     123     102  T3TSBA: .ASCIZ  'TSBA Incorrect After TSDB Low Write'
368  024646     124     123     123  T3TSSR: .ASCIZ  'TSSR Incorrect After TSDB Low Write'
369  024712     116     157     040  T3SSR:  .ASCIZ  'No Sub-System Ready After TSDB Low Write'
370  024763     127     162     141  TST3ID: .ASCIZ  'Wrap Data - Low Byte'
371                                          .EVEN
372  025010                                  ENDTST
     025010                                                                          L10042:
     025010  104401                                                                    TRAP    C$ETST
373
374                                          .SBTTL  TEST  4: RAM TEST
375
376                                  ;        THIS TEST VERIFIES THAT ALL LOCATIONS OF THE RAM ON THE M7196
377                                  ;        CAN PROPERLY STORE AND READ BACK ALL DATA PATTERNS, AND THAT
378                                  ;        EACH RAM LOCATION IS UNIQUELY ADDRESSED (I.E., THAT ONE AND ONLY
379                                  ;        ONE LOCATION IS ACCESSED BY ANY PARTICULAR ADDRESS).  THESE
380                                  ;        TESTS ARE PERFORMED BY THREE SUBTESTS, DESCRIBED BELOW.  A
381                                  ;        BYPRODUCT OF THESE TESTS IS A VERIFICATION OF TWO REGISTERS IN
382                                  ;        THE 2901 AND THE CAPABLITY OF THE 2901 TO CORRECTLY PERFORM AN
383                                  ;        ADD.
384                                  ;
```

```
                                        ;
                                        ;
                                        ;       TEST 4 , SUBTEST 1:
                                        ;
                                        ;       THIS SUBTEST VERIFIES EACH RAM LOCATION BY FIRST PLACING THE
                                        ;       M7196 INTO MAINTENANCE MODE BY WRITING INTO THE LOW BYTE OF TSDB
                                        ;       AND THEN PERFORMING THE FOLLOWING SEQUENCE FOR EACH ADDRESS
                                        ;       0-7777 (OCTAL):
                                        ;
                                        ;
                                        ;            1.  THE ADDRESS TO BE TESTED IS LOADED INTO THE TSDB (VIA A
                                        ;                WORD WRITE).
                                        ;
                                        ;            2.  THE ADDRESSED RAM LOCATION IS WRITTEN, THEN READ INTO
                                        ;                THE LOW BYTE OF TSBA, BY WRITING A DATA BYTE INTO THE
                                        ;                LOW BYTE OF TSDB.
                                        ;
                                        ;            3.  THE LOW BYTE OF TSBA IS CHECKED TO SEE IF IT CONTAINS
                                        ;                THE DATA PATTERN ORIGINALLY WRITTEN;  A DISCREPANCY IS
                                        ;                REPORTED AS AN ERROR.
                                        ;
                                        ;            4.  THE ADDRESS OF THE LOCATION BEING TESTED IS AGAIN
                                        ;                WRITTEN INTO TSDB (WORD WRITE), TO CAUSE THE LOCATION
                                        ;                UNDER TEST TO AGAIN BE READ INTO THE LOW BYTE OF TSBA.
                                        ;                THE LOW BYTE OF TSBA IS AGAIN CHECKED AND DISCREPANCIES
                                        ;                REPORTED.
                                        ;
                                        ;            5.  THE HIGH BYTE OF TSBA IS CHECKED;  IT SHOULD CONTAIN
                                        ;                THE SUM OF THE HIGH AND LOW BYTES LAST WRITTEN INTO
                                        ;                TSDB AS A WORD.  A DISCREPANCY IS REPORTED AS A 2901
                                        ;                PROBLEM.
                                        ;
                                        ;            6.  THE CONTENT OF TSSR IS CHECKED;  SETTING OF THE SC BIT
                                        ;                IS IGNORED.  OTHER DISCREPANCIES IN TSSR ARE REPORTED.
                                        ;
                                        ;
                                        ;
025012                                          BGNTST
025012                                                                                          T4::

025012                                          BGNSUB                          ;///////////// BEGIN SUBTEST ////////////
025012                                                                                          T4.1:
025012  104402                                                                                          TRAP    C$BSUB

025014  012700  026322                          MOV     #TST4ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
025020  004737  016510                          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
025024  012737  000005  002214                  MOV     #5,LOOPCNT              ;PERFORM 5 ITERATIONS
025032                                  T4LOOP:
025032  004737  015774                          JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
025036  103405                                  BCS     20$                     ;BR IF INIT WAS OK
025040  010001                                  MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
025042                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
025042  104455                                                                                          TRAP    C$ERDF
025044  000621                                                                                          .WORD   401
025046  003650                                                                                          .WORD   SFIERR
025050  012034                                                                                          .WORD   SFIMSG
025052  005004                          20$:    CLR     R4                      ;SET RAM ADDRESS AT ZERO
025054  004737  016336                          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
```

```
442 025060  105065  000000                CLRB    TSDB(R5)                ;SET INTO MAINTENANCE MODE
443 025064                          25$:
444 025064                                BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
    025064  104404                                                                                  TRAP    C$BSEG
445
446 025066  110402                        MOVB    R4,R2                   ;EXPECTED DATA FROM WRAP-AROUND
447 025070  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
448 025074  010465  000000                MOV     R4,TSDB(R5)             ;LOAD ADDRESS INTO TSDB
449 025100  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
450 025104  110265  000000                MOVB    R2,TSDB(R5)             ;LOADS DATA INTO RAM LOCATION
451 025110  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
452 025114  016501  000000                MOV     TSBA(R5),R1             ;READS WRAP DATA
453 025120  120102                        CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ
454 025122  001404                        BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
458 025124                                ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
    025124  104456                                                                                  TRAP    C$ERHRD
    025126  000622                                                                                  .WORD   402
    025130  026160                                                                                  .WORD   TSBAM2
    025132  015474                                                                                  .WORD   EXPREC
459 025134                          30$:
460 025134                                ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<<
    025134                                                                                  10000$:
    025134  104405                                                                                  TRAP    C$ESEG
461
462 025136  005204                        INC     R4                      ;NEXT ADDRESS
463 025140  020427  010000                CMP     R4,#10000               ;END OF RAM MEMORY CHECK
464 025144  001347                        BNE     25$                     ;LOOP TILL ALL RAM WRITTEN
465 025146                                ESCAPE  SUB                     ;DON'T CONTINUE IF ERROR ON WRITE
    025146  104410                                                                                  TRAP    C$ESCAPE
    025150  000102                                                                                  .WORD   L10044-.
466
467 025152  005002                        CLR     R2                      ;CLEAR OUT R2 HIGH BITS
468 025154  005304                        DEC     R4                      ;SET BACK TO 7777
469 025156  110402                  40$:  MOVB    R4,R2                   ;GET DATA PATTERN BACK IN SHAPE
470 025160  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
471 025164  010465  000000                MOV     R4,TSDB(R5)             ;LOAD UP RAM ADDRESS POINTER
472 025170  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
473 025174  016501  000000                MOV     TSBA(R5),R1             ;READ RAM CONTENTS BACK
474 025200  120102                        CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
475 025202  001404                        BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
479 025204                                ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
    025204  104456                                                                                  TRAP    C$ERHRD
    025206  000623                                                                                  .WORD   403
    025210  026160                                                                                  .WORD   TSBAM2
    025212  015474                                                                                  .WORD   EXPREC
480 025214                          45$:  CKLOOP                          ;SCOPE LOOP
    025214  104406                                                                                  TRAP    C$CLP1
481 025216  116501  000001                MOVB    TSBAH(R5),R1            ;HIGH BYTE READ OF TSBA
482 025222  010402                        MOV     R4,R2                   ;DATA PATTERN WRITTEN
483 025224  000302                        SWAB    R2                      ;HIGH TO LOW
484 025226  060402                        ADD     R4,R2                   ;TOTAL OF BYTES IN LOW BYTE
485 025230  120102                        CMPB    R1,R2                   ;SUM OF BYTES WRITTEN TO TSDB = TSBAH
486 025232  001404                        BEQ     50$                     ;BR IF OK, THEY SHOULD BE
490 025234                                ERRHRD  ERRNO,M2901,EXPREC      ;2901 PROBLEM ADDER
    025234  104456                                                                                  TRAP    C$ERHRD
    025236  000624                                                                                  .WORD   404
    025240  026070                                                                                  .WORD   M2901
```

```
        025242  015474                                                                    .WORD   EXPREC
    491 025244                          50$:    CKLOOP                          ;SCOPE LOOP
        025244  104406                                                                    TRAP    C$CLP1
    492 025246  005304                          DEC     R4                      ;DROP DATA COUNTER (PATTERN)
    493 025250  002342                          BGE     40$                     ;NOT AT LOC. ZERO YET
    494
    495 025252                                  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        025252                                                                          L10044:
        025252  104403                                                                    TRAP    C$ESUB
    496
    497
    498 025254                                  BGNSUB                          ;///////////// BEGIN SUBTEST ///////////
        025254                                                                          T4.2:
        025254  104402                                                                    TRAP    C$BSUB
    499                                 ;       TEST 4, SUBTEST 2
    500
    501                                 ;
    502                                 ;       THIS SUBTEST WRITES RAM WITH ALL ZEROS
    503                                 ;       THEN WALKS AN ALL ONES WORD DOWN THROUGH MEMORY
    504                                 ;
    505 025256  004737  015774                  JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
    506 025262  103405                          BCS     20$                     ;BR IF INIT WAS OK
    510 025264  010001                          MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
    511 025266                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
        025266  104455                                                                    TRAP    C$ERDF
        025270  000625                                                                    .WORD   405
        025272  003650                                                                    .WORD   SFIERR
        025274  012034                                                                    .WORD   SFIMSG
    512 025276  005002                          20$:    CLR     R2                      ;TEST DATA = 0
    513 025300  005004                          CLR     R4                      ;STARTING RAM ADDRESS = 0
    514 025302  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    515 025306  105065  000000                  CLRB    TSDB(R5)                ;SET INTO MAINTENANCE MODE
    516 025312                          25$:
    517 025312                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
        025312  104404                                                                    TRAP    C$BSEG
    518
    519 025314  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    520 025320  010465  000000                  MOV     R4,TSDB(R5)             ;LOAD ADDRESS INTO TSDB
    521 025324  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    522 025330  110265  000000                  MOVB    R2,TSDB(R5)             ;LOADS DATA INTO RAM LOCATION
    523 025334  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    524 025340  016501  000000                  MOV     TSBA(R5),R1             ;READS WRAP DATA
    525 025344  120102                          CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ ?
    526 025346  001404                          BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
    530 025350                                  ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
        025350  104456                                                                    TRAP    C$ERHRD
        025352  000626                                                                    .WORD   406
        025354  026160                                                                    .WORD   TSBAM2
        025356  015474                                                                    .WORD   EXPREC
    531 025360                          30$:
    532 025360                                  ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<<
        025360                                                                          10000$:
        025360  104405                                                                    TRAP    C$ESEG
    533
    534 025362  005204                          INC     R4                      ;NEXT ADDRESS
    535 025364  020427  010000                  CMP     R4 #10000               ;END OF RAM MEMORY CHECK
    536 025370  001350                          BNE     25$                     ;BR, MORE RAM TO GO
```

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

```
585
586                                     ;       TEST 4, SUBTEST 3
587                                     ;
588                                     ;       THIS SUBTEST WRITES RAM WITH ALL ONES
589                                     ;       THEN WALKS A ZERO WORD DOWN THROUGH MEMORY
590                                     ;
591 025556  004737  015774                      JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
592 025562  103405                              BCS     20$                     ;BR IF INIT WAS OK
596 025564  010001                              MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
597 025566                                      ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
    025566  104455                                                                          TRAP    C$ERDF
    025570  000632                                                                          .WORD   410
    025572  003650                                                                          .WORD   SFIERR
    025574  012034                                                                          .WORD   SFIMSG
598 025576  012702  177777              20$:    MOV     #177777,R2              ;SET DATA AT ALL ONES
599 025602  005004                              CLR     R4                      ;SET RAM ADDRESS AT ZERO
600 025604  004737  016336                      JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
601 025610  105065  000000                      CLRB    TSDB(R5)                ;SET INTO MAINTENANCE MODE
602 025614                              25$:
603 025614                                      BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
    025614  104404                                                                          TRAP    C$BSEG
604
605 025616  004737  016336                      JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
606 025622  010465  000000                      MOV     R4,TSDB(R5)             ;LOAD ADDRESS INTO TSDB
607 025626  004737  016336                      JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
608 025632  110265  000000                      MOVB    R2,TSDB(R5)             ;LOADS DATA INTO RAM LOCATION
609 025636  004737  016336                      JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
610 025642  016501  000000                      MOV     TSBA(R5),R1             ;READS WRAP DATA
611 025646  120102                              CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ ?
612 025650  001404                              BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
616 025652                                      ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
    025652  104456                                                                          TRAP    C$ERHRD
    025654  000633                                                                          .WORD   411
    025656  026160                                                                          .WORD   TSBAM2
    025660  015474                                                                          .WORD   EXPREC
617 025662                              30$:
618 025662                                      ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
    025662                                                                                  10000$:
    025662  104405                                                                          TRAP    C$ESEG
619
620 025664  005204                              INC     R4                      ;NEXT ADDRESS
621 025666  020427  010000                      CMP     R4,#10000               ;END OF RAM MEMORY CHECK
622 025672  001350                              BNE     25$                     ;BR, MORE RAM TO GO
623
624 025674                                      ESCAPE  SUB                     ;NO CHECK IF WRITTEN INCORRECTLY
    025674  104410                                                                          TRAP    C$ESCAPE
    025676  000152                                                                          .WORD   L10046-.
625 025700  005304                      35$:    DEC     R4                      ;SET BACK TO 7777
626 025702  004737  016336              40$:    JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
627 025706  012702  000377                      MOV     #000377,R2              ;SET UP EXPECTED DATA REGISTER
628 025712  005001                              CLR     R1                      ;CLEAN OUT REGISTER
629 025714  010465  000000                      MOV     R4,TSDB(R5)             ;SELECT ADDRESS IN RAM
630 025720  004737  016336                      JSR     PC,CHKTSSR              ;WAIT FOR READY (SSR)
631 025724  016501  000000                      MOV     TSBA(R5),R1             ;PICK UP RAM CONTENTS
632 025730  120102                              CMPB    R1,R2                   ;IS MEMORY STILL ALL ONES
633 025732  001404                              BEQ     43$                     ;BR, IF OK (ALL ONES)
637 025734                                      ERRHRD  ERRNO,TSBAM3,EXPREC     ;MEMORY CHANGED AFTER ALL ONES WRITE
```

```
        025734  104456                                                                    TRAP    C$ERHRD
        025736  000634                                                                    .WORD   412
        025740  026242                                                                    .WORD   TSBAM3
        025742  015474                                                                    .WORD   EXPREC
    638 025744  005002                  43$:    CLR     R2                      ;SET UP NEW EXPECTED
    639 025746  010465  000000                  MOV     R4,TSDB(R5)             ;LOAD UP RAM ADDRESS POINTER
    640 025752  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    641 025756  110265  000000                  MOVB    R2,TSDB(R5)             ;WRITE DATA INTO RAM
    642 025762  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    643 025766  016501  000000                  MOV     TSBA(R5),R1             ;READ RAM CONTENTS BACK
    644 025772  120102                          CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
    645 025774  001404                          BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
    649 025776                                  ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
        025776  104456                                                                    TRAP    C$ERHRD
        026000  000635                                                                    .WORD   413
        026002  026160                                                                    .WORD   TSBAM2
        026004  015474                                                                    .WORD   EXPREC
    650 026006                          45$:    CKLOOP                          ;SCOPE LOOP
        026006  104406                                                                    TRAP    C$CLP1
    651 026010  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    652 026014  116501  000001                  MOVB    TSBAH(R5),R1            ;HIGH BYTE READ OF TSBA
    653 026020  010203                          MOV     R2,R3                   ;DATA PATTERN WRITTEN
    654 026022  000303                          SWAB    R3                      ;HIGH TO LOW
    655 026024  060203                          ADD     R2,R3                   ;TOTAL OF BYTES IN LOW BYTE
    656 026026  120103                          CMPB    R1,R3                   ;SUM OF BYTES WRITTEN TO TSDB = TSBAH
    657 026030  001404                          BEQ     50$                     ;BR IF OK, THEY SHOULD BE
    661 026032                                  ERRHRD  ERRNO,M2901,EXPREC      ;2901 PROBLEM ADDER
        026032  104456                                                                    TRAP    C$ERHRD
        026034  000636                                                                    .WORD   414
        026036  026070                                                                    .WORD   M2901
        026040  015474                                                                    .WORD   EXPREC
    662 026042                          50$:    CKLOOP                          ;SCOPE LOOP
        026042  104406                                                                    TRAP    C$CLP1
    663 026044  005304                          DEC     R4                      ;DROP RAM ADDRESS POINTER
    664 026046  001315                          BNE     40$                     ;NOT AT LOC. ZERO YET
    665
    666 026050                                  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        026050                                                                          L10046:
        026050  104403                                                                    TRAP    C$ESUB
    667
    668 026052  004737  016456                  JSR     PC,TSTLOOP              ;DO WE NEED TO ITERATE TEST ?
    669 026056  103002                          BCC     63$                     ;BRANCH IF NOT
    670 026060  000137  025032                  JMP     T4LOOP                  ;EXECUTE AGAIN
    671 026064                          63$:    EXIT    TST                     ;ALL DONE THIS TEST
        026064  104432                                                                    TRAP    C$EXIT
        026066  000256                                                                    .WORD   L10043-.
    672                                 ;+
    673                                 ;LOCAL TEXT MESSAGES FOR TEST
    674                                 ;-
    675
    676 026070      040     124     123 M2901:  .ASCIZ  ' TSBA High Byte Not Sum of Last TSDB Write (2901 Error)'
    677 026160      040     127     162 TSBAM2: .ASCIZ  ' Write to TSDB Not Equal to Read of TSBA Low Byte'
    678 026242      127     162     151 TSBAM3: .ASCIZ  'Write To RAM Location Modified Another Location'
    679 026322      122     101     115 TST4ID: .ASCIZ  'RAM Verification'
    680                                         .EVEN
    681 026344                                  ENDTST
        026344                                                                          L10043:
```

```
      026344  104401                                                                    TRAP    C$ETST
682
683                                                   .SBTTL  TEST  5: SECOND INITIALIZATION TEST
684
685                                         ;         THIS TEST VERIFIES THE SAME ELEMENTS AS DID INITIALIZATION TEST
686                                         ;         #1 AND ALSO CHECKS THAT CERTAIN PARTS OF RAM IS CLEARED TO ZERO
687                                         ;         AND THAT 2901 REGISTERS 10 AND 11 ARE ALSO CLEARED TO ZERO.
688                                         ;         THIS IS A CONFIDENCE CHECK OF A PART OF THE SELF-TEST SEQUENCE
689                                         ;         (I.E., THAT IT IS REALLY BEING EXECUTED).  FOR EACH OF TWO
690                                         ;         SUBTESTS (ONE FOR INITIALIZING VIA A BUS INIT, THE OTHER FOR
691                                         ;         INITIALIZING BY WRITING INTO THE TSSR), THE FOLLOWING SEQUENCE
692                                         ;         IS PERFORMED:
693                                         ;
694                                         ;
695                                         ;              1.  EACH RAM LOCATION AND 2901 REGISTERS 10 AND 11 ARE SET
696                                         ;                  TO ALL 1'S BY USING WRITES INTO THE TSDB REGISTER (LOW
697                                         ;                  BYTE AND MAINTENANCE MODE WORD WRITES).
698                                         ;
699                                         ;              2.  THE CONTROLLER IS INITIALIZED AND THE VARIOUS CHECKS ON
700                                         ;                  THE TSSR DESCRIBED IN INITIALIZATION TEST #1 ARE
701                                         ;                  PERFORMED.
702                                         ;
703                                         ;              3.  #1'S (377 OCTAL) ARE WRITTEN INTO THE LOW BYTE OF TSDB,
704                                         ;                  WHICH SHOULD CAUSE RAM LOCATION 0 TO BE WRITTEN TO ALL
705                                         ;                  1'S SINCE 2901 REGISTERS 10 AND 11, SPECIFYING THE RAM
706                                         ;                  ADDRESS, SHOULD BE 0.  RAM LOCATION 0 IS VERIFIED BY
707                                         ;                  WRITING A WORD OF ZEROS INTO THE TSDB.  THE RESULTING
708                                         ;                  LOW BYTE OF TSBA SHOULD CONTAIN ALL 1'S.
709                                         ;
710                                         ;              4.  THE ENTIRE RAM IS SCANNED.  LOCATION 0 SHOULD CONTAIN
711                                         ;                  ALL 1'S AND THE REMAINING LOCATIONS, EXCEPT FOR THE
712                                         ;                  MESSAGE BUFFER IMAGE AREA, SHOULD CONTAIN 0.
713                                         ;                  DISCREPANCIES ARE REPORTED.  AN ERROR AT THIS POINT IS
714                                         ;                  MOST LIKELY DUE TO A ROM, PIPELINE OR SEQUENCER PROBLEM
715                                         ;                  OR A TIMING PROBLEM.
716                                         ;
717                                         ;
718                                         ;
719                                         ;
720 026346                                            BGNTST
    026346                                                                                T5::
725 026346  012700  027320                            MOV     #TST5ID,R0                ;ASCII MESSAGE TO IDENTIFY TEST
726 026352  004737  016510                            JSR     PC,TSTSETUP               ;DO INITIAL TEST SETUP
727 026356  012737  000024  002214                    MOV     #20.,LOOPCNT              ;PERFORM 20 ITERATIONS
728 026364                                  T5LOOP:
729 026364  005037  002220                            CLR     FATFLG                    ;CLEAR THE FATAL ERROR FLAG
730
731 026370                                            BGNSUB                            ;///////////// BEGIN SUBTEST ////////////
    026370                                                                                T5.1:
    026370  104402                                                                                TRAP    C$BSUB
732
733 026372  004737  015774                            JSR     PC,SOFINIT                ;DO A SOFT TO START
734 026376  103404                                    BCS     10$                       ;BRANCH IF O.K.
738 026400                                            ERRDF   ERRNO,SFIERR,SFIMSG       ;REPORT ERROR AND DROP DRIVE
    026400  104455                                                                                TRAP    C$ERDF
    026402  000765                                                                                .WORD   501
    026404  003650                                                                                .WORD   SFIERR
```

```
     026406  012034                                                                  .WORD   SFIMSG
739  026410  012702  177777          10$:    MOV     #1,R2                   ;ALL ONE DATA PATTERN
740  026414  005004                          CLR     R4                      ;STARTING RAM ADDRESS
741  026416  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
742  026422  105065  000000          15$:    CLRB    TSDB(R5)                ;SET MAINTENANCE MODE
743  026426  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
744  026432  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE NEXT RAM ADDRESS
745  026436  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
746  026442  110265  000000                  MOVB    R2,TSDB(R5)             ;LOAD TEST DATA
747  026446  005204                          INC     R4                      ;NEXT ADDRESS TO TEST
748  026450  020427  007777                  CMP     R4,#7777                ;COMPARE TO LAST ADDRESS
749  026454  003762                          BLE     15$                     ;BRANCH TILL ALL DATA WRITTEN
750  026456                                  BRESET                          ;ISSUE A BUS RESET
     026456  104433                                                                  TRAP    C$RESET
751  026460  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
752  026464  016501  000002                  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
753  026470  010102                          MOV     R1,R2                   ;CONTENTS OF TSSR
754  026472  042702  176277                  BIC     #^C<HIADDR!OFL>,R2      ;THESE BITS MAY BE SET
755  026476  052702  002200                  BIS     #SSR!NBA,R2             ;READY AND NEW DATA SHOULD BE SET
756  026502  020102                          CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
757  026504  001406                          BEQ     20$                     ;BRANCH IF COMPARE
761  026506                                  ERRDF   ERRNO,SFHERR,SFFMSG     ;REPORT A FATAL ERROR
     026506  104455                                                                  TRAP    C$ERDF
     026510  000766                                                                  .WORD   502
     026512  003703                                                                  .WORD   SFHERR
     026514  012102                                                                  .WORD   SFFMSG
762  026516  005237  002220                  INC     FATFLG                  ;SET FATAL ERROR FLAG
763  026522                          20$:    CKLOOP                          ;LOOP ON ERROR IF FLAG SET
     026522  104406                                                                  TRAP    C$CLP1
764  026524                                  ESCAPE  SUB                     ;EXIT IF FATAL ERROR DETECTED
     026524  104410                                                                  TRAP    C$ESCAPE
     026526  000170                                                                  .WORD   L10050-.
765  026530  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
766  026534  105065  000000                  CLRB    TSDB(R5)                ;PUT BACK INTO MAINTENANCE MODE
767  026540  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
768  026544  005065  000000                  CLR     TSDB(R5)                ;SET ADDRESS BACK TO 0000
769  026550  012702  000377                  MOV     #377,R2                 ;
770  026554  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
771  026560  110265  000000                  MOVB    R2,TSDB(R5)             ;SHOULD POINT TO RAM 0
772  026564  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
773  026570  005065  000000                  CLR     TSDB(R5)                ;SELECT LOCATION 0
774  026574  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
775  026600  116501  000000                  MOVB    TSBA(R5),R1             ;READ RAM LOCATION SPECIFIED
776  026604  120102                          CMPB    R1,R2                   ;LOCATION SHOULD BE 377 OCTAL
777  026606  001406                          BEQ     25$                     ;BR IF OK
778  026610                                  ERRDF   ERRNO,TSADDR,EXPREC     ;WASN'T POINTING TO CORRECT LOC.
     026610  104455                                                                  TRAP    C$ERDF
     026612  000766                                                                  .WORD   502
     026614  027406                                                                  .WORD   TSADDR
     026616  015474                                                                  .WORD   EXPREC
779  026620  005237  002220                  INC     FATFLG                  ;SET THE FATAL ERROR FLAG
780  026624                          25$:    CKLOOP                          ;SCOPE LOOP
     026624  104406                                                                  TRAP    C$CLP1
781  026626                                  ESCAPE  SUB                     ;NO MORE CHECKS IF FATAL ERROR
     026626  104410                                                                  TRAP    C$ESCAPE
     026630  000066                                                                  .WORD   L10050-.
782  026632  012704  000310                  MOV     #310,R4                 ;START WITH LOC 310
```

```
026636  005002                          CLR     R2                      ;MEMORY EXPECTED SHOULD BE 000000
026640  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
026644  010465  000000          30$:    MOV     R4,TSDB(R5)             ;SELECT LOCATION SPECIFIED
026650  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
026654  116501  000000                  MOVB    TSBA(R5),R1             ;READ LOC CONTENTS
026660  120102                          CMPB    R1,R2                   ;CHECK MEMORY FOR 000000
026662  001406                          BEQ     40$                     ;BRANCH IF DATA OKAY
026664                                  ERROF   ERRNO,TSMEM,SFFMSG      ;MEMORY NOT ZERO AFTER INIT.
026664  104455                                                                          TRAP    C$ERDF
026666  000766                                                                          .WORD   502
026670  027350                                                                          .WORD   TSMEM
026672  012102                                                                          .WORD   SFFMSG
026674  005237  002220                  INC     FATFLG                  ;SET THE FATAL ERROR FLAG
026700                          40$:    CKLOOP
026700  104406                                                                          TRAP    C$CLP1
026702                                  ESCAPE  SUB                     ;EXIT ON FATAL ERROR
026702  104410                                                                          TRAP    C$ESCAPE
026704  000012                                                                          .WORD   L10050-.
026706  005204                          INC     R4                      ;LOOK AT NEXT RAM LOC.
026710  020427  000400                  CMP     R4,#400                 ;AT TOP OF RAM ADDRESS SPACE
026714  001353                          BNE     30$                     ;BRANCH TILL ALL MEMORY TESTED

026716                                  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
026716                                                                                  L10050:
026716  104403                                                                          TRAP    C$ESUB

026720  005737  002220                  TST     FATFLG                  ;IS FATAL ERROR FLAG SET ?
026724  001404                          BEQ     50$                     ;BRANCH IF NOT
026726  004737  017202                  JSR     PC,CKDROP               ;NO LOOP, TRY TO DROP DEVICE
026732  005037  002220                  CLR     FATFLG                  ;CLEAR THE FATAL ERROR FLAG
026736                          50$:

026736                                  BGNSUB                          ;///////////// BEGIN SUBTEST ///////////
026736                                                                                  T5.2:
026736  104402                                                                          TRAP    C$BSUB

026740  004737  015774                  JSR     PC,SOFINIT              ;DO A SOFT TO START
026744  103404                          BCS     10$                     ;BRANCH IF O.K.
026746                                  ERROF   ERRNO,SFIERR,SFIMSG     ;REPORT ERROR AND DROP DRIVE
026746  104455                                                                          TRAP    C$ERDF
026750  000767                                                                          .WORD   503
026752  003650                                                                          .WORD   SFIERR
026754  012034                                                                          .WORD   SFIMSG
026756  012702  177777          10$:    MOV     #-1,R2                  ;ALL ONE DATA PATTERN
026762  005004                          CLR     R4                      ;STARTING RAM ADDRESS
026764  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
026770  105065  000000          15$:    CLRB    TSDB(R5)                ;SET MAINTENANCE MODE
026774  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
027000  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE NEXT RAM ADDRESS
027004  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
027010  110265  000000                  MOVB    R2,TSDB(R5)             ;LOAD TEST DATA
027014  005204                          INC     R4                      ;NEXT ADDRESS TO TEST
027016  020427  007777                  CMP     R4,#7777                ;COMPARE TO LAST ADDRESS
027022  003762                          BLE     15$                     ;BRANCH TILL ALL DATA WRITTEN
027024  005065  000002                  CLR     TSSR(R5)                ;ISSUE A SOFT RESET
027030  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
027034  016501  000002                  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
```

```
828 027040  010102                      MOV     R1,R2                   ;CONTENTS OF TSSR
829 027042  042702  176277              BIC     #^C<HIADDR!OFL>,R2      ;THESE BITS MAY BE SET
830 027046  052702  002200              BIS     #SSR!NBA,R2             ;READY AND NEW DATA SHOULD BE SET
831 027052  020102                      CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
832 027054  001406                      BEQ     20$                     ;BRANCH IF COMPARE
836 027056                              ERRDF   ERRNO,SFHERR,SFFMSG     ;REPORT A FATAL ERROR
    027056  104455                                                                          TRAP    C$ERDF
    027060  000770                                                                          .WORD   504
    027062  003703                                                                          .WORD   SFHERR
    027064  012102                                                                          .WORD   SFFMSG
837 027066  005237  002220              INC     FATFLG                  ;SET FATAL ERROR FLAG
838 027072                      20$:    CKLOOP                          ;LOOP ON ERROR IF FLAG SET
    027072  104406                                                                          TRAP    C$CLP1
839 027074                              ESCAPE  SUB                     ;EXIT IF FATAL ERROR DETECTED
    027074  104410                                                                          TRAP    C$ESCAPE
    027076  000170                                                                          .WORD   L10051-.
840 027100  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
841 027104  105065  000000              CLRB    TSDB(R5)                ;PUT BACK INTO MAINTENANCE MODE
842 027110  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
843 027114  005065  000000              CLR     TSDB(R5)                ;SET ADDRESS BACK TO 0000
844 027120  012702  000377              MOV     #377,R2                 ;
845 027124  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
846 027130  110265  000000              MOVB    R2,TSDB(R5)             ;SHOULD POINT TO RAM 0
847 027134  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
848 027140  005065  000000              CLR     TSDB(R5)                ;SELECT LOCATION 0
849 027144  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
850 027150  116501  000000              MOVB    TSBA(R5),R1             ;READ RAM LOCATION SPECIFIED
851 027154  120102                      CMPB    R1,R2                   ;LOCATION SHOULD BE 377 OCTAL
852 027156  001406                      BEQ     25$                     ;BR IF OK
853 027160                              ERRDF   ERRNO,T5ADDR,EXPREC     ;WASN'T POINTING TO CORRECT LOC.
    027160  104455                                                                          TRAP    C$ERDF
    027162  000770                                                                          .WORD   504
    027164  027406                                                                          .WORD   T5ADDR
    027166  015474                                                                          .WORD   EXPREC
854 027170  005237  002220              INC     FATFLG                  ;SET THE FATAL ERROR FLAG
855 027174                      25$:    CKLOOP                          ;SCOPE LOOP
    027174  104406                                                                          TRAP    C$CLP1
856 027176                              ESCAPE  SUB                     ;NO MORE CHECKS IF FATAL ERROR
    027176  104410                                                                          TRAP    C$ESCAPE
    027200  000066                                                                          .WORD   L10051 .
857 027202  012704  000310              MOV     #310,R4                 ;START WITH LOC 310
858 027206  005002                      CLR     R2                      ;MEMORY EXPECTED SHOULD BE 000000
859 027210  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
860 027214  010465  000000      30$:    MOV     R4,TSDB(R5)             ;SELECT LOCATION SPECIFIED
861 027220  004737  016336              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
862 027224  116501  000000              MOVB    TSBA(R5),R1             ;READ LOC CONTENTS
863 027230  120102                      CMPB    R1,R2                   ;CHECK MEMORY FOR 000000
864 027232  001406                      BEQ     40$                     ;BRANCH IF DATA OKAY
865 027234                              ERRDF   ERRNO,T5MEM,SFFMSG      ;MEMORY NOT ZERO AFTER INIT.
    027234  104455                                                                          TRAP    C$ERDF
    027236  000770                                                                          .WORD   504
    027240  027350                                                                          .WORD   T5MEM
    027242  012102                                                                          .WORD   SFFMSG
866 027244  005237  002220              INC     FATFLG                  ;SET THE FATAL ERROR FLAG
867 027250                      40$:    CKLOOP
    027250  104406                                                                          TRAP    C$CLP1
868 027252                              ESCAPE  SUB                     ;EXIT ON FATAL ERROR
```

```
        027252  104410                                                                    TRAP    C$ESCAPE
        027254  000012                                                                    .WORD   L10051 .
    869 027256  005204                          INC     R4                      ;LOOK AT NEXT RAM LOC.
    870 027260  020427  000400                  CMP     R4,#400                 ;AT TOP OF RAM ADDRESS SPACE
    871 027264  001353                          BNE     30$                     ;BRANCH TILL ALL MEMORY TESTED
    872
    873 027266                                  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        027266                                                                          L10051:
        027266  104403                                                                    TRAP    C$ESUB
    874
    875 027270  005737  002220                  TST     FATFLG                  ;IS FATAL ERROR FLAG SET ?
    876 027274  001402                          BEQ     50$                     ;BRANCH IF NOT
    877 027276  004737  017202                  JSR     PC,CKDROP               ;NO LOOP, TRY TO DROP DEVICE
    878 027302  004737  016456          50$:    JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
    879 027306  103002                          BCC     60$                     ;BRANCH IF NOT
    880 027310  000137  026364                  JMP     T5LOOP                  ;LOOP UNTIL COUNT EXPIRED
    881 027314                          60$:    EXIT    TST                     ;ALL DONE THIS TEST
        027314  104432                                                                    TRAP    C$EXIT
        027316  000132                                                                    .WORD   L10047-.
    882
    883                                 ;+
    884                                 ;LOCAL TEXT MESSAGES FOR TEST
    885                                 ;-
    886
    887 027320      105     170     164 TST5ID: .ASCIZ  'Extended Initialization'
    888 027350      111     156     143 T5MEM:  .ASCIZ  'Incorrect RAM Data After Init'
    889 027406      111     156     143 T5ADDR: .ASCIZ  'Incorrect RAM Address After Init'
    890                                         .EVEN
    891 027450                                  ENDTST
        027450                                                                          L10047:
        027450  104401                                                                    TRAP    C$ETST
    892
    893
    894                                 .SBTTL  TEST  6: COMMAND REJECT
    895
    896
    897                         ;       THIS TEST VERIFIES THAT ALL COMMANDS OTHER THAN WRITE
    898                         ;       CHARACTERISTICS ARE REJECTED DUE TO THE NEED BUFFER ADDRESS
    899                         ;       (NBA) BIT BEING SET IN TSSR, AND THAT THE TSBA AND TSSR
    900                         ;       REGISTERS ARE LEFT IN THE PROPER STATE AFTER EACH COMMAND IS
    901                         ;       REJECTED.  THIS TEST CHECKS MICROPROCESSOR SEQUENCING, BASIC
    902                         ;       COMMAND DECODING AND DATI DMA HANDLING.  THIS TEST CONTAINS TWO
    903                         ;       SUBTESTS:  SUBTEST 1 SEQUENCES THROUGH ALL COMMAND WORDS (OTHER
    904                         ;       THAN WRITE CHARACTERISTICS) WITH THE INTERRUPT ENABLE (IE) BIT
    905                         ;       CLEAR AND VERIFIES THAT AN INTERRUPT IS NOT GENERATED BY THE
    906                         ;       REJECTED COMMAND;  SUBTEST 2 PERFORMS SIMILARLY TO SUBTEST 1 BUT
    907                         ;       SETS THE IE BIT IN EACH COMMAND WORD AND VERIFIES THAT AN
    908                         ;       INTERRUPT IS GENERATED WHEN THE COMMAND IS REJECTED.  SUBTEST 1
    909                         ;       SETS UP THE INTERRUPT SERVICE ROUTINE TO FLAG UNEXPECTED
    910                         ;       INTERRUPTS.  THE COMMAND WORD IN THE COMMAND BUFFER IS
    911                         ;       INITIALIZED TO 100000 (OCTAL) AND THE REMAINING THREE WORDS IN
    912                         ;       THE COMMAND BUFFER ARE SET TO KNOWN UNIQUE PATTERNS.  THEN THE
    913                         ;       FOLLOWING SEQUENCE IS PERFORMED:
    914                         ;
    915                         ;
    916                         ;           1.  INITIALIZE THE CONTROLLER BY WRITING INTO THE TSSR;
    917                         ;               PROPER INITIAL CONDITIONS ARE VERIFIED.
```

```
                                        ;
                                        ;          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.
                                        ;
027452                                          BGNTST
027452                                                                                  T6::
027452  012700  030703                          MOV     #TST6ID,R0                      ;ASCII MESSAGE TO IDENTIFY TEST
027456  004737  016510                          JSR     PC,TSTSETUP                     ;DO INITIAL TEST SETUP
027462  012737  000024  002214                  MOV     #20.,LOOPCNT                    ;PERFORM 20 ITERATIONS
027470                                  T6LOOP:
027470                                          BGNSUB                                  ;///////////// BEGIN SUBTEST ///////////
027470                                                                                  T6.1:
027470  104402                                                                                  TRAP    C$BSUB
027472                                          SETPRI  #PRI00                          ;LOWER PRIORITY TO ALLOW INTERRUPTS
027472  012700  000000                                                                          MOV     #PRI00,R0
027476  104441                                                                                  TRAP    C$SPRI
027500  012704  030360                          MOV     #T6PACKET,R4                    ;GET THE ADDRESS OF COMMAND PACKET
027504  012703  002750                          MOV     #TSTBLK,R3                      ;BLOCK OF TEST DATA
027510  012314                          5$:     MOV     (R3)+,(R4)                      ;INSERT THE NEXT TEST DATA WORD
027512                                          BGNSEG                                  ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
027512  104404                                                                                  TRAP    C$BSEG
027514  004737  015774                          JSR     PC,SOFINIT                      ;DO SOFT INIT OF CONTROLLER
027520  103405                                  BCS     10$                             ;BR IF SOFT INIT = OK
027522  010001                                  MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
```

```
  976 027524                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      027524  104455                                                                            TRAP    C$ERDF
      027526  001131                                                                            .WORD   601
      027530  003650                                                                            .WORD   SFIERR
      027532  012034                                                                            .WORD   SFIMSG
  977 027534  005037  002220          10$:  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
  978 027540  005037  002222                INTRECV                         ;CLEAR INTERRUPT RECEIVED FLAG
  979 027544  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
  980 027550  042714  000200                BIC     #BIT7,(R4)              ;DISABLE INTERRUPTS
  981 027554  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
  982 027560  004737  016250                JSR     PC,WAITF                ;WAIT FOR SSR TO SET
  983 027564  103407                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
  984 027566  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  988 027570                                ERRDF   ERRNO,T6SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      027570  104455                                                                            TRAP    C$ERDF
      027572  001132                                                                            .WORD   602
      027574  030415                                                                            .WORD   T6SSR
      027576  012046                                                                            .WORD   PKTSSR
  989 027600  005237  002220                INC     FATFLG                  ;SET FATAL ERROR FLAG
  990 027604                          15$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      027604  104406                                                                            TRAP    C$CLP1
  991 027606                                ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
      027606  104410                                                                            TRAP    C$ESCAPE
      027610  000170                                                                            .WORD   L10053 .
  992 027612  005737  002222                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
  993 027616  001404                        BEQ     22$                     ;BRANCH IF NOT
  997 027620                                ERRHRD  ERRNO,T6INT,PKTSSR
      027620  104456                                                                            TRAP    C$ERHRD
      027622  001133                                                                            .WORD   603
      027624  030473                                                                            .WORD   T6INT
      027626  012046                                                                            .WORD   PKTSSR
  998 027630  012702  102206          22$:  MOV     #SC!NBA!SSR!TSREJ,R2    ;EXPECTED CONTENTS OF TSSR
  999 027634  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
 1000 027640  016501  000002                MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 1001 027644  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 1002 027650  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 1003 027652  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 1004 027656  020201                 25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 1005 027660  001404                        BEQ     30$                     ;OKAY IF MATCH
 1009 027662                                ERRHRD  ERRNO,T6NBA,PKTSSR      ;NBA NOT SET TO REJECT
      027662  104456                                                                            TRAP    C$ERHRD
      027664  001134                                                                            .WORD   604
      027666  030370                                                                            .WORD   T6NBA
      027670  012046                                                                            .WORD   PKTSSR
 1010 027672                          30$:  CKLOOP                          ;LOOP ON ERROR ?
      027672  104406                                                                            TRAP    C$CLP1
 1011 027674  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
 1012 027700  016501  000000                MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
 1013 027704  010402                        MOV     R4,R2                   ;START OF THE PACKET
 1014 027706  062702  000010                ADD     #10,R2                  ;EXPECT TSDA TO PACKET + 10
 1015 027712  020102                        CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
 1016 027714  001404                        BEQ     35$                     ;ERROR IF NOT EQUAL
 1020 027716                                ERRHRD  ERRNO,T6TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
      027716  104456                                                                            TRAP    C$ERHRD
      027720  001135                                                                            .WORD   605
      027722  030631                                                                            .WORD   T6TSBA
      027724  015474                                                                            .WORD   EXPREC
```

```
027726  004737  011114          35$:    JSR     PC,CKRAM                ;SEE IF DATA IN RAM IS CORRECT
027732  103404                          BCS     40$                     ;BRANCH IF PACKET IN RAM IS CORRECT
027734                                  ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
027734  104456                                                                          TRAP    C$ERHRD
027736  001136                                                                          .WORD   606
027740  004743                                                                          .WORD   PKTRAM
027742  015510                                                                          .WORD   RAMERR
027744                          40$:    ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
027744                                                                                  10000$:
027744  104405                                                                          TRAP    C$ESEG
027746  011300                          MOV     (R3),R0                 ;PACKET COMMAND WORD
027750  042700  177740                  BIC     #177740,R0              ;GET BITS 0-4
027754  020027  000004                  CMP     R0,#4                   ;DON'T TEST WRITE CHARACTERISTICS
027760  001002                          BNE     45$                     ;BRANCH IF OK
027762  062703  000002                  ADD     #2,R3                   ;GET NEXT WORD FROM DATA TABLE
027766  020327  003060          45$:    CMP     R3,#TBLEND              ;REACHED END OF TABLE ?
027772  103002                          BHIS    50$                     ;BRANCH IF END OF TABLE
027774  000137  027510                  JMP     5$                      ;CONTINUE TEST WITH NEW DATA

030000                          50$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
030000                                                                                  L10053:
030000  104403                                                                          TRAP    C$ESUB

030002  005737  002220                  TST     FATFLG                  ;ANY FATAL ERRORS ?
030006  001402                          BEQ     60$                     ;BRANCH IF NOT
030010  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT

030014                          60$:    BGNSUB                          ;///////////// BEGIN SUBTEST /////////////
030014                                                                                  T6.2:
030014  104402                                                                          TRAP    C$BSUB

030016                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
030016  012700  000000                                                                  MOV     #PRI00,R0
030022  104441                                                                          TRAP    C$SPRI
030024  012704  030360                  MOV     #T6PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
030030  012703  002750                  MOV     #TSTBLK,R3              ;START OF TEST DATA
030034  012314                  5$:     MOV     (R3)+,(R4)              ;PLACE NEXT DATA WORD IN PACKET
030036                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
030036  104404                                                                          TRAP    C$BSEG
030040  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
030044  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
030046  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
030050                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
030050  104455                                                                          TRAP    C$ERDF
030052  001137                                                                          .WORD   607
030054  003650                                                                          .WORD   SFIERR
030056  012034                                                                          .WORD   SFIMSG
030060  005037  002220          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
030064  005037  002222                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
030070  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
030074  052714  000200                  BIS     #BIT7,(R4)              ;ENABLE INTERRUPTS
030100  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
030104  004737  016250                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
030110  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
030112  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

```
1070 030114                                ERRDF   ERRNO,T6SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     030114  104455                                                                          TRAP    C$ERDF
     030116  001140                                                                          .WORD   608
     030120  030415                                                                          .WORD   T6SSR
     030122  012046                                                                          .WORD   PKTSSR
1071 030124  005237  002220                INC     FATFLG                  ;SET FATAL ERROR FLAG
1072 030130                          15$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     030130  104406                                                                          TRAP    C$CLP1
1073 030132                                ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
     030132  104410                                                                          TRAP    C$ESCAPE
     030134  000170                                                                          .WORD   L10054-.
1074 030136  005737  002222                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1075 030142  001004                        BNE     22$                     ;BRANCH IF YES
1079 030144                                ERRHRD  ERRNO,T6NINT,PKTSSR     ;REPORT ERROR IF NO INTERRUPT
     030144  104456                                                                          TRAP    C$ERHRD
     030146  001141                                                                          .WORD   609
     030150  030551                                                                          .WORD   T6NINT
     030152  012046                                                                          .WORD   PKTSSR
1080 030154  012702  102206          22$:  MOV     #SC!NBA!SSR!TSREJ,R2    ;EXPECTED CONTENTS OF TSSR
1081 030160  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1082 030164  016501  000002                MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1083 030170  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1084 030174  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1085 030176  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1086 030202  020201                  25$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1087 030204  001404                        BEQ     30$                     ;OKAY IF MATCH
1091 030206                                ERRHRD  ERRNO,T6NBA,PKTSSR      ;NBA NOT SET TO REJECT
     030206  104456                                                                          TRAP    C$ERHRD
     030210  001142                                                                          .WORD   610
     030212  030370                                                                          .WORD   T6NBA
     030214  012046                                                                          .WORD   PKTSSR
1092 030216                          30$:  CKLOOP                          ;LOOP ON ERROR ?
     030216  104406                                                                          TRAP    C$CLP1
1093 030220  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1094 030224  016501  000000                MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
1095 030230  010402                        MOV     R4,R2                   ;START OF THE PACKET
1096 030232  062702  000010                ADD     #10,R2                  ;EXPECT TSDA TO PACKET + 10
1097 030236  020102                        CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1098 030240  001404                        BEQ     35$                     ;ERROR IF NOT EQUAL
1102 030242                                ERRHRD  ERRNO,T6TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     030242  104456                                                                          TRAP    C$ERHRD
     030244  001143                                                                          .WORD   611
     030246  030631                                                                          .WORD   T6TSBA
     030250  015474                                                                          .WORD   EXPREC
1103
1104
1105 030252  004737  011114          35$:  JSR     PC,CKRAM                ;SEE IF DATA IN RAM IS CORRECT
1106 030256  103404                        BCS     40$                     ;BRANCH IF PACKET IN RAM IS CORRECT
1110 030260                                ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
     030260  104456                                                                          TRAP    C$ERHRD
     030262  001144                                                                          .WORD   612
     030264  004743                                                                          .WORD   PKTRAM
     030266  015510                                                                          .WORD   RAMERR
1111 030270                          40$:  ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     030270                                                                        10000$:
     030270  104405                                                                          TRAP    C$ESEG
1112 030272  011300                        MOV     (R3),R0                 ;NEXT PACKET COMMAND WORD
```

```
030274  042700  177740                  BIC     #177740,R0              ;GET BITS 0-4
030300  020027  000004                  CMP     R0,#4                   ;DON'T TEST WRITE CHARACTERISTICS
030304  001002                          BNE     45$                     ;BRANCH IF NOT WRITE CHARACTERISTICS
030306  062703  000002                  ADD     #2,R3                   ;BY-PASS WRITE CHARACTERISTCS
030312  020327  003060          45$:    CMP     R3,#TBLEND              ;HAVE WE COMPLETED DATA TABLE ?
030316  103002                          BHIS    50$                     ;BRANCH IF ALL TESTED
030320  000137  030034                  JMP     5$                      ;TEST WITH NEXT DATA

030324                          50$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
030324                                                                                  L10054:
030324  104403                                                                                  TRAP    C$ESUB
030326  005737  002220                  TST     FATFLG                  ;ANY FATAL ERRORS ?
030332  001402                          BEQ     60$                     ;BRANCH IF NOT
030334  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
030340  004737  016456          60$:    JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
030344  103002                          BCC     62$                     ;BRANCH IF NOT
030346  000137  027470                  JMP     T6LOOP                  ;LOOP UNTIL COUNT EXPIRED
030352                          62$:    EXIT    TST                     ;ALL DONE THIS TEST
030352  104432                                                                                  TRAP    C$EXIT
030354  000346                                                                                  .WORD   L10052-.

                                ;+
                                ;LOCAL STORAGE FOR THIS TEST
                                ;-

030356                                  .BLKB   10-<.-TSV2&7>
030360                          16PACKET:                               ;COMMAND PACKET FOR TEST
030360  000000                          .WORD   0                       ;WILL CONTAIN VARIABLE COMMANDS
030362  052525                          .WORD   052525
030364  125252                          .WORD   125252
030366  052525                          .WORD   052525


                                ;+
                                ;LOCAL TEXT MESSAGES FOR TEST
                                ;-

030370     103     157     155  T6NBA:  .ASCIZ  'Command Not Rejected'
030415     103     157     156  T6SSR:  .ASCIZ  'Contents of TSSR Incorrect After Write Packet'
030473     125     156     145  T6INT:  .ASCIZ  'Unexpected Interrupt Received On Write Packet'
030551     105     170     160  T6NINT: .ASCIZ  'Expected Interrupt Not Received On Write Packet'
030631     111     156     143  T6TSBA: .ASCIZ  'Incorrect TSBA Address After Packet Write'
030703     103     157     155  TST6ID: .ASCIZ  'Command Reject'
                                        .EVEN
030722                                  ENDTST
030722                                                                          L10052:
030722  104401                                                                                  TRAP    C$ETST

                                        .SBTTL  TEST  7: WRITE CHARACTERISTICS


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

```
                                        ;       DATA BITS OPERATE PROPERLY;  THE FUNCTIONING OF THESE BITS IS
                                        ;       VERIFIED IN SUBSEQUENT TESTS.  ALL COMMANDS EXECUTED IN THIS
                                        ;       TEST HAVE THE INTERRUPT ENABLE (IE) BIT CLEARED TO ZERO, SO NO
                                        ;       INTERRUPTS SHOULD BE GENERATED.  HOWEVER, THE PROGRAM RUNS AT
                                        ;       PROCESSOR PRIORITY 0, WITH THE INTERRUPT SERVICE ROUTINE SET UP
                                        ;       TO FLAG UNEXPECTED INTERRUPTS.  IF AN INTERRUPT OCCURS, A
                                        ;       PROBLEM EXISTS IN EITHER THE LSI-11 BUS INTERFACE SECTION OR IN
                                        ;       THE ROM OR PIPELINE.
                                        ;
                                        ;       THIS TEST CHECKS VARIOUS MICROPROGRAM SEQUENCES, COMMAND
                                        ;       DECODING, DMA LOGIC, AND BASIC PACKET PROTOCOL HANDLING.  THIS
                                        ;       IS THE FIRST TEST IN WHICH DATO DMA CYCLES (FOR STORING THE
                                        ;       MESSAGE PACKET) ARE PERFORMED.  ANY ERRORS IN THE BODY OF THE
                                        ;       TEST (I.E. ERRORS OTHER THAN INITIALIZATION ERRORS RELATED TO
                                        ;       THE TRANSPORT BUS) DEFINITELY INDICATE A BAD M7196 MODULE.
                                        ;
                                        ;
                                        ;
030724                                          BGNTST
030724                                                                                          T7::
030724  012700  034213                          MOV     #TST7ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
030730  004737  016510                          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
030734  012737  000024  002214                  MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
030742                                  T7LOOP:
030742                                          BGNSUB                          ;////////// BEGIN SUBTEST ////////////
030742                                                                                          T7.1:
030742  104402                                                                                          TRAP    C$BSUB
030744  004737  034242                          JSR     PC,T7REST               ;SET PACKET TO START-UP VALUES

030750                                          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
030750  012700  000000                                                                                  MOV     #PRI00,R0
030754  104441                                                                                          TRAP    C$SPRI
030756  012703  002762                          MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
030762  012704  033060                          MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
030766  012764  000010  000006                  MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
030774                                  5$:
030774                                          BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
030774  104404                                                                                          TRAP    C$BSEG

030776  004737  015774                          JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
031002  103405                                  BCS     10$                     ;BR IF SOFT INIT = OK
031004  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
031006                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
031006  104455                                                                                          TRAP    C$ERDF
031010  001275                                                                                          .WORD   701
031012  003650                                                                                          .WORD   SFIERR
031014  012034                                                                                          .WORD   SFIMSG
031016  005037  002220                  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
031022  005037  002222                          CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
031026  010465  000000                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
031032  004737  016336                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
031036  103407                                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
031040  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
031042                                          ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
031042  104455                                                                                          TRAP    C$ERDF
031044  001276                                                                                          .WORD   702
031046  033661                                                                                          .WORD   T7SSR
```

```
     031050  012046                                                                    .WORD   PKTSSR
1220 031052  005237  002220                INC     FATFLG                  ;SET FATAL ERROR FLAG
1221 031056                         15$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     031056  104406                                                                    TRAP    C$CLP1
1222 031060                                ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     031060  104410                                                                    TRAP    C$ESCAPE
     031062  000152                                                                    .WORD   10000$ .
1223 031064  005737  002222                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1224 031070  001404                        BEQ     22$                     ;BRANCH IF NOT
1228 031072                                ERRHRD  ERRNO,T7INT,PKTSSR
     031072  104456                                                                    TRAP    C$ERHRD
     031074  001277                                                                    .WORD   703
     031076  034041                                                                    .WORD   T7INT
     031100  012046                                                                    .WORD   PKTSSR
1229 031102  016501  000002         22$:   MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1230 031106  012702  000200                MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
1231 031112  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1232 031116  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1233 031120  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1234 031124  020201                 25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1235 031126  001404                        BEQ     30$                     ;OKAY IF MATCH
1239 031130                                ERRHRD  ERRNO,T7NBA,PKTSSR      ;NBA NOT ZERO
     031130  104456                                                                    TRAP    C$ERHRD
     031132  001300                                                                    .WORD   704
     031134  033220                                                                    .WORD   T7NBA
     031136  012046                                                                    .WORD   PKTSSR
1240 031140                         30$:   CKLOOP                          ;LOOP ON ERROR ?
     031140  104406                                                                    TRAP    C$CLP1
1241 031142  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1242 031146  016501  000000                MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
1243 031152  012702  033106                MOV     #T7BFR,R2               ;START OF THE DATA BUFFER
1244 031156  032762  000200  000012        BIT     #BIT7,XST2(R2)          ;IS EXTENDED FEATURES BIT SET ?
1245 031164  001404                        BEQ     32$                     ;BRANCH IF EXTENDED FEATURES OFF
1246 031166  005237  002224                INC     EXTFEA                  ;SET EXTENDED FEATURES FOR SUBTEST 6
1247 031172  062702  000002                ADD     #2,R2                   ;EXTRA WORD IF SPECIAL FEATURES
1248 031176  062702  000016         32$:   ADD     #14.,R2                 ;EXPECTED CONTENTS OF TSDA
1249 031202  020102                        CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1250 031204  001404                        BEQ     35$                     ;ERROR IF NOT EQUAL
1254 031206                                ERRHRD  ERRNO,T7TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     031206  104456                                                                    TRAP    C$ERHRD
     031210  001301                                                                    .WORD   705
     031212  034130                                                                    .WORD   T7TSBA
     031214  015474                                                                    .WORD   EXPREC
1255
1256
1257 031216  004737  011114         35$:   JSR     PC,CKRAM                ;SEE IF DATA IN RAM IS CORRECT
1258 031222  103404                        BCS     40$                     ;BRANCH IF PACKET IN RAM IS CORRECT
1262 031224                                ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
     031224  104456                                                                    TRAP    C$ERHRD
     031226  001302                                                                    .WORD   706
     031230  004743                                                                    .WORD   PKTRAM
     031232  015510                                                                    .WORD   RAMERR
1263
1264 031234                         40$:   ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     031234                                                                            10000$:
     031234  104405                                                                    TRAP    C$ESEG
1265
```

```
031236  012364  000006                  MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
031242  020327  003060                  CMP     R3,#TBLEND              ;HAS ALL DATA BEEN TESTED ?
031246  103002                          BHIS    55$                     ;BRANCH IF ALL DATA DONE
031250  000137  030774                  JMP     5$                      ;BRANCH TILL BACK TO ZERO

031254                          55$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
031254                                                                          L10056:
031254  104403                                                                          TRAP    C$ESUB

031256  005737  002220                  TST     FATFLG                  ;ANY FATAL ERRORS ?
031262  001402                          BEQ     60$                     ;BRANCH IF NOT
031264  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
031270                          60$:

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

031270                                  BGNSUB                          ;///////////// BEGIN SUBTEST ///////////
031270                                                                          T7.2:
031270  104402                                                                          TRAP    C$BSUB

031272                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
031272  012700  000000                                                                  MOV     #PRI00,R0
031276  104441                                                                          TRAP    C$SPRI
031300  012703  033126                  MOV     #T72DATA,R3             ;START OF TEST DATA FOR SUBTEST
031304  012704  033060          5$:     MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
031310  004737  034242                  JSR     PC,T7REST               ;RESTORE PACKET TO STARTING VALUES

031314                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
031314  104404                                                                          TRAP    C$BSEG

031316  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
031322  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
031324  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
031326                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
031326  104455                                                                          TRAP    C$ERDF
031330  001303                                                                          .WORD   707
031332  003650                                                                          .WORD   SFIERR
031334  012034                                                                          .WORD   SFIMSG
031336  005037  002222          10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
031342  010400                          MOV     R4,R0                   ;START OF THE COMMAND PACKET
031344  061300                          ADD     (R3),R0                 ;OFFSET TO THE DATA WORD TO TEST
031346  056310  000002                  BIS     2(R3),(R0)              ;SET THE DATA BITS TO BE TESTED
031352  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
031356  004737  016250                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
031362  103405                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
031364  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
031366                                  ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
031366  104455                                                                          TRAP    C$ERDF
031370  001304                                                                          .WORD   708
031372  033661                                                                          .WORD   T7SSR
```

```
        031374  012046                                                                    .WORD   PKTSSR
   1315 031376                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        031376  104406                                                                    TRAP    C$CLP1
   1316 031400                                  ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
        031400  104410                                                                    TRAP    C$ESCAPE
        031402  000116                                                                    .WORD   10000$ .
   1317 031404  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
   1318 031410  001404                          BEQ     22$                     ;BRANCH IF NOT
   1322 031412                                  ERRHRD  ERRNO,T7INT,PKTSSR
        031412  104456                                                                    TRAP    C$ERHRD
        031414  001305                                                                    .WORD   709
        031416  034041                                                                    .WORD   T7INT
        031420  012046                                                                    .WORD   PKTSSR
   1323 031422  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
   1324 031426  012702  102206                  MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
   1325 031432  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
   1326 031436  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
   1327 031440  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
   1328 031444  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
   1329 031446  001414                          BEQ     30$                     ;OKAY IF MATCH
   1330 031450  010100                          MOV     R1,R0                   ;DATA FROM TSSR
   1331 031452                                  XOR     R2,R0                   ;FIND BITS IN ERROR
   1332 031462  020027  002000                  CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
   1333 031466  001404                          BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
   1337 031470                                  ERRHRD  ERRNO,T72REJ,PKTSSR     ;COMMAND NOT REJECTED
        031470  104456                                                                    TRAP    C$ERHRD
        031472  001306                                                                    .WORD   710
        031474  033273                                                                    .WORD   T72REJ
        031476  012046                                                                    .WORD   PKTSSR
   1338 031500                          30$:    CKLOOP                          ;LOOP ON ERROR ?
        031500  104406                                                                    TRAP    C$CLP1
   1339 031502  032701  002000                  BIT     #NBA,R1                 ;IS NBA BIT SET ?
   1340 031506  001004                          BNE     35$                     ;OKAY IF NBA SET
   1344 031510                                  ERRHRD  ERRNO,T72NBA,PKTSSR     ;NBA NOT SET
        031510  104456                                                                    TRAP    C$ERHRD
        031512  001307                                                                    .WORD   711
        031514  033142                                                                    .WORD   T72NBA
        031516  012046                                                                    .WORD   PKTSSR
   1345 031520                          35$:
   1346 031520                                  ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        031520                                                                          10000$:
        031520  104405                                                                    TRAP    C$ESEG
   1347
   1348 031522  062703  000004                  ADD     #4,R3                   ;POINT TO NEXT DATA PAIR
   1349 031526  020327  033142                  CMP     R3,#T72DONE             ;COMPARE TO END OF TEST DATA
   1350 031532  103002                          BHIS    57$                     ;BRANCH IF ALL DATA TESTED
   1351 031534  000137  031304                  JMP     5$                      ;BRANCH TILL BACK TO ZERO
   1352
   1353 031540                          57$:    ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
        031540                                                                          L10057:
        031540  104403                                                                    TRAP    C$ESUB
   1354
   1355                                 ;+
   1356                                 ;
   1357                                 ;TEST 7, SUBTEST 3
   1358                                 ;
   1359                                 ;CHECK THE WRITE CHARACTERISTICS COMMAND IS REJECTED
```

```
                                        ;IF ISSUED WITH AN INVALID DATA BLOCK BYTE COUNT
                                        ;
                                        ;

031542                                  BGNSUB          ;////////////// BEGIN SUBTEST ////////////
031542                                                                          T7.3:
031542  104402                                                                          TRAP    C$BSUB

031544                                  SETPRI  #PRI00  ;LOWER PRIORITY TO ALLOW INTERRUPTS
031544  012700  000000                                                                  MOV     #PRI00,R0
031550  104441                                                                          TRAP    C$SPRI
031552  012703  000001                  MOV     #1,R3   ;STARTING BYTE COUNT
031556  012704  033060          5$:     MOV     #T7PACKET,R4    ;GET THE ADDRESS OF COMMAND PACKET
031562  004737  034242                  JSR     PC,T7REST       ;RESTORE PACKET TO STARTING VALUES

031566                                  BGNSEG          ;>>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
031566  104404                                                                          TRAP    C$BSEG

031570  004737  015774                  JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
031574  103405                          BCS     10$             ;BR IF SOFT INIT = OK
031576  010001                          MOV     R0,R1           ;SAVE CONTENTS OF TSSR
031600                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
031600  104455                                                                          TRAP    C$ERDF
031602  001310                                                                          .WORD   712
031604  003650                                                                          .WORD   SFIERR
031606  012034                                                                          .WORD   SFIMSG
031610  005037  002222          10$:    CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
031614  010364  000006                  MOV     R3,PKBCNT(R4)   ;INSERT THE BYTE COUNT FOR TEST
031620  010465  000000                  MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
031624  004737  016250                  JSR     PC,WAITF        ;WAIT FOR SSR TO SET
031630  103405                          BCS     15$             ;BR IF CARRY SET (GOOD RETURN)
031632  010001                          MOV     R0,R1           ;SAVE CONTENTS OF TSSR
031634                                  ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
031634  104455                                                                          TRAP    C$ERDF
031636  001311                                                                          .WORD   713
031640  033661                                                                          .WORD   T7SSR
031642  012046                                                                          .WORD   PKTSSR
031644                          15$:    CKLOOP          ;LOOP ON ERROR, IF FLAG SET
031644  104406                                                                          TRAP    C$CLP1
031646                                  ESCAPE  SEG     ;BY-PASS SUBTEST IF FATAL ERROR
031646  104410                                                                          TRAP    C$ESCAPE
031650  000116                                                                          .WORD   100000$ .
031652  005737  002222                  TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
031656  001404                          BEQ     22$             ;BRANCH IF NOT
031660                                  ERRHRD  ERRNO,T7INT,PKTSSR
031660  104456                                                                          TRAP    C$ERHRD
031662  001312                                                                          .WORD   714
031664  034041                                                                          .WORD   T7INT
031666  012046                                                                          .WORD   PKTSSR
031670  016501  000002          22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
031674  012702  102206                  MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
031700  032701  000100                  BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
031704  001402                          BEQ     25$             ;BRANCH IF NOT OFF-LINE
031706  052702  000100                  BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
031712  020201                  25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
031714  001414                          BEQ     30$             ;OKAY IF MATCH
031716  010100                          MOV     R1,R0           ;DATA FROM TSSR
```

```
031720                                  XOR     R2,R0                   ;FIND BITS IN ERROR
031730  020027  002000                  CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
031734  001404                          BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
031736                                  ERRHRD  ERRNO,T73REJ,PKTSSR     ;COMMAND NOT REJECTED
031736  104456                                                                                    TRAP    C$ERHRD
031740  001313                                                                                    .WORD   715
031742  033372                                                                                    .WORD   T73REJ
031744  012046                                                                                    .WORD   PKTSSR
031746                          30$:    CKLOOP                          ;LOOP ON ERROR ?
031746  104406                                                                                    TRAP    C$CLP1
031750  032701  002000                  BIT     #NBA,R1                 ;IS NBA BIT SET ?
031754  001004                          BNE     35$                     ;OKAY IF NBA SET
031756                                  ERRHRD  ERRNO,T72NBA,PKTSSR     ;NBA NOT SET
031756  104456                                                                                    TRAP    C$ERHRD
031760  001314                                                                                    .WORD   716
031762  033142                                                                                    .WORD   T72NBA
031764  012046                                                                                    .WORD   PKTSSR
031766                          35$:
031766                                  ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
031766                                                                                            10000$:
031766  104405                                                                                    TRAP    C$ESEG

031770  005203                          INC     R3                      ;NEXT BYTE COUNT
031772  020327  000006                  CMP     R3,#6                   ;TESTED ALL INVALID ?
031776  002002                          BGE     57$                     ;BRANCH IF TEST DONE
032000  000137  031556                  JMP     5$                      ;BRANCH TILL BACK TO ZERO

                                57$:    ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
032004                                                                                            L10060:
032004  104403                                                                                    TRAP    C$ESUB

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

032006                                  BGNSUB                          ;//////////// BEGIN SUBTEST ////////////
032006                                                                                            T7.4:
032006  104402                                                                                    TRAP    C$BSUB

032010                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
032010  012700  000000                                                                            MOV     #PRI00,R0
032014  104441                                                                                    TRAP    C$SPRI
032016  012703  033126                  MOV     #T72DATA,R3             ;START OF TEST DATA FOR SUBTEST
032022  012704  033060          5$:     MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
032026  004737  034242                  JSR     PC,T7REST               ;RESTORE PACKET TO STARTING VALUES


032032  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
032036  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
032040  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
032042                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
032042  104455                                                                                    TRAP    C$ERDF
```

```
       032044  001315                                                                          .WORD   717
       032046  003650                                                                          .WORD   SFIERR
       032050  012034                                                                          .WORD   SFIMSG
  1454 032052  005037  002222          10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
  1455 032056  052737  000001  033070          BIS     #1,T7DATA               ;MAKE ADDRESS ODD
  1456 032064  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
  1457 032070  004737  016250                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
  1458 032074  103405                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
  1459 032076  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  1463 032100                                  ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
       032100  104455                                                                          TRAP    C$ERDF
       032102  001316                                                                          .WORD   718
       032104  033661                                                                          .WORD   T7SSR
       032106  012046                                                                          .WORD   PKTSSR
  1464 032110                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
       032110  104406                                                                          TRAP    C$CLP1
  1465 032112                                  ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
       032112  104410                                                                          TRAP    C$ESCAPE
       032114  000116                                                                          .WORD   L10061 .
  1466 032116  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
  1467 032122  001404                          BEQ     22$                     ;BRANCH IF NOT
  1471 032124                                  ERRHRD  ERRNO,T7INT,PKTSSR
       032124  104456                                                                          TRAP    C$ERHRD
       032126  001317                                                                          .WORD   719
       032130  034041                                                                          .WORD   T7INT
       032132  012046                                                                          .WORD   PKTSSR
  1472 032134  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
  1473 032140  012702  102206                  MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
  1474 032144  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
  1475 032150  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
  1476 032152  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
  1477 032156  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
  1478 032160  001414                          BEQ     30$                     ;OKAY IF MATCH
  1479 032162  010100                          MOV     R1,R0                   ;DATA FROM TSSR
  1480 032164                                  XOR     R2,R0                   ;FIND BITS IN ERROR
  1481 032174  020027  002000                  CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
  1482 032200  001404                          BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
  1486 032202                                  ERRHRD  ERRNO,T74REJ,PKTSSR     ;COMMAND NOT REJECTED
       032202  104456                                                                          TRAP    C$ERHRD
       032204  001320                                                                          .WORD   720
       032206  033465                                                                          .WORD   T74REJ
       032210  012046                                                                          .WORD   PKTSSR
  1487 032212                          30$:    CKLOOP                          ;LOOP ON ERROR ?
       032212  104406                                                                          TRAP    C$CLP1
  1488 032214  032701  002000                  BIT     #NBA,R1                 ;IS NBA BIT SET ?
  1489 032220  001004                          BNE     35$                     ;OKAY IF NBA SET
  1493 032222                                  ERRHRD  ERRNO,T72NBA,PKTSSR     ;NBA NOT SET
       032222  104456                                                                          TRAP    C$ERHRD
       032224  001321                                                                          .WORD   721
       032226  033142                                                                          .WORD   T72NBA
       032230  012046                                                                          .WORD   PKTSSR
  1494
  1495 032232                          35$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
       032232                                                                          L10061:
       032232  104403                                                                          TRAP    C$ESUB
  1496
  1497
```

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

 032234                                         BGNSUB                          ;///////////// BEGIN SUBTEST ////////////
 032234                                                                                          T7.5:
 032234  104402                                                                                          TRAP    C$BSUB

 032236                                         SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
 032236  012700  000000                                                                                  MOV     #PRI00,R0
 032242  104441                                                                                          TRAP    C$SPRI
 032244  012703  000001                         MOV     #1,R3                   ;STARTING BUFFER LENGTH
 032250  012704  033060          5$:            MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
 032254  004737  034242                         JSR     PC,T7REST               ;RESTORE PACKET TO STARTING VALUES

 032260                                         BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
 032260  104404                                                                                          TRAP    C$BSEG

 032262  004737  015774                         JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 032266  103405                                 BCS     10$                     ;BR IF SOFT INIT - OK
 032270  010001                                 MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 032272                                         ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
 032272  104455                                                                                          TRAP    C$ERDF
 032274  001322                                                                                          .WORD   722
 032276  003650                                                                                          .WORD   SFIERR
 032300  012034                                                                                          .WORD   SFIMSG
 032302  005037  002222          10$:           CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 032306  010337  033074                         MOV     R3,T7DATA+4             ;INSERT THE BAD MESSAGE LENGTH
 032312  010465  000000                         MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 032316  004737  016250                         JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 032322  103405                                 BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 032324  010001                                 MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 032326                                         ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
 032326  104455                                                                                          TRAP    C$ERDF
 032330  001323                                                                                          .WORD   723
 032332  033661                                                                                          .WORD   T7SSR
 032334  012046                                                                                          .WORD   PKTSSR
 032336                          15$:           CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
 032336  104406                                                                                          TRAP    C$CLP1
 032340                                         ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
 032340  104410                                                                                          TRAP    C$ESCAPE
 032342  000116                                                                                          .WORD   10000$-.
 032344  005737  002222                         TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 032350  001404                                 BEQ     22$                     ;BRANCH IF NOT
 032352                                         ERRHRD  ERRNO,T7INT,PKTSSR
 032352  104456                                                                                          TRAP    C$ERHRD
 032354  001324                                                                                          .WORD   724
 032356  034041                                                                                          .WORD   T7INT
 032360  012046                                                                                          .WORD   PKTSSR
 032362  016501  000002          22$:           MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 032366  012702  102206                         MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
 032372  032701  000100                         BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
```

```
032376  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
032400  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
032404  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
032406  001414                          BEQ     30$                     ;OKAY IF MATCH
032410  010100                          MOV     R1,R0                   ;DATA FROM TSSR
032412                                  XOR     R2,R0                   ;FIND BITS IN ERROR
032422  020027  002000                  CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
032426  001404                          BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
032430                                  ERRHRD  ERRNO,T75REJ,PKTSSR     ;COMMAND NOT REJECTED
032430  104456                                                                                TRAP    C$ERHRD
032432  001325                                                                                .WORD   725
032434  033563                                                                                .WORD   T75REJ
032436  012046                                                                                .WORD   PKTSSR
032440                          30$:    CKLOOP                          ;LOOP ON ERROR ?
032440  104406                                                                                TRAP    C$CLP1
032442  032701  002000                  BIT     #NBA,R1                 ;IS NBA BIT SET ?
032446  001004                          BNE     35$                     ;OKAY IF NBA SET
032450                                  ERRHRD  ERRNO,T72NBA,PKTSSR     ;NBA NOT SET
032450  104456                                                                                TRAP    C$ERHRD
032452  001326                                                                                .WORD   726
032454  033142                                                                                .WORD   T72NBA
032456  012046                                                                                .WORD   PKTSSR
032460                          35$:
032460                                  ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
032460                                                                                  10000$:
032460  104405                                                                                TRAP    C$ESEG

032462  005203                          INC     R3                      ;NEXT BUFFER LENGTH
032464  020327  000016                  CMP     R3,#14.                 ;HAVE ALL BAD VALUES BEEN TESTED ?
032470  002002                          BGE     57$                     ;BRANCH IF ALL TESTED
032472  000137  032250                  JMP     5$                      ;BRANCH TILL BACK TO ZERO

032476                          57$:    ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
032476                                                                                  L10062:
032476  104403                                                                                TRAP    C$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.
                                ;
                                ;-
032500                                  BGNSUB                          ;//////////// BEGIN SUBTEST ////////////
032500                                                                                  T7.6:
032500  104402                                                                                TRAP    C$BSUB
032502  005737  002224                  TST     EXTFEA                  ;IS EXTENDED FEATURES SOFT. SW SET?
032506  001002                          BNE     4$                      ;BR. IF SOFTWARE SWITCH IS SET (ON)
032510                                  EXIT    TST                     ;NO EXTENDED FEATURES EXIT THIS TEST
032510  104432                                                                                TRAP    C$EXIT
032512  001576                                                                                .WORD   L10055-.
032514  004737  034242          4$:     JSR     PC,T7REST               ;SET PACKET TO START-UP VALUES
```

```
1590                                                                                    .
1591 032520                                 SETPRI  #PRIOO                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     032520  012700  000000                                                                     MOV     #PRIOO,RO
     032524  104441                                                                             TRAP    C$SPRI
1592 032526  012703  002764                 MOV     #TSTBLK+12.,R3          ;START OF TEST DATA
1593 032532  012704  033060                 MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1594 032536  012737  000200  033100         MOV     #200,T7SP               ;SPECIAL BIT SET FOR EXTFEA RAM RD
1595 032544  012764  000012  000006         MOV     #10.,PKBCNT(R4)         ;START WITH EXTENDED FEATURES VALUE
1596 032552                          5$:
1597 032552                                 BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     032552  104404                                                                             TRAP    C$BSEG
1598
1599 032554  004737  015774                 JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1600 032560  103405                         BCS     10$                     ;BR IF SOFT INIT = OK
1604 032562  010001                         MOV     RO,R1                   ;SAVE CONTENTS OF TSSR
1605 032564                                 ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     032564  104455                                                                             TRAP    C$ERDF
     032566  001327                                                                             .WORD   727
     032570  003650                                                                             .WORD   SFIERR
     032572  012034                                                                             .WORD   SFIMSG
1606 032574  005037  002220          10$:   CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1607 032600  005037  002222                 CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1608 032604  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
1609 032610  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1610 032614  103407                         BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1611 032616  010001                         MOV     RO,R1                   ;SAVE CONTENTS OF TSSR
1615 032620                                 ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     032620  104455                                                                             TRAP    C$ERDF
     032622  001330                                                                             .WORD   728
     032624  033661                                                                             .WORD   T7SSR
     032626  012046                                                                             .WORD   PKTSSR
1616 032630  005237  002220                 INC     FATFLG                  ;SET FATAL ERROR FLAG
1617 032634                          15$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     032634  104406                                                                             TRAP    C$CLP1
1618 032636                                 ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     032636  104410                                                                             TRAP    C$ESCAPE
     032640  000156                                                                             .WORD   10000$-.
1619 032642  005737  002222                 TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1620 032646  001404                         BEQ     22$                     ;BRANCH IF NOT
1624 032650                                 ERRHRD  ERRNO,T7INT,PKTSSR
     032650  104456                                                                             TRAP    C$ERHRD
     032652  001331                                                                             .WORD   729
     032654  034041                                                                             .WORD   T7INT
     032656  012046                                                                             .WORD   PKTSSR
1625 032660  016501  000002          22$:   MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1626 032664  012702  000200                 MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
1627 032670  032701  000100                 BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1628 032674  001402                         BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1629 032676  052702  000100                 BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1630 032702  020201                  25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1631 032704  001404                         BEQ     30$                     ;OKAY IF MATCH
1635 032706                                 ERRHRD  ERRNO,T7NBA,PKTSSR      ;NBA NOT ZERO
     032706  104456                                                                             TRAP    C$ERHRD
     032710  001332                                                                             .WORD   730
     032712  033220                                                                             .WORD   T7NBA
     032714  012046                                                                             .WORD   PKTSSR
1636 032716                          30$:   CKLOOP                          ;LOOP ON ERROR ?
```

```
        032716  104406                                                                    TRAP    C$CLP1
   1637 032720  004737  016336                JSR     PC,CHKTSSR          ;WAIT FOR READY, NON-AMBIGUOUS
   1638 032724  016501  000000                MOV     TSBA(R5),R1         ;GET TSBA REGISTER CONTENTS
   1639 032730  012702  033106                MOV     #T7BFR,R2           ;START OF THE DATA BUFFER
   1640 032734  032762  000200  000012        BIT     #BIT7,XST2(R2)      ;IS EXTENDED FEATURES BIT SET ?
   1641 032742  001402                        BEQ     32$                 ;BRANCH IF EXTENDED FEATURES OFF
   1642 032744  062702  000002                ADD     #2,R2               ;EXTRA WORD IF SPECIAL FEATURES
   1643 032750  062702  000016        32$:    ADD     #14.,R2             ;EXPECTED CONTENTS OF TSDA
   1644 032754  020102                        CMP     R1,R2               ;COMPARE EXPECTED TO RECEIVED
   1645 032756  001404                        BEQ     35$                 ;ERROR IF NOT EQUAL
   1649 032760                                ERRHRD  ERRNO,T7TSBA,EXPREC ;PRINT THE ERROR & EXPD/RECV
        032760  104456                                                                    TRAP    C$ERHRD
        032762  001333                                                                    .WORD   731
        032764  034130                                                                    .WORD   T7TSBA
        032766  015474                                                                    .WORD   EXPREC
   1650
   1651
   1652 032770  012704  033070        35$:    MOV     #T7DATA,R4          ;SET POINTER FOR CHECKER
   1653 032774  004737  011224                JSR     PC,CKRAM2           ;SEE IF DATA IN RAM IS CORRECT
   1654 033000  103404                        BCS     40$                 ;BRANCH IF PACKET IN RAM IS CORRECT
   1658 033002                                ERRHRD  ERRNO,PKTRAM,RAMERR ;REPORT THE RAM ERROR(S)
        033002  104456                                                                    TRAP    C$ERHRD
        033004  001334                                                                    .WORD   732
        033006  004743                                                                    .WORD   PKTRAM
        033010  015510                                                                    .WORD   RAMERR
   1659
   1660 033012  012704  033060        40$:    MOV     #T7PACKET,R4        ;RESET PACKET POINTER
   1661 033016                                ENDSEG                      ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        033016                                                                        10000$:
        033016  104405                                                                    TRAP    C$ESEG
   1662
   1663 033020  012364  000006                MOV     (R3)+,PKBCNT(R4)    ;SET THE TEST WORD
   1664 033024  020327  003060                CMP     R3,#TBLEND          ;HAS ALL DATA BEEN TESTED ?
   1665 033030  103002                        BHIS    55$                 ;BRANCH IF ALL DATA DONE
   1666 033032  000137  032552                JMP     5$                  ;BRANCH TILL BACK TO ZERO
   1667
   1668 033036                        55$:    ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        033036                                                                        L10063:
        033036  104403                                                                    TRAP    C$ESUB
   1669
   1670 033040  005737  002220                TST     FATFLG              ;ANY FATAL ERRORS ?
   1671 033044  001402                        BEQ     60$                 ;BRANCH IF NOT
   1672 033046  004737  017202                JSR     PC,CKDROP           ;TRY TO DROP THE UNIT
   1673 033052                        60$:
   1674 033052                                EXIT    TST                 ;ALL DONE THIS TEST
        033052  104432                                                                    TRAP    C$EXIT
        033054  001234                                                                    .WORD   L10055 .
   1675
   1676                               ;+
   1677                               ;LOCAL STORAGE FOR THIS TEST
   1678                               ;-
   1679
   1681 033056                                .BLKB   10-<.-TSV2&7>
   1683 033060                        T7PACKET:                           ;COMMAND PACKET FOR TEST
   1684 033060  100004                        .WORD   100004              ;WRITE CHARACTERISTICS COMMAND, WITH ACK
   1685 033062  033070                        .WORD   T7DATA              ;ADDRESS OF CHARACTERISTICS BLOCK
   1686 033064  000000                        .WORD   0
```

```
033066  000010                          .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE

033070                          T7DATA:                                 ;CHARACTERISTICS DATA BLOCK
033070  033106                          .WORD   T7BFR                   ;ADDRESS OF MESSAGE BUFFER
033072  000000                          .WORD   0
033074  000016                          .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
033076  000000                          .WORD   0
033100  000000                  T7SP:   .WORD   0                       ;EXTFEA EXTRA WORD

033102  000000  000000                  .WORD   0,0                     ;SPACE
033106                          T7BFR:  .BLKW   8.                      ;MESSAGE BUFFER

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

033126                          T72DATA:
033126  000000  037140                  .WORD   0,BIT5!BIT6!BIT9!BIT10!BIT11!BIT12!BIT13
033132  000002  000001                  .WORD   2,BIT0
033136  000004  100100                  .WORD   4,BIT6!BIT15
        033142                  T72DONE=.


                                ;+
                                ;LOCAL TEXT MESSAGES FOR TEST
                                ;-

033142     116     102     101  T72NBA: .ASCIZ  'NBA Not Set On Rejected WRITE CHARACTERISTICS
033220     127     122     111  T7NBA:  .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
033273     127     122     111  T72REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Non-Zero Unused Fields
033372     127     122     111  T73REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Data Count'
033465     127     122     111  T74REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Block Address
033563     127     122     111  T75REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Buffer Length
033661     103     157     156  T7SSR:  .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
033750     105     170     160  T7NINT: .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS
034041     125     156     145  T7INT:  .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
034130     111     156     143  T7TSBA: .ASCIZ  'Incorrect TSBA Address After WRITE CHARACTERISTICS'
034213     127     162     151  TST7ID: .ASCIZ  'Write Characteristics'
                                        .EVEN


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

034242                          T7REST:
034242                                  SAVREG                  ;SAVE THE REGISTERS
034246  012701  033060                  MOV     @T7PACKET,R1    ;START OF THE PACKET
```

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


                                .SBTTL  TEST  8: VOLUME CHECK


                        ;       THIS TEST VERIFIES THAT THE VOLUME CHECK (VCK) BIT, A FLAG HELD
                        ;       WITHIN THE M7196 AND APPEARING IN XSTO, IS SET BY INITIALIZE AND
                        ;       CLEARED BY EXECUTING A WRITE CHARACTERISTICS COMMAND WITH THE
                        ;       CVC BIT SET.  IT IS ALSO VERIFIED THAT A WRITE CHARACTERISTICS
                        ;       COMMAND WITH THE CVC BIT CLEAR DOES NOT AFFECT THE STATE OF THE
                        ;       VOLUME CHECK BIT.  THE ACTUAL FUNCTION OF VOLUME CHECK, THAT OF
                        ;       PREVENTING OR ALLOWING A TAPE MOTION COMMAND DEPENDING UPON
                        ;       WHETHER VOLUME CHECK IS SET OR CLEAR, IS NOT CHECKED BY THIS
                        ;       TEST;  THIS FUNCTIONALITY IS CHECKED IN THE INDIVIDUAL TESTS OF
                        ;       TAPE MOTION COMMANDS.
                        ;
                        ;       THE TEST PROCEEDS AS FOLLOWS:
                        ;
                        ;            1.  THE CONTROLLER IS INITIALIZED BY WRITING INTO THE TSSR.
                        ;
                        ;            2.  A WRITE CHARACTERISTICS COMMAND IS ISSUED (WITH CVC=0)
                        ;                AND XSTO IN THE RETURNED MESSAGE BUFFER IS EXAMINED;
                        ;                THE VCK BIT SHOULD BE CLEAR (0).
                        ;
                        ;            3.  THE PREVIOUS STEP IS REPEATED TO VERIFY THAT VCK DOES
                        ;                NOT CHANGE (REMAINS AT 0).
                        ;
                        ;            4.  A WRITE CHARACTERISTICS COMMAND IS ISSUED WITH CVC=1
                        ;                AND THE VCK BIT IN XSTO IN THE MESSAGE BUFFER IS
                        ;                EXAMINED;  THE VCK BIT SHOULD BE CLEAR (0).
                        ;
                        ;            5.  A WRITE CHARACTERISTICS COMMAND IS ISSUED WITH CVC=0
                        ;                AND THE VCK BIT IN XSTO IN THE MESSAGE BUFFER IS
                        ;                EXAMINED;  THE VCK BIT SHOULD REMAIN CLEAR (0).
                        ;
                        ;
034312                          BGNTST
034312                                                          T8::
034312  012700  035177          MOV     #TST8ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
034316  004737  016510          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
034322  012737  000024  002214  MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
034330                  T8LOOP:

034330  012704  034720          MOV     #T8PACKET,R4            ;PACKET FOR WRITE CHARACTERISTICS
```

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

```
        034522  104406                                                                          TRAP    C$CLP1
  1847  034524  052714  040000                BIS     #BIT14,(R4)             ;SET THE CVC BIT
  1848  034530  010465  000000                MOV     R4,T5DB(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
  1849  034534  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
  1850  034540  103405                        BCS     35$                     ;BR IF CARRY SET (GOOD RETURN)
  1851  034542  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  1855  034544                                ERRDF   ERRNO,T8SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
        034544  104455                                                                          TRAP    C$ERDF
        034546  001446                                                                          .WORD   806
        034550  035110                                                                          .WORD   T8SSR
        034552  012046                                                                          .WORD   PKTSSR
  1856  034554                        35$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        034554  104406                                                                          TRAP    C$CLP1
  1857  034556                                ESCAPE  TST                     ;EXIT IF FATAL ERROR
        034556  104410                                                                          TRAP    C$ESCAPE
        034560  000434                                                                          .WORD   L10064-.
  1858  034562  032762  000020  000006        BIT     #XSOVCK,XSTO(R2)        ;IS VOLUME CHECK CLEAR IN XSTO ?
  1859  034570  001406                        BEQ     40$                     ;OKAY IF VOLUME CHECK IS CLEARED
  1863  034572  016501  000002                MOV     TSSR(R5),R1             ;CONTENTS OF TSSR FOR ERROR REPORT
  1864  034576                                ERRHRD  ERRNO,T8VCK,PKTMES      ;VOLUME CHECK NOT CLEARED
        034576  104456                                                                          TRAP    C$ERHRD
        034600  001447                                                                          .WORD   807
        034602  034762                                                                          .WORD   T8VCK
        034604  012110                                                                          .WORD   PKTMES
  1865  034606                        40$:    CKLOOP                          ;LOOP ON ERROR ?
        034606  104406                                                                          TRAP    C$CLP1
  1866  034610  042714  040000                BIC     #BIT14,(R4)             ;CLEAR THE CVC BIT
  1867  034614  010465  000000                MOV     R4,T5DB(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
  1868  034620  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
  1869  034624  103405                        BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
  1870  034626  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  1874  034630                                ERRDF   ERRNO,T8SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
        034630  104455                                                                          TRAP    C$ERDF
        034632  001450                                                                          .WORD   80C
        034634  035110                                                                          .WORD   T8SSR
        034636  012046                                                                          .WORD   PKTSSR
  1875  034640                        45$:    CKLOOP                          ; OOP ON ERROR, IF FLAG SET
        034640  104406                                                                          TRAP    C$CLP1
  1876  034642                                ESCAPE  TST                     ;EXIT IF FATAL ERROR
        034642  104410                                                                          TRAP    C$ESCAPE
        034644  000350                                                                          .WORD   L10064 .
  1877  034646  032762  000020  000006        BIT     #XSOVCK,XSTO(R2)        ;IS VOLUME CHECK CLEAR IN XSTO ?
  1878  034654  001406                        BEQ     50$                     ;OKAY IF VOLUME CHECK IS CLEARED
  1882  034656  016501  000002                MOV     TSSR(R5),R1             ;CONTENTS OF TSSR FOR ERROR REPORT
  1883  034662                                ERRHRD  ERRNO,T8VCK,PKTMES      ;VOLUME CHECK NOT CLEARED
        034662  104456                                                                          TRAP    C$ERHRD
        034664  001451                                                                          .WORD   809
        034666  034762                                                                          .WORD   T8VCK
        034670  012110                                                                          .WORD   PKTMES
  1884  034672                        50$:    CKLOOP                          ;LOOP ON ERROR ?
        034672  104406                                                                          TRAP    C$CLP1
  1885  034674  004737  016456        60$:    JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
  1886  034700  103002                        BCC     62$                     ;BRANCH IF NOT
  1887  034702  000137  034330                JMP     T8LOOP                  ;LOOP UNTIL COUNT EXPIRED
  1888  034706                        62$:    EXIT    TST                     ;ALL DONE THIS TEST
        034706  104432                                                                          TRAP    C$EXIT
        034710  000304                                                                          .WORD   L10064 .
```

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

034712                                          .BLKB   10-<.-TSV2&7>
034720                                  T8PACKET:                               ;COMMAND PACKET FOR TEST
034720  100004                                  .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND
034722  034730                                  .WORD   T8DATA                  ;ADDRESS OF CHARACTERISTICS BLOCK
034724  000000                                  .WORD   0
034726  000010                                  .WORD   10                      ;STARTING VALUE OF COUNTER

034730                                  T8DATA:                                 ;CHARACTERISTICS DATA BLOCK
034730  034742                                  .WORD   T8BFR                   ;ADDRESS OF MESSAGE BUFFER
034732  000000                                  .WORD   0
034734  000020                                  .WORD   16.                     ;LENGTH OF MESSAGE BUFFER
034736  000000  000000                          .WORD   0,0

034742                                  T8BFR:  .BLKW   8.                      ;MESSAGE BUFFER


                                        ;+
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-

034762     126     157     154          T8VCK:  .ASCIZ  'Volume Check Bit Not Cleared'
035017     126     157     154          T8NVCK: .ASCIZ  'Volume Check Bit (VCK) Not Clear After Initialize (XSTO)'
035110     103     157     156          T8SSR:  .ASCIZ  'Contents of TSSR Incorrect After Write Characteristics
035177     126     157     154          TST8ID: .ASCIZ  'Volume Check'
                                        .EVEN
035214                                  ENDTST
035214                                                                  L10064:
035214  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 XSTO
                                ;       OF ANY MESSAGE PACKET IS SET AND THAT A COMPLETION INTERRUPT IS
                                ;       GENLRATED.  FINALLY, A SEQUENCE OF TWO COMMANDS ARE ISSUED, THE
                                ;       FIRST WITH IE=1 AND THE SECOND WITH IE=0.  IT IS VERIFIED THAT
                                ;       NO INTERRUPT IS GENERATED AFTER THE SECOND COMMAND AND THAT THE
                                ;       IE BIT IN XSTO IS 0.
                                ;
035216                                  BGNTST
035216                                                                  T9::
035216  005037  002224                  CLR     EXTFEA                  ;CLEAR EXTENDED FEATURES SWITCH
035222  012700  040241                  MOV     #TST9ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
```

```
035226  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
035232  012737  000024  002214          MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
035240                          T9LOOP:
035240                                  BGNSUB                          ;//////////// BEGIN SUBTEST ////////////
035240                                                                                  T9.1:
035240  104402                                                                                  TRAP    C$BSUB

035242  004737  040266                  JSR     PC,T9REST               ;SET PACKET TO INITIAL VALUES
035246                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
035246  012700  000000                                                                          MOV     #PRI00,R0
035252  104441                                                                                  TRAP    C$SPRI
035254  012703  002762                  MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
035260  012704  037170                  MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
035264  012764  000010  000006          MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
035272                          5$:
035272                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
035272  104404                                                                                  TRAP    C$BSEG

035274  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
035300  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
035302  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
035304                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
035304  104455                                                                                  TRAP    C$ERDF
035306  001605                                                                                  .WORD   901
035310  003650                                                                                  .WORD   SFIERR
035312  012034                                                                                  .WORD   SFIMSG
035314  005037  002220          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
035320  005037  002222                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
035324  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
035330  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
035334  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
035336  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
035340                                  ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
035340  104455                                                                                  TRAP    C$ERDF
035342  001606                                                                                  .WORD   902
035344  037707                                                                                  .WORD   T9SSR
035346  012046                                                                                  .WORD   PKTSSR
035350  005237  002220                  INC     FATFLG                  ;SET FATAL ERROR FLAG
035354                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
035354  104406                                                                                  TRAP    C$CLP1
035356                                  ESCAPE  SEG                     ;BY PASS SUBTEST IF FATAL ERROR
035356  104410                                                                                  TRAP    C$ESCAPE
035360  000056                                                                                  .WORD   10000$ .
035362  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
035366  001004                          BNE     22$                     ;BRANCH IF YES
035370                                  ERRHRD  ERRNO,T9NINT,PKTSSR
035370  104456                                                                                  TRAP    C$ERHRD
035372  001607                                                                                  .WORD   903
035374  037776                                                                                  .WORD   T9NINT
035376  012046                                                                                  .WORD   PKTSSR
035400  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
035404  012702  000200                  MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
035410  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
035414  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
035416  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
035422  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
035424  001404                          BEQ     30$                     ;OKAY IF MATCH
```

```
035426                                  ERRHRD  ERRNO,T9NBA,PKTSSR     ;NBA NOT ZERO
035426  104456                                                                          TRAP    C$ERHRD
035430  001610                                                                          .WORD   904
035432  037246                                                                          .WORD   T9NBA
035434  012046                                                                          .WORD   PKTSSR
035436                          30$:
035436                                  ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
035436                                                                                  100008:
035436  104405                                                                          TRAP    C$ESEG

035440  012364  000006                  MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
035444  020327  003060                  CMP     R3,#TBLEND              ;HAS ALL DATA BEEN TESTED ?
035450  103002                          BHIS    55$                     ;BRANCH IF ALL DATA DONE
035452  000137  035272                  JMP     5$                      ;BRANCH TILL BACK TO ZERO

035456                          55$:    ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
035456                                                                                  L10066:
035456  104403                                                                          TRAP    C$ESUB

035460  005737  002220                  TST     FATFLG                  ;ANY FATAL ERRORS ?
035464  001402                          BEQ     60$                     ;BRANCH IF NOT
035466  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
035472  033727  037224  00C200  60$:    BIT     T9BFR+12,#BIT7          ;EXTENDED FEATURES SET?
035500  001402                          BEQ     70$                     ;BR IF NO
035502  005237  002224                  INC     EXTFEA                  ;SET EXT FEATURE FLAG
035506                          70$:

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

035506                                  BGNSUB                          ;//////////// BEGIN SUBTEST ////////////
035506                                                                                  T9.2:
035506  104402                                                                          TRAP    C$BSUB

035510                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
035510  012700  000000                                                                  MOV     #PRI00,R0
035514  104441                                                                          TRAP    C$SPRI
035516  012703  037232                  MOV     #T92DATA,R3             ;START OF TEST DATA FOR SUBTEST
035522  012704  037170          5$:     MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
035526  004737  040266                  JSR     PC,T9REST               ;RESTORE PACKET TO STARTING VALUES

035532                                  BGNSEG                          ;>>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
035532  104404                                                                          TRAP    C$BSEG

035534  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
035540  103405                          BCS     10$                     ;BR IF SOFT INIT - OK
035542  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
035544                                  ERROF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
035544  104455                                                                          TRAP    C$EROF
035546  001611                                                                          .WORD   905
035550  003650                                                                          .WORD   SFIERR
```

```
     035552  012034                                                                    .WORD   SFIMSG
2042 035554  005037  002222          10$:   CLR     INTRECV                ;CLEAR INTERRUPT RECEIVED FLAG
2043 035560  010400                         MOV     R4,R0                  ;START OF THE COMMAND PACKET
2044 035562  061300                         ADD     (R3),R0                ;OFFSET TO THE DATA WORD TO TEST
2045 035564  056310  000002                 BIS     2(R3),(R0)             ;SET THE DATA BITS TO BE TESTED
2046 035570  010465  000000                 MOV     R4,TSDR(R5)            ;SET THE PACKET ADDRESS
2047 035574  004737  016250                 JSR     PC,WAITF               ;WAIT FOR SSR TO SET
2048 035600  103405                         BCS     15$                    ;BR IF CARRY SET (GOOD RETURN)
2049 035602  010001                         MOV     R0,R1                  ;SAVE CONTENTS OF TSSR
2053 035604                                 ERRDF   ERRNO,T9SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     035604  104455                                                                    TRAP    C$ERDF
     035606  001612                                                                    .WORD   906
     035610  037707                                                                    .WORD   T9SSR
     035612  012046                                                                    .WORD   PKTSSR
2054 035614                          15$:   CKLOOP                         ;LOOP ON ERROR, IF FLAG SET
     035614  104406                                                                    TRAP    C$CLP1
2055 035616                                 ESCAPE  SEG                    ;BY-PASS CHECKS IF FATAL ERROR
     035616  104410                                                                    TRAP    C$ESCAPE
     035620  000056                                                                    .WORD   10000$ .
2056 035622  005737  002222                 TST     INTRECV                ;DID AN INTERRUPT OCCUR ?
2057 035626  001004                         BNE     22$                    ;BRANCH IF YES
2061 035630                                 ERRHRD  ERRNO,T9NINT,PKTSSR
     035630  104456                                                                    TRAP    C$ERHRD
     035632  001613                                                                    .WORD   907
     035634  037776                                                                    .WORD   T9NINT
     035636  012046                                                                    .WORD   PKTSSR
2062 035640  016501  000002          22$:   MOV     TSSR(R5),R1            ;GET THE CONTENTS OF TSSR
2063 035644  012702  102206                 MOV     #SC!SSR!TSREJ!NBA,R2   ;EXPECTED CONTENTS OF TSSR
2064 035650  032701  000100                 BIT     #OFL,R1                ;IS OFF-LINE BIT SET ?
2065 035654  001402                         BEQ     25$                    ;BRANCH IF NOT OFF-LINE
2066 035656  052702  000100                 BIS     #OFL,R2                ;SET OFF-LINE IN EXPECTED DATA
2067 035662  020201                  25$:   CMP     R2,R1                  ;DOES EXPECTED MATCH RECEIVED ?
2068 035664  001404                         BEQ     30$                    ;OKAY IF MATCH
2072 035666                                 ERRHRD  ERRNO,T92REJ,PKTSSR    ;COMMAND NOT REJECTED
     035666  104456                                                                    TRAP    C$ERHRD
     035670  001614                                                                    .WORD   908
     035672  037321                                                                    .WORD   T92REJ
     035674  012046                                                                    .WORD   PKTSSR
2073 035676                          30$:
2074 035676                                 ENDSEG                         ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     035676                                                                            10000$:
     035676  104405                                                                    TRAP    C$ESEG
2075
2076 035700  062703  000004                 ADD     #4,R3                  ;POINT TO NEXT DATA PAIR
2077 035704  020327  037246                 CMP     R3,#T92DONE            ;COMPARE TO END OF TEST DATA
2078 035710  103002                         BHIS    59$                    ;BRANCH IF ALL DATA TESTED
2079 035712  000137  035522                 JMP     5$                     ;BRANCH TILL BACK TO ZERO
2080
2081 035716                          59$:   ENDSUB                         ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     035716                                                                            L10067:
     035716  104403                                                                    TRAP    C$ESUB
2082
2083                                 ;*
2084                                 ;
2085                                 ;TEST 9, SUBTEST 3
2086                                 ;
2087                                 ;CHECK THE WRITE CHARACTERISTICS COMMAND IS REJECTED
```

```
                                        ;IF ISSUED WITH AN INVALID DATA BLOCK BYTE COUNT
                                        ;
                                        ;

035720                                  BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
035720                                                                                  T9.3:
035720  104402                                                                                  TRAP    C$BSUB

035722                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
035722  012700  000000                                                                          MOV     #PRI00,R0
035726  104441                                                                                  TRAP    C$SPRI
035730  012703  000001                  MOV     #1,R3                   ;STARTING BYTE COUNT
035734  012704  037170          5$:     MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
035740  004737  040266                  JSR     PC,T9REST               ;RESTORE PACKET TO STARTING VALUES

035744                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
035744  104404                                                                                  TRAP    C$BSEG

035746  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
035752  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
035754  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
035756                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
035756  104455                                                                                  TRAP    C$EPOF
035760  001615                                                                                  .WORD   909
035762  003650                                                                                  .WORD   SFIERR
035764  012034                                                                                  .WORD   SFIMSG
035766  005037  002222          10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
035772  010364  000006                  MOV     R3,PKBCNT(R4)           ;INSERT THE BYTE COUNT FOR TEST
035776  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
036002  004737  016250                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
036006  103405                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
036010  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
036012                                  ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
036012  104455                                                                                  TRAP    C$ERDF
036014  001616                                                                                  .WORD   910
036016  037707                                                                                  .WORD   T9SSR
036020  012046                                                                                  .WORD   PKTSSR
036022                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
036022  104406                                                                                  TRAP    C$CLP1
036024                                  ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
036024  104410                                                                                  TRAP    C$ESCAPE
036026  000056                                                                                  .WORD   10000$ .
036030  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
036034  001004                          BNE     22$                     ;BRANCH IF YES
036036                                  ERRHRD  ERRNO,T9NINT,PKTSSR
036036  104456                                                                                  TRAP    C$ERHRD
036040  001617                                                                                  .WORD   911
036042  037776                                                                                  .WORD   T9NINT
036044  012046                                                                                  .WORD   PKTSSR
036046  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
036052  012702  102206                  MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
036056  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
036062  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
036064  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
036070  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
036072  001404                          BEQ     30$                     ;OKAY IF MATCH
036074                                  ERRHRD  ERRNO,T93REJ,PKTSSR     ;COMMAND NOT REJECTED
```

```
        036074  104456                                                                    TRAP    C$ERMRD
        036076  001620                                                                    .WORD   912
        036100  037420                                                                    .WORD   T93REJ
        036102  012046                                                                    .WORD   PKTSSR
   2137 036104                          30$:
   2138 036104                                  ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        036104                                                                                  10000$:
        036104  104405                                                                    TRAP    C$ESEG
   2139
   2140 036106  005203                          INC     R3                      ;NEXT BYTE COUNT
   2141 036110  020327  000006                  CMP     R3,#6                   ;TESTED ALL INVALID ?
   2142 036114  002002                          BGE     59$                     ;BRANCH IF TEST DONE
   2143 036116  000137  035734                  JMP     5$                      ;BRANCH TILL BACK TO ZERO
   2144
   2145 036122                          59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        036122                                                                                  L10070:
        036122  104403                                                                    TRAP    C$ESUB
   2146
   2147                                 ;+
   2148                                 ;
   2149                                 ;TEST 9, SUBTEST 4
   2150                                 ;
   2151                                 ;SUBTEST TO VERIFY THAT A WRITE CHARACTERISTICS COMMAND IS
   2152                                 ;REJECTED IF AN ILLEGAL DATA BLOCK ADDRESS IS ISSUED.
   2153                                 ;
   2154                                 ;-
   2155
   2156 036124                                  BGNSUB                          ;//////////// BEGIN SUBTEST ////////////
        036124                                                                                  T9.4:
        036124  104402                                                                    TRAP    C$BSUB
   2157
   2158 036126                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
        036126  012700  000000                                                            MOV     #PRI00,R0
        036132  104441                                                                    TRAP    C$SPRI
   2159 036134  012703  037232                  MOV     #T92DATA,R3             ;START OF TEST DATA FOR SUBTEST
   2160 036140  012704  037170          5$:     MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
   2161 036144  004737  040266                  JSR     PC,T9REST               ;RESTORE PACKET TO STARTING VALUES
   2162
   2163
   2164 036150  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
   2165 036154  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
   2169 036156  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
   2170 036160                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
        036160  104455                                                                    TRAP    C$ERDF
        036162  001621                                                                    .WORD   913
        036164  003650                                                                    .WORD   SFIERR
        036166  012034                                                                    .WORD   SFIMSG
   2171 036170  005037  002222          10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
   2172 036174  052737  000001  037200          BIS     #1,T9DATA               ;MAKE ADDRESS ODD
   2173 036202  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
   2174 036206  004737  016250                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
   2175 036212  103405                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
   2176 036214  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
   2180 036216                                  ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
        036216  104455                                                                    TRAP    C$ERDF
        036220  001622                                                                    .WORD   914
        036222  037707                                                                    .WORD   T9SSR
```

```
     036224  012046                                                                  .WORD   PKTSSR
2181 036226                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036226  104406                                                                  TRAP    C$CLP1
2182 036230                                  ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
     036230  104410                                                                  TRAP    C$ESCAPE
     036232  000056                                                                  .WORD   L10071-.
2183 036234  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2184 036240  001004                          BNE     22$                     ;BRANCH IF YES
2188 036242                                  ERRHRD  ERRNO,T9NINT,PKTSSR
     036242  104456                                                                  TRAP    C$ERHRD
     036244  001623                                                                  .WORD   915
     036246  037776                                                                  .WORD   T9NINT
     036250  012046                                                                  .WORD   PKTSSR
2189 036252  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
2190 036256  012702  102206                  MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
2191 036262  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
2192 036266  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
2193 036270  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2194 036274  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2195 036276  001404                          BEQ     30$                     ;OKAY IF MATCH
2199 036300                                  ERRHRD  ERRNO,T94REJ,PKTSSR     ;COMMAND NOT REJECTED
     036300  104456                                                                  TRAP    C$ERHRD
     036302  001624                                                                  .WORD   916
     036304  037513                                                                  .WORD   T94REJ
     036306  012046                                                                  .WORD   PKTSSR
2200 036310                          30$:
2201
2202 036310                                  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     036310                                                                          L10071:
     036310  104403                                                                  TRAP    C$ESUB
2203
2204
2205                                 ;+
2206                                 ;
2207                                 ;TEST 9, SUBTEST 5
2208                                 ;
2209                                 ;CHECK THAT WRITE CHARACTERISTICS COMMAND IS REJECTED IF THE
2210                                 ;MESSAGE BUFFER LENGTH SPECIFIES AN INVALID COUNT (LESS THAN 14)
2211                                 ;
2212                                 ;-
2213
2214 036312                                  BGNSUB                          ;///////////// BEGIN SUBTEST ///////////
     036312                                                                          T9.5:
     036312  104402                                                                  TRAP    C$BSUB
2215
2216 036314                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     036314  012700  000000                                                          MOV     #PRI00,R0
     036320  104441                                                                  TRAP    C$SPRI
2217 036322  012703  000001                  MOV     #1,R3                   ;STARTING BUFFER LENGTH
2218 036326  012704  037170          5$:     MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
2219 036332  004737  040266                  JSR     PC,T9REST               ;RESTORE PACKET TO STARTING VALUES
2220
2221 036336                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     036336  104404                                                                  TRAP    C$BSEG
2222
2223 036340  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2224 036344  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
```

```
036346  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
036350                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
036350  104455                                                                          TRAP    C$ERDF
036352  001625                                                                          .WORD   917
036354  003650                                                                          .WORD   SFIERR
036356  012034                                                                          .WORD   SFIMSG
036360  005037  002222          10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
036364  010337  037204                  MOV     R3,T9DATA+4             ;INSERT THE BAD MESSAGE LENGTH
036370  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
036374  004737  016250                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
036400  103405                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
036402  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
036404                                  ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
036404  104455                                                                          TRAP    C$ERDF
036406  001626                                                                          .WORD   918
036410  037707                                                                          .WORD   T9SSR
036412  012046                                                                          .WORD   PKTSSR
036414                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
036414  104406                                                                          TRAP    C$CLP1
036416                                  ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
036416  104410                                                                          TRAP    C$ESCAPE
036420  000056                                                                          .WORD   10000$-.
036422  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
036426  001004                          BNE     22$                     ;BRANCH IF YES
036430                                  ERRHRD  ERRNO,T9NINT,PKTSSR
036430  104456                                                                          TRAP    C$ERHRD
036432  001627                                                                          .WORD   919
036434  037776                                                                          .WORD   T9NINT
036436  012046                                                                          .WORD   PKTSSR
036440  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
036444  012702  102206                  MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
036450  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
036454  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
036456  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
036462  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
036464  001404                          BEQ     30$                     ;OKAY IF MATCH
036466                                  ERRHRD  ERRNO,T95REJ,PKTSSR     ;COMMAND NOT REJECTED
036466  104456                                                                          TRAP    C$ERHRD
036470  001630                                                                          .WORD   920
036472  037611                                                                          .WORD   T95REJ
036474  012046                                                                          .WORD   PKTSSR
036476                          30$:
036476                                  ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
036476                                                                                  10000$:
036476  104405                                                                          TRAP    C$ESEG

036500  005203                          INC     R3                      ;NEXT BUFFER LENGTH
036502  020327  000014                  CMP     R3,#14                  ;HAVE ALL BAD VALUES BEEN TESTED ?
036506  002002                          BGE     59$                     ;BRANCH IF ALL TESTED
036510  000137  036326                  JMP     5$                      ;BRANCH TILL BACK TO ZERO

036514                          59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
036514                                                                                  L10072:
036514  104403                                                                          TRAP    C$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.
                                        ;
                                        ;
036516                                  BGNSUB                          ;///////////// BEGIN SUBTEST ////////////
036516                                                                                          T9.6:
036516  104402                                                                                          TRAP    C$BSUB
036520  005737  002224                  TST     EXTFEA                  ;IS EXTENDED FEATURES SOFT. SW SET?
036524  001002                          BNE     4$                      ;BR, IF SOFTWARE SWITCH IS SET (ON)
036526  000137  036746                  JMP     55$                     ;EXIT SUBTEST
036532  004737  040266          4$:     JSR     PC,T9REST               ;SET PACKET TO START-UP VALUES

036536                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
036536  012700  000000                                                                          MOV     #PRI00,R0
036542  104441                                                                                  TRAP    C$SPRI
036544  012703  002762                  MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
036550  012704  037170                  MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
036554  012764  000012  000006          MOV     #10.,PKBCNT(R4)         ;START WITH EXTENDED FEATURES VALUE
036562                          5$:
036562                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
036562  104404                                                                                  TRAP    C$BSEG

036564  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
036570  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
036572  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
036574                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
036574  104455                                                                                  TRAP    C$ERDF
036576  001631                                                                                  .WORD   921
036600  003650                                                                                  .WORD   SFIERR
036602  012034                                                                                  .WORD   SFIMSG
036604  005037  002220          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
036610  005037  002222                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
036614  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
036620  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
036624  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
036626  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
036630                                  ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
036630  104455                                                                                  TRAP    C$ERDF
036632  001632                                                                                  .WORD   922
036634  037707                                                                                  .WORD   T9SSR
036636  012046                                                                                  .WORD   PKTSSR
036640  005237  002220                  INC     FATFLG                  ;SET FATAL ERROR FLAG
036644                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
036644  104406                                                                                  TRAP    C$CLP1
036646                                  ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
036646  104410                                                                                  TRAP    C$ESCAPE
036650  000056                                                                                  .WORD   10000$-.
036652  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
036656  001004                          BNE     22$                     ;BRANCH IF YES
036660                                  ERRHRD  ERRNO,T9NINT,PKTSSR
036660  104456                                                                                  TRAP    C$ERHRD
036662  001633                                                                                  .WORD   923
```

```
        036664  037776                                                                          .WORD   T9NINT
        036666  012046                                                                          .WORD   PKTSSR
  2319  036670  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
  2320  036674  012702  000200                  MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
  2321  036700  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
  2322  036704  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
  2323  036706  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
  2324  036712  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
  2325  036714  001404                          BEQ     30$                     ;OKAY IF MATCH
  2329  036716                                  ERRHRD  ERRNO,T9NBA,PKTSSR      ;NBA NOT ZERO
        036716  104456                                                                          TRAP    C$ERHRD
        036720  001634                                                                          .WORD   924
        036722  037246                                                                          .WORD   T9NBA
        036724  012046                                                                          .WORD   PKTSSR
  2330  036726                          30$:
  2331  036726                                  ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        036726                                                                          10000$:
        036726  104405                                                                          TRAP    C$ESEG
  2332
  2333  036730  012364  000006                  MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
  2334  036734  020327  003060                  CMP     R3,#TBLEND              ;HAS ALL DATA BEEN TESTED ?
  2335  036740  103002                          BHIS    55$                     ;BRANCH IF ALL DATA DONE
  2336  036742  000137  036562                  JMP     5$                      ;BRANCH TILL BACK TO ZERO
  2337
  2338  036746                          55$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        036746                                                                          L10073:
        036746  104403                                                                          TRAP    C$ESUB
  2339
  2340
  2341                                  ;+
  2342                                  ;
  2343                                  ;TEST 9, SUBTEST 7
  2344                                  ;
  2345                                  ;TEST WRITE CHARACTERISTICS WITH/WITHOUT INTERRUPTS ENABLED
  2346                                  ;
  2347                                  ;-
  2348
  2349  036750                                  BGNSUB                          ;//////////// BEGIN SUBTEST ////////////
        036750                                                                          T9.7:
        036750  104402                                                                          TRAP    C$BSUB
  2350
  2351  036752                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
        036752  012700  000000                                                                  MOV     #PRI00,R0
        036756  104441                                                                          TRAP    C$SPRI
  2352  036760  012704  037170                  MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
  2353  036764  004737  040266                  JSR     PC,T9REST               ;SET UP A VALID PACKET
  2354  036770  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
  2355  036774  103405                          BCS     10$                     ;BR IF SOFT INIT - OK
  2359  036776  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  2360  037000                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
        037000  104455                                                                          TRAP    C$ERDF
        037002  001635                                                                          .WORD   925
        037004  003650                                                                          .WORD   SFIERR
        037006  012034                                                                          .WORD   SFIMSG
  2361  037010  005037  002222          10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
  2362  037014  052714  000200                  BIS     #BIT7,(R4)              ;ENABLE INTERRUPTS
  2363  037020  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
```

```
037024  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
037030  103405                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
037032  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
037034                                  ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
037034  104455                                                                                  TRAP    C$ERDF
037036  001636                                                                                  .WORD   926
037040  037707                                                                                  .WORD   T9SSR
037042  012046                                                                                  .WORD   PKTSSR
037044                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
037044  104406                                                                                  TRAP    C$CLP1
037046                                  ESCAPE  SUB                     ;BY PASS SUBTEST IF FATAL ERROR
037046  104410                                                                                  TRAP    C$ESCAPE
037050  000102                                                                                  .WORD   L10074-.
037052  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
037056  001004                          BNE     22$                     ;BRANCH IF YES
037060                                  ERRHRD  ERRNO,T9NINT,PKTSSR
037060  104456                                                                                  TRAP    C$ERHRD
037062  001637                                                                                  .WORD   927
037064  037776                                                                                  .WORD   T9NINT
037066  012046                                                                                  .WORD   PKTSSR
037070                          22$:    CKLOOP                          ;LOOP ON ERROR ?
037070  104406                                                                                  TRAP    C$CLP1

037072  005037  002222                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
037076  042714  000200                  BIC     @BIT7,(R4)              ;DISABLE INTERRUPTS
037102  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
037106  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
037112  103405                          BCS     25$                     ;BR IF CARRY SET (GOOD RETURN)
037114  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
037116                                  ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
037116  104455                                                                                  TRAP    C$ERDF
037120  001640                                                                                  .WORD   928
037122  037707                                                                                  .WORD   T9SSR
037124  012046                                                                                  .WORD   PKTSSR
037126                          25$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
037126  104406                                                                                  TRAP    C$CLP1
037130                                  ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
037130  104410                                                                                  TRAP    C$ESCAPE
037132  000020                                                                                  .WORD   L10074 .
037134  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
037140  001404                          BEQ     30$                     ;BRANCH IF NOT
037142                                  ERRHRD  ERRNO,T9INT,PKTSSR
037142  104456                                                                                  TRAP    C$ERHRD
037144  001641                                                                                  .WORD   929
037146  040067                                                                                  .WORD   T9INT
037150  012046                                                                                  .WORD   PKTSSR
037152                          30$:
037152                                  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
037152                                                                                          L10074:
037152  104403                                                                                  TRAP    C$ESUB

037154                                  EXIT    TST                     ;ALL DONE THIS TEST
037154  104432                                                                                  TRAP    C$EXIT
037156  001162                                                                                  .WORD   L10065 .

                                ;+
                                ;LOCAL STORAGE FOR THIS TEST
```

```
                                        ;
                                                .BLKB   10-<. TSV2&7>
037160                                  T9PACKET:                               ;COMMAND PACKET FOR TEST
037170  100204                                  .WORD   100204                  ;WRITE CHAR COMMAND, WITH IE, ACK
037172  037200                                  .WORD   T9DATA                  ;ADDRESS OF CHARACTERISTICS BLOCK
037174  000000                                  .WORD   0
037176  000010                                  .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE

037200                                  T9DATA:                                 ;CHARACTERISTICS DATA BLOCK
037200  037212                                  .WORD   T9BFR                   ;ADDRESS OF MESSAGE BUFFER
037202  000000                                  .WORD   0
037204  000016                                  .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
037206  000000  000000                          .WORD   0,0

037212                                  T9BFR:  .BLKW   8.                      ;MESSAGE BUFFER

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

037232                                  T92DATA:
037232  000000  037140                          .WORD   0,BIT5!BIT6!BIT9!BIT10!BIT11!BIT12!BIT13
037236  000002  000001                          .WORD   2,BIT0
037242  000004  100100                          .WORD   4,BIT6!BIT15
        037246                          T92DONE=.


                                        ;+
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-

037246     127     122     111          T9NBA:  .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
037321     127     122     111          T92REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Non-Zero Unused Fields'
037420     127     122     111          T93REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Data Count'
037513     127     122     111          T94REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Block Address
037611     127     122     111          T95REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Buffer Length
037707     103     157     156          T9SSR:  .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
037776     105     170     160          T9NINT: .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
040067     125     156     145          T9INT:  .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
040156     111     156     143          T9TSBA: .ASCIZ  'Incorrect TSBA Address After WRITE CHARACTERISTICS'
040241     103     157     155          TST9ID: .ASCIZ  'Completion Interrupt'
                                                .EVEN


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

```
                                T9REST:
040266                                  SAVREG                          ;SAVE THE REGISTERS
040266
040272  012701  037170                  MOV     #T9PACKET,R1            ;START OF THE PACKET
040276  012721  100204                  MOV     #100204,(R1)+           ;WRITE CHARACTERISTICS WITH ACK, IE
040302  012721  037200                  MOV     #T9DATA,(R1)+           ;ADDRESS OF CHAR DATA BLOCK
040306  005021                          CLR     (R1)+                   ;EXTENDED ADDRESS
040310  012721  000010                  MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
040314  012721  037212                  MOV     #T9BFR,(R1)+            ;ADDRESS OF MESSAGE BUFFER
040320  005021                          CLR     (R1)+
040322  012721  000016                  MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
040326  005021                          CLR     (R1)+
040330  005011                          CLR     (R1)
040332  005037  037212                  CLR     T9BFR                   ;CLEAR 1ST LOC IN MESSAGE BUFFER
040336  000207                          RTS     PC                      ;RETURN
040340                                  ENDTST
040340                                                                          L10065:
040340  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
                                ;
                                ;-
040342                                  BGNTST
040342                                                                          T10::
040342  012700  043277                  MOV     #TST10ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
040346  004737  016510                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
040352  012737  000024  002214          MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
040360                          T10LOOP:

040360                                  BGNSUB                          ;//////////// BEGIN SUBTEST ////////////
040360                                                                          T10.1:
040360  104402                                                                           TRAP    C$BSUB

040362  004737  043326                  JSR     PC,T10RST               ;SET PACKET TO INITIAL VALUES
040366                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
040366  012700  000000                                                           MOV     #PRI00,R0
040372  104441                                                                   TRAP    C$SPRI
040374  012704  042470                  MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
040400  012764  000010  000006          MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
040406                          5$:
040406                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
040406  104404                                                                   TRAP    C$BSEG

040410  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
040414  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
```

```
040416  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040420                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
040420  104455                                                                          TRAP    C$ERDF
040422  001751                                                                          .WORD   1001
040424  003650                                                                          .WORD   SFIERR
040426  012034                                                                          .WORD   SFIMSG
040430  005037  002220          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
040434  005037  002222                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
040440  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
040444  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
040450  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
040452  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040454                                  ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
040454  104455                                                                          TRAP    C$ERDF
040456  001752                                                                          .WORD   1002
040460  043030                                                                          .WORD   T10SSR
040462  012046                                                                          .WORD   PKTSSR
040464  005237  002220                  INC     FATFLG                  ;SET FATAL ERROR FLAG
040470                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
040470  104406                                                                          TRAP    C$CLP1
040472                                  ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
040472  104410                                                                          TRAP    C$ESCAPE
040474  000056                                                                          .WORD   10000$-.
040476  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
040502  001004                          BNE     22$                     ;BRANCH IF YES
040504                                  ERRHRD  ERRNO,T10NINT,PKTSSR
040504  104456                                                                          TRAP    C$ERHRD
040506  001753                                                                          .WORD   1003
040510  043117                                                                          .WORD   T10NINT
040512  012046                                                                          .WORD   PKTSSR
040514  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
040520  012702  000200                  MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
040524  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
040530  001402                          BEQ     25$                     ;BRANCH IF NOT OFF LINE
040532  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
040536  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
040540  001404                          BEQ     30$                     ;OKAY IF MATCH
040542                                  ERRHRD  ERRNO,T10NBA,PKTSSR     ;NBA NOT ZERO
040542  104456                                                                          TRAP    C$ERHRD
040544  001754                                                                          .WORD   1004
040546  042671                                                                          .WORD   T10NBA
040550  012046                                                                          .WORD   PKTSSR
040552                          30$:
040552                                  ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
040552                                                                                  10000$:
040552  104405                                                                          TRAP    C$ESEG
040554                                  BGNSEG                          ;>>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
040554  104404                                                                          TRAP    C$BSEG

040556  005037  002222                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
040562  012737  025252  042512          MOV     #025252,T10BFR          ;WIPE OUT MESSAGE BUFFER AREA
040570  012714  100212                  MOV     #100212,(R4)            ;SET COMMAND PACKET TO MESS BUF REL
040574  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
040600  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
040604  103407                          BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
040606  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040610                                  ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
```

```
        040610  104455                                                                    TRAP    C$ERDF
        040612  001755                                                                    .WORD   1005
        040614  043030                                                                    .WORD   T10SSR
        040616  012046                                                                    .WORD   PKTSSR
  2568  040620  005237  002220                INC     FATFLG                  ;SET FATAL ERROR FLAG
  2569  040624                        45$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        040624  104406                                                                    TRAP    C$CLP1
  2570  040626  005737  002222                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
  2571  040632  001404                        BEQ     52$                     ;BRANCH IF NO
  2575  040634                                ERRHRD  ERRNO,T10INT,PKTSSR
        040634  104456                                                                    TRAP    C$ERHRD
        040636  001756                                                                    .WORD   1006
        040640  043210                                                                    .WORD   T10INT
        040642  012046                                                                    .WORD   PKTSSR
  2576  040644  016501  000002        52$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
  2577  040650  012702  000200                MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
  2578  040654  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
  2579  040660  001402                        BEQ     55$                     ;BRANCH IF NOT OFF-LINE
  2580  040662  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
  2581  040666  020201                55$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
  2582  040670  001404                        BEQ     60$                     ;OKAY IF MATCH
  2586  040672                                ERRHRD  ERRNO,T10NNBA,PKTSSR    ;NBA  SET
        040672  104456                                                                    TRAP    C$ERHRD
        040674  001757                                                                    .WORD   1007
        040676  042753                                                                    .WORD   T10NNBA
        040700  012046                                                                    .WORD   PKTSSR
  2587  040702                        60$:
  2588  040702  013701  042512                MOV     T10BFR,R1               ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
  2589  040706  012702  025252                MOV     #025252,R2              ;SET UP EXPECTED DATA
  2590  040712  020102                        CMP     R1,R2                   ;WAS ANY MESSAGE REC'D
  2591  040714  001404                        BEQ     70$                     ;BR, IF OK (EQUAL)
  2595  040716                                ERRHRD  ERRNO,T10MBF,EXPREC     ;MESSAGE BUFFER WAS MODIFIED
        040716  104456                                                                    TRAP    C$ERHRD
        040720  001760                                                                    .WORD   1008
        040722  042574                                                                    .WORD   T10MBF
        040724  015474                                                                    .WORD   EXPREC
  2596  040726                        70$:
  2597  040726  005737  002220                TST     FATFLG                  ;ANY FATAL ERRORS
  2598  040732  001403                        BEQ     80$                     ;BR, IF NO FATAL ERRORS
  2599  040734  004737  017202                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
  2600  040740                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<
        040740                                                                          10001$:
        040740  104405                                                                    TRAP    C$ESEG
  2601  040742                        80$:
  2602  040742                                ENDSUB                          ;///////////// END SUBTEST ///////////
        040742                                                                          L10076:
        040742  104403                                                                    TRAP    C$ESUB
  2603
  2604                                ;+
  2605                                ;TEST 10 SUBTEST 2
  2606                                ;
  2607                                ;CHECKS THAT THE MESSAGE BUFFER RELEASE COMMAND WORKS
  2608                                ;PROPERLY AND THAT THERE IS AN INTERRUPT IF THE "IE"
  2609                                ;BIT IS SET IN THE COMMAND PACKET AND THE "ERI" BIT
  2610                                ;IS SET IN THE CHARACTERISTICS DATA PACKET
  2611                                ;
  2612                                ;
```

```
040744                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
040744                                                                                  T10.2:
040744  104402                                                                                  TRAP    C$BSUB

040746  004737  043326                  JSR     PC,T10RST               ;SET PACKET TO INITIAL VALUES
040752                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
040752  012700  000000                                                                          MOV     #PRI00,R0
040756  104441                                                                                  TRAP    C$SPRI
040760  012704  042470                  MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
040764  012764  000010  000006          MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
040772                          5$:
040772                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
040772  104404                                                                                  TRAP    C$BSEG

040774  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
041000  103405                          BCS     10$                     ;BR IF SOFT INIT - OK
041002  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
041004                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
041004  104455                                                                                  TRAP    C$ERDF
041006  001761                                                                                  .WORD   1009
041010  003650                                                                                  .WORD   SFIERR
041012  012034                                                                                  .WORD   SFIMSG
041014  005037  002220          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
041020  005037  002222                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
041024  012737  000020  042506          MOV     #000020,T10DATA+6       ;SET ERI IN CHARACTERISTICS DATA
041032  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
041036  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
041042  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
041044  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
041046                                  ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
041046  104455                                                                                  TRAP    C$ERDF
041050  001762                                                                                  .WORD   1010
041052  043030                                                                                  .WORD   T10SSR
041054  012046                                                                                  .WORD   PKTSSR
041056  005237  002220                  INC     FATFLG                  ;SET FATAL ERROR FLAG
041062                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
041062  104406                                                                                  TRAP    C$CLP1
041064                                  ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
041064  104410                                                                                  TRAP    C$ESCAPE
041066  000056                                                                                  .WORD   10000$ .
041070  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
041074  001004                          BNE     22$                     ;BRANCH IF YES
041076                                  ERRHRD  ERRNO,T10NINT,PKTSSR
041076  104456                                                                                  TRAP    C$ERHRD
041100  001763                                                                                  .WORD   1011
041102  043117                                                                                  .WORD   T10NINT
041104  012046                                                                                  .WORD   PKTSSR
041106  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
041112  012702  000200                  MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
041116  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
041122  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
041124  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
041130  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
041132  001404                          BEQ     30$                     ;OKAY IF MATCH
041134                                  ERRHRD  ERRNO,T10NBA,PKTSSR     ;NBA NOT ZERO
041134  104456                                                                                  TRAP    C$ERHRD
041136  001764                                                                                  .WORD   1012
```

```
        041140  042671                                                                    .WORD   T10NBA
        041142  012046                                                                    .WORD   PKTSSR
   2660 041144                          30$:
   2661 041144                                  ENDSEG                          ;<<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<<
        041144                                                                                  100000$:
        041144  104405                                                                    TRAP    C$ESEG
   2662 041146                                  BGNSEG                          ;>>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>>
        041146  104404                                                                    TRAP    C$BSEG
   2663
   2664 041150  005037  002222                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
   2665 041154  012737  025252  042512          MOV     #025252,T10BFR          ;WIPE OUT MESSAGE BUFFER AREA
   2666 041162  012714  100212                  MOV     #100212,(R4)            ;SET COMMAND PACKET TO MESS BUF REL
   2667 041166  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
   2668 041172  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
   2669 041176  103407                          BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
   2670 041200  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
   2674 041202                                  ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        041202  104455                                                                    TRAP    C$ERDF
        041204  001765                                                                    .WORD   1013
        041206  043030                                                                    .WORD   T10SSR
        041210  012046                                                                    .WORD   PKTSSR
   2675 041212  005237  002220                  INC     FATFLG                  ;SET FATAL ERROR FLAG
   2676 041216                          45$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        041216  104406                                                                    TRAP    C$CLP1
   2677 041220  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
   2678 041224  001004                          BNE     52$                     ;BRANCH IF YES
   2682 041226                                  ERRHRD  ERRNO,T10INT,PKTSSR
        041226  104456                                                                    TRAP    C$ERHRD
        041230  001766                                                                    .WORD   1014
        041232  043210                                                                    .WORD   T10INT
        041234  012046                                                                    .WORD   PKTSSR
   2683 041236  016501  000002          52$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
   2684 041242  012702  000200                  MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
   2685 041246  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
   2686 041252  001402                          BEQ     55$                     ;BRANCH IF NOT OFF-LINE
   2687 041254  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
   2688 041260  020201                  55$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
   2689 041262  001404                          BEQ     60$                     ;OKAY IF MATCH
   2693 041264                                  ERRHRD  ERRNO,T10NNBA,PKTSSR    ;NBA NOT SET
        041264  104456                                                                    TRAP    C$ERHRD
        041266  001767                                                                    .WORD   1015
        041270  042753                                                                    .WORD   T10NNBA
        041272  012046                                                                    .WORD   PKTSSR
   2694 041274                          60$:
   2695 041274  013701  042512                  MOV     T10BFR,R1               ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
   2696 041300  012702  025252                  MOV     #025252,R2              ;SET UP EXPECTED DATA
   2697 041304  020102                          CMP     R1,R2                   ;WAS ANY MESSAGE REC'D
   2698 041306  001404                          BEQ     70$                     ;BR, IF OK (EQUAL)
   2702 041310                                  ERRHRD  ERRNO,T10MBF,EXPREC     ;MESSAGE BUFFER WAS MODIFIED
        041310  104456                                                                    TRAP    C$ERHRD
        041312  001770                                                                    .WORD   1016
        041314  042574                                                                    .WORD   T10MBF
        041316  015474                                                                    .WORD   EXPREC
   2703
   2704 041320                          70$:
   2705 041320  005737  002220                  TST     FATFLG                  ;ANY FATAL ERRORS
   2706 041324  001402                          BEQ     80$                     ;BR, IF NO FATAL ERRORS
```

```
041326  004737  017202                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
041332                          80$:
041332                                  ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
041332                                                                          10001$:
041332  104405                                                                          TRAP    C$ESEG
041334                                  ENDSUB                          ;//////////// END SUBTEST ////////////
041334                                                                          L10077:
041334  104403                                                                          TRAP    C$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
                                ;
                                ;
041336                                  BGNSUB                          ;//////////// BEGIN SUBTEST ////////////
041336                                                                          T10.3:
041336  104402                                                                          TRAP    C$BSUB

041340  004737  043326                  JSR     PC,T10RST               ;SET PACKET TO INITIAL VALUES
041344                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
041344  012700  000000                                                                  MOV     #PRI00,R0
041350  104441                                                                          TRAP    C$SPRI
041352  012704  042470                  MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
041356  012764  000010  000006          MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
041364                          5$:
041364                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
041364  104404                                                                          TRAP    C$BSEG

041366  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
041372  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
041374  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
041376                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
041376  104455                                                                          TRAP    C$ERDF
041400  001771                                                                          .WORD   1017
041402  003650                                                                          .WORD   SFIERR
041404  012034                                                                          .WORD   SFIMSG
041406  005037  002220          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
041412  005037  002222                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
041416  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
041422  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
041426  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
041430  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
041432                                  ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
041432  104455                                                                          TRAP    C$ERDF
041434  001772                                                                          .WORD   1018
041436  043030                                                                          .WORD   T10SSR
041440  012046                                                                          .WORD   PKTSSR
041442  005237  002220                  INC     FATFLG                  ;SET FATAL ERROR FLAG
041446                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
041446  104406                                                                          TRAP    C$CLP1
041450                                  ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
041450  104410                                                                          TRAP    C$ESCAPE
041452  000056                                                                          .WORD   10000$ .
041454  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
```

```
041460  001004                          BNE     22$                         ;BRANCH IF YES
041462                                  ERRHRD  ERRNO,T10NINT,PKTSSR
041462  104456                                                              TRAP    C$ERHRD
041464  001773                                                              .WORD   1019
041466  043117                                                              .WORD   T10NINT
041470  012046                                                              .WORD   PKTSSR
041472  016501  000002          22$:    MOV     TSSR(R5),R1                 ;GET THE CONTENTS OF TSSR
041476  012702  000200                  MOV     #SSR,R2                     ;EXPECTED CONTENTS OF TSSR
041502  032701  000100                  BIT     #OFL,R1                     ;IS OFF-LINE BIT SET ?
041506  001402                          BEQ     25$                         ;BRANCH IF NOT OFF-LINE
041510  052702  000100                  BIS     #OFL,R2                     ;SET OFF-LINE IN EXPECTED DATA
041514  020201                  25$:    CMP     R2,R1                       ;DOES EXPECTED MATCH RECEIVED ?
041516  001404                          BEQ     30$                         ;OKAY IF MATCH
041520                                  ERRHRD  ERRNO,T10NBA,PKTSSR         ;NBA NOT ZERO
041520  104456                                                              TRAP    C$ERHRD
041522  001774                                                              .WORD   1020
041524  042671                                                              .WORD   T10NBA
041526  012046                                                              .WORD   PKTSSR
041530                          30$:
041530                                  ENDSEG                              ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
041530                                                                              10000$:
041530  104405                                                              TRAP    C$ESEG
041532                                  BGNSEG                              ;>>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
041532  104404                                                              TRAP    C$BSEG
041534  004737  043326                  JSR     PC,T10RST                   ;RESET PACKETS AND COMMANDS
041540  005037  002222                  CLR     INTRECV                     ;CLEAR INTERRUPT RECEIVED FLAG
041544  012737  025252  042512          MOV     #025252,T10BFR              ;WIPE OUT MESSAGE BUFFER AREA
041552  012714  100212                  MOV     #100212,(R4)                ;SET COMMAND PACKET TO MESS BUF REL
041556  010465  000000                  MOV     R4,TSDB(R5)                 ;SET THE PACKET ADDRESS
041562  004737  016336                  JSR     PC,CHKTSSR                  ;WAIT FOR SSR TO SET
041566  103407                          BCS     45$                         ;BR IF CARRY SET (GOOD RETURN)
041570  010001                          MOV     R0,R1                       ;SAVE CONTENTS OF TSSR
041572                                  ERRDF   ERRNO,T10SSR,PKTSSR         ;DEVICE FATAL SSR FAILED TO SET
041572  104455                                                              TRAP    C$ERDF
041574  001775                                                              .WORD   1021
041576  043030                                                              .WORD   T10SSR
041600  012046                                                              .WORD   PKTSSR
041602  005237  002220                  INC     FATFLG                      ;SET FATAL ERROR FLAG
041606                          45$:    CKLOOP                              ;LOOP ON ERROR, IF FLAG SET
041606  104406                                                              TRAP    C$CLP1
041610  005737  002222                  TST     INTRECV                     ;DID AN INTERRUPT OCCUR ?
041614  001404                          BEQ     52$                         ;BRANCH IF NO
041616                                  ERRHRD  ERRNO,T10INT,PKTSSR
041616  104456                                                              TRAP    C$ERHRD
041620  001776                                                              .WORD   1022
041622  043210                                                              .WORD   T10INT
041624  012046                                                              .WORD   PKTSSR
041626  016501  000002          52$:    MOV     TSSR(R5),R1                 ;GET THE CONTENTS OF TSSR
041632  012702  000200                  MOV     #SSR,R2                     ;EXPECTED CONTENTS OF TSSR
041636  032701  000100                  BIT     #OFL,R1                     ;IS OFF-LINE BIT SET ?
041642  001402                          BEQ     55$                         ;BRANCH IF NOT OFF-LINE
041644  052702  000100                  BIS     #OFL,R2                     ;SET OFF-LINE IN EXPECTED DATA
041650  020201                  55$:    CMP     R2,R1                       ;DOES EXPECTED MATCH RECEIVED ?
041652  001404                          BEQ     60$                         ;OKAY IF MATCH
041654                                  ERRHRD  ERRNO,T10NNBA,PKTSSR        ;NBA NOT SET
041654  104456                                                              TRAP    C$ERHRD
041656  001777                                                              .WORD   1023
```

```
        041660  042753                                                                    .WORD   T10NNBA
        041662  012046                                                                    .WORD   PKTSSR
  2800  041664                          60$:
  2801  041664  013701  042512                MOV     T10BFR,R1               ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
  2802  041670  012702  025252                MOV     #025252,R2              ;SET UP EXPECTED DATA
  2803  041674  020102                        CMP     R1,R2                   ;WAS ANY MESSAGE REC'D
  2804  041676  001404                        BEQ     70$                     ;BR, IF OK (EQUAL)
  2808  041700                                ERRHRD  ERRNO,T10MBF,EXPREC     ;MESSAGE BUFFER WAS MODIFIED
        041700  104456                                                                    TRAP    C$ERHRD
        041702  002000                                                                    .WORD   1024
        041704  042574                                                                    .WORD   T10MBF
        041706  015474                                                                    .WORD   EXPREC
  2809
  2810  041710                          70$:  CKLOOP                          ;LOOP ON ERROR IF FLAG SET
        041710  104406                                                                    TRAP    C$CLP1
  2811
  2812  041712  005037  002222                CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
  2813  041716  004737  043326                JSR     PC,T10RST               ;RESET THE PACKETS AND COMMANDS
  2814  041722  042714  100000                BIC     #100000,(R4)            ;CLEAR THE ACK BIT
  2815  041726  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
  2816  041732  004737  016336                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
  2817  041736  103407                        BCS     75$                     ;BR IF CARRY SET (GOOD RETURN)
  2818  041740  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
  2822  041742                                ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        041742  104455                                                                    TRAP    C$ERDF
        041744  002001                                                                    .WORD   1025
        041746  043030                                                                    .WORD   T10SSR
        041750  012046                                                                    .WORD   PKTSSR
  2823  041752  005237  002220                INC     FATFLG                  ;SET FATAL ERROR FLAG
  2824  041756                          75$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        041756  104406                                                                    TRAP    C$CLP1
  2825  041760                                ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
        041760  104410                                                                    TRAP    C$ESCAPE
        041762  000062                                                                    .WORD   10001$-.
  2826  041764  005737  002222                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
  2827  041770  001006                        BNE     82$                     ;BRANCH IF YES
  2831  041772  016500  000002                MOV     TSSR(R5),R0             ;GET TSSR FOR ERROR REPORT
  2832  041776                                ERRHRD  ERRNO,T10NINT,PKTSSR
        041776  104456                                                                    TRAP    C$ERHRD
        042000  002002                                                                    .WORD   1026
        042002  043117                                                                    .WORD   T10NINT
        042004  012046                                                                    .WORD   PKTSSR
  2833  042006  016501  000002          82$:  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
  2834  042012  012702  000200                MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
  2835  042016  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
  2836  042022  001402                        BEQ     85$                     ;BRANCH IF NOT OFF-LINE
  2837  042024  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
  2838  042030  020201                  85$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
  2839  042032  001404                        BEQ     90$                     ;OKAY IF MATCH
  2843  042034                                ERRHRD  ERRNO,T10SSR,PKTSSR     ;NBA NOT ZERO
        042034  104456                                                                    TRAP    C$ERHRD
        042036  002003                                                                    .WORD   1027
        042040  043030                                                                    .WORD   T10SSR
        042042  012046                                                                    .WORD   PKTSSR
  2844  042044                          90$:
  2845  042044                                ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        042044                                                                            10001$:
```

```
      042044  104405                                                                    TRAP    C$ESEG
 2846 042046  005737  002220          TST     FATFLG                  ;ANY FATAL ERRORS
 2847 042052  001403                  BEQ     95$                     ;BR, IF NO FATAL ERRORS
 2848 042054  004737  017202          JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 2849
 2850 042060                          BGNSEG                          ;<<<<<<<<<<<<<< BGN SEGMENT <<<<<<<<<<<<<
      042060  104404                                                                    TRAP    C$BSEG
 2851 042062  005037  002222    95$:  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 2852 042066  004737  043326          JSR     PC,T10RST               ;RESET THE PACKETS AND COMMANDS
 2853 042072  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 2854 042076  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 2855 042102  103407                  BCS     100$                    ;BR IF CARRY SET (GOOD RETURN)
 2856 042104  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2860 042106                          ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      042106  104455                                                                    TRAP    C$ERDF
      042110  002004                                                                    .WORD   1028
      042112  043030                                                                    .WORD   T10SSR
      042114  012046                                                                    .WORD   PKTSSR
 2861 042116  005237  002220          INC     FATFLG                  ;SET FATAL ERROR FLAG
 2862 042122                    100$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      042122  104406                                                                    TRAP    C$CLP1
 2863 042124                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
      042124  104410                                                                    TRAP    C$ESCAPE
      042126  000062                                                                    .WORD   10002$-.
 2864 042130  005737  002222          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 2865 042134  001006                  BNE     112$                    ;BRANCH IF YES
 2869 042136  016500  000002          MOV     TSSR(R5),R0             ;GET TSSR FOR ERROR REPORT
 2870 042142                          ERRHRD  ERRNO,T10NINT,PKTSSR
      042142  104456                                                                    TRAP    C$ERHRD
      042144  002005                                                                    .WORD   1029
      042146  043117                                                                    .WORD   T10NINT
      042150  012046                                                                    .WORD   PKTSSR
 2871 042152  016501  000002   112$:  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 2872 042156  012702  000200          MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
 2873 042162  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 2874 042166  001402                  BEQ     125$                    ;BRANCH IF NOT OFF-LINE
 2875 042170  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 2876 042174  020201            125$: CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 2877 042176  001405                  BEQ     130$                    ;OKAY IF MATCH
 2881 042200                          ERRHRD  ERRNO,T10SSR,PKTSSR     ;NBA NOT ZERO
      042200  104456                                                                    TRAP    C$ERHRD
      042202  002006                                                                    .WORD   1030
      042204  043030                                                                    .WORD   T10SSR
      042206  012046                                                                    .WORD   PKTSSR
 2882 042210                          ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      042210                                                                      10002$:
      042210  104405                                                                    TRAP    C$ESEG
 2883 042212  005737  002220   130$:  TST     FATFLG                  ;CHECK FATAL FLAG FOR SET
 2884 042216  001402                  BEQ     140$                    ;BR, IF NOT SET (OK)
 2885 042220  004737  017202          JSR     PC,CKDROP               ;DROP DEVICE IF INDICATED
 2886 042224                    140$:
 2887 042224                          ENDSUB                          ;//////////// END SUBTEST ////////////
      042224                                                                      L10100:
      042224  104403                                                                    TRAP    C$ESUB
 2888 042226                    30$:
 2889
 2890                           ;*
```

```
                                        ;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
                                        ;
                                        ;

042226                                          BGNSUB                          ;//////////// BEGIN SUBTEST ////////////
042226                                                                                          T10.4:
042226  104402                                                                                          TRAP    C$BSUB

042230  004737  043400                          JSR     PC,T10RT2               ;SET SECOND PACKET UP
042234  004737  043326                          JSR     PC,T10RST               ;SET PACKET TO INITIAL VALUES
042240                                          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
042240  012700  000000                                                                          MOV     #PRI00,R0
042244  104441                                                                                  TRAP    C$SPRI
042246  012704  042470                          MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
042252  012703  042532                          MOV     #T10PKT,R3              ;GET THE ADDRESS OF 2ND CMD PACKET
042256  012764  000010  000006                  MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
042264  012763  000010  000006                  MOV     #8.,PKBCNT(R3)          ;START WITH MINIMUM ALLOWABLE VALUE
042272                                  5$:
042272                                          BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
042272  104404                                                                                  TRAP    C$BSEG
042274  004737  015774                          JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
042300  103405                                  BCS     10$                     ;BR IF SOFT INIT = OK
042302  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
042304                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
042304  104455                                                                                  TRAP    C$ERDF
042306  002007                                                                                  .WORD   1031
042310  003650                                                                                  .WORD   SFIERR
042312  012034                                                                                  .WORD   SFIMSG
042314  005037  002220                  10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
042320  005037  002222                          CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
042324  010465  000000                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
042330  010365  000000                          MOV     R3,TSDB(R5)             ;SECOND COMMAND PACKET
042334  004737  016250                          JSR     PC,WAITF                ;WAIT FOR SSR TO SET
042340  016501  000002                          MOV     TSSR(R5),R1             ;GET CONTENTS OF TSSR REGISTER
042344  032701  000200                          BIT     #SSR,R1                 ;CHECK FOR SSR (TSSR) SET
042350  001006                                  BNE     15$                     ;BR, IF SSR SET (COOD)
042352                                          ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
042352  104455                                                                                  TRAP    C$ERDF
042354  002010                                                                                  .WORD   1032
042356  043030                                                                                  .WORD   T10SSR
042360  012046                                                                                  .WORD   PKTSSR
042362  005237  002220                          INC     FATFLG                  ;SET FATAL ERROR FLAG
042366                                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
042366  104406                                                                                  TRAP    C$CLP1
042370                                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
042370  104410                                                                                  TRAP    C$ESCAPE
042372  000056                                                                                  .WORD   10000$ .
042374  005737  002222                          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
042400  001004                                  BNE     22$                     ;BRANCH IF YES


042402                                          ERRHRD  ERRNO,T10NINT,PKTSSR
042402  104456                                                                                  TRAP    C$ERHRD
```

```
        042404  002011                                                                    .WORD   1033
        042406  043117                                                                    .WORD   T10NINT
        042410  012046                                                                    .WORD   PKTSSR
   2940 042412  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
   2941 042416  012702  110200                  MOV     #SSR!RMR!SC,R2          ;EXPECTED CONTENTS OF TSSR
   2942 042422  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
   2943 042426  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
   2944 042430  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
   2945 042434  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
   2946 042436  001404                          BEQ     30$                     ;OKAY IF MATCH
   2950 042440                                  ERRHRD  ERRNO,T10SSR,PKTSSR     ;NBA NOT ZERO
        042440  104456                                                                    TRAP    C$ERHRD
        042442  002012                                                                    .WORD   1034
        042444  043030                                                                    .WORD   T10SSR
        042446  012046                                                                    .WORD   PKTSSR
   2951 042450                          30$:
   2952 042450                                  ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        042450                                                                          10000$:
        042450  104405                                                                    TRAP    C$ESEG
   2953 042452                                  ENDSUB                          ;///////////// END SUBTEST /////////////
        042452                                                                          L10101:
        042452  104403                                                                    TRAP    C$ESUB
   2954 042454                                  EXIT    TST                     ;ALL DONE WITH THIS TEST
        042454  104432                                                                    TRAP    C$EXIT
        042456  000774                                                                    .WORD   L10075 .
   2955
   2956                                 ;+
   2957                                 ;LOCAL STORAGE FOR THIS TEST
   2958                                 ;
   2959
   2961 042460                                  .BLKB   10-<.-TSV2&7>
   2963 042470                          T10PACKET:                              ;COMMAND PACKET FOR TEST
   2964 042470  100204                          .WORD   100204                  ;WRITE CHAR COMMAND, WITH IE, ACK
   2965 042472  042500                          .WORD   T10DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
   2966 042474  000000                          .WORD   0
   2967 042476  000010                          .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
   2968
   2969 042500                          T10DATA:                                        ;CHARACTERISTICS DATA BLOCK
   2970 042500  042512                          .WORD   T10BFR                  ;ADDRESS OF MESSAGE BUFFER
   2971 042502  000000                          .WORD   0
   2972 042504  000016                          .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
   2973 042506  000000  000000                  .WORD   0,0
   2974
   2975 042512                          T10BFR: .BLKW   8.                      ;MESSAGE BUFFER
   2976
   2977                                 ;+
   2978                                 ;
   2979                                 ;TEST DATA FOR SUBTEST FOUR
   2980                                 ;
   2981 042532                          T10PKT:                                 ;COMMAND PACKET FOR TEST
   2982 042532  100204                          .WORD   100204                  ;WRITE CHAR COMMAND, WITH IE, ACK
   2983 042534  042542                          .WORD   T10DTA                  ;ADDRESS OF CHARACTERISTICS BLOCK
   2984 042536  000000                          .WORD   0
   2985 042540  000010                          .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
   2986
   2987 042542                          T10DTA:                                 ;CHARACTERISTICS DATA BLOCK
   2988 042542  042554                          .WORD   T10BUFR                 ;ADDRESS OF MESSAGE BUFFER
```

```
042544  000000                          .WORD   0
042546  000016                          .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
042550  000000  000000                  .WORD   0,0

042554                          T10BUFR:.BLKW   8.                      ;MESSAGE BUFFER

                                ;+
                                ;LOCAL TEXT MESSAGES FOR TEST
                                ;


042574     115     145     163  T10MBF: .ASCIZ  'Message Buffer Modified after MESSAGE BUFFER RELEASE Command
042671     116     102     101  T10NBA: .ASCIZ  'NBA Not Clear After WRITE CHARACTERISTICS Command'
042753     116     102     101  T10NNBA:        .ASCIZ  'NBA Set After MESSAGE BUFFER RELEASE Command
043030     103     157     156  T10SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS
043117     105     170     160  T10NINT:        .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS
043210     125     156     145  T10INT: .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
043277     102     141     163  TST10ID:        .ASCIZ  'Basic Packet Protocol'
                                        .EVEN



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

043326                          T10RST:
043326                                  SAVREG                          ;SAVE THE REGISTERS
043332  012701  042470                  MOV     #T10PACKET,R1           ;START OF THE PACKET
043336  012721  100204                  MOV     #100204,(R1)+           ;WRITE CHARACTERISTICS WITH ACK, IE
043342  012721  042500                  MOV     #T10DATA,(R1)+          ;ADDRESS OF CHAR DATA BLOCK
043346  005021                          CLR     (R1)+                   ;EXTENDED ADDRESS
043350  012721  000010                  MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
043354  012721  042512                  MOV     #T10BFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
043360  005021                          CLR     (R1)+
043362  012721  000016                  MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
043366  005021                          CLR     (R1)+
043370  005011                          CLR     (R1)
043372  005037  042512                  CLR     T10BFR                  ;CLEAR 1ST LOC IN MESSAGE BUFFER
043376  000207                          RTS     PC                      ;RETURN
                                ;+
                                ;
                                ;ROUTINE TO RESTORE COMMAND PACKET #2 TO START-UP (DEFAULT) VALUES
                                ;
                                ;

043400                          T10RT2:
043400                                  SAVREG                          ;SAVE THE REGISTERS
043404  012701  042532                  MOV     #T10PKT,R1              ;START OF THE PACKET
043410  012721  100204                  MOV     #100204,(R1)+           ;WRITE CHARACTERISTICS WITH ACK, IE
043414  012721  042542                  MOV     #T10DTA,(R1)+           ;ADDRESS OF CHAR DATA BLOCK
043420  005021                          CLR     (R1)+                   ;EXTENDED ADDRESS
043422  012721  000010                  MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
043426  012721  042554                  MOV     #T10BUFR,(R1)+          ;ADDRESS OF MESSAGE BUFFER
043432  005021                          CLR     (R1)+
```

```
3046 043434  012721  000016                MOV     #14.,(R1)+      ;LENGTH OF MESSAGE BUFFER
3047 043440  005021                        CLR     (R1)+
3048 043442  005011                        CLR     (R1)
3049 043444  005037  042554                CLR     T10BUFR         ;CLEAR 1ST LOC IN MESSAGE BUFFER
3050 043450  000207                        RTS     PC              ;RETURN
3051 043452                                ENDTST
     043452                                                                     L10075:
     043452  104401                                                                     TRAP    C$ETST
3052
3053                                       .SBTTL  TEST  11: NON-TAPE MOTION COMMANDS
3054
3055                                ;+
3056                                ;
3057                                ;THIS TEST VERIFIES PROPER OPERATION OF THE INITIALIZE
3058                                ;COMMAND. TWO SUBTESTS ARE USED. THE FIRST VERIFIES THAT
3059                                ;THE COMMAND RUNS TO COMPLETION AND STORES A VALID
3060                                ;MESSAGE PACKET. THE SECOND VERIFIES THAT NON-ZERO
3061                                ;VALUES IN THE COMMAND MODE FIELD CAUSES COMMAND REJECT.
3062                                ;
3063                                ;
3064
3065 043454                                BGNTST
     043454                                                                     T11::
3070 043454  012700  045462                MOV     #TST11ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
3071 043460  004737  016510                JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
3072 043464  012737  000024  002214        MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
3073 043472                         T11LOOP:
3074 043472                                BGNSUB                          ;//////////// BEGIN SUBTEST ////////////
     043472                                                                     T11.1:
     043472  104402                                                                     TRAP    C$BSUB
3075
3076 043474                                SETPRT  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     043474  012700  000000                                                             MOV     #PRI00,R0
     043500  104441                                                                     TRAP    C$SPPI
3077 043502  004737  015774                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
3078 043506  103405                        BCS     3$                      ;BR IF SOFT INIT = OK
3082 043510  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3083 043512                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     043512  104455                                                                     TRAP    C$ERDF
     043514  002115                                                                     .WORD   1101
     043516  003650                                                                     .WORD   SFIERR
     043520  012034                                                                     .WORD   SFIMSG
3084 043522                         3$:
3085 043522  012704  044710                MOV     #T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
3086 043526  004737  010662                JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
3087 043532  103404                        BCS     4$                      ;BR, IF COMMAND ISSUED OK
3091 043534                                ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     043534  104456                                                                     TRAP    C$ERHRD
     043536  002116                                                                     .WORD   1102
     043540  005054                                                                     .WORD   WPTMSG
     043542  012034                                                                     .WORD   SFIMSG
3092 043544                         4$:
3093 043544  004737  045514                JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
3094 043550  012704  044640                MOV     #T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3095 043554                         5$:
3096 043554                                BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     043554  104404                                                                     TRAP    C$BSEG
```

```
3097
3098 043556  005037  002220          10$:   CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
3099 043562  005037  002222                 CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
3100 043566  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
3101 043572  004737  016336                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3102 043576  103407                         BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
3103 043600  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3107 043602                                 ERRDF   ERRNO,T11SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     043602  104455                                                                                TRAP    C$ERDF
     043604  002117                                                                                .WORD   1103
     043606  045204                                                                                .WORD   T11SSR
     043610  012046                                                                                .WORD   PKTSSR
3108 043612  005237  002220                 INC     FATFLG                  ;SET FATAL ERROR FLAG
3109 043616                          15$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043616  104406                                                                                TRAP    C$CLP1
3110 043620                                 ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     043620  104410                                                                                TRAP    C$ESCAPE
     043622  000074                                                                                .WORD   10000$ .
3111 043624  005737  002222                  ST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
3112 043630  001004                         BNE     22$                     ;BRANCH IF YES
3116 043632                                 ERRHRD  ERRNO,T11NINT,PKTSSR
     043632  104456                                                                                TRAP    C$ERHRD
     043634  002120                                                                                .WORD   1104
     043636  045334                                                                                .WORD   T11NINT
     043640  012046                                                                                .WORD   PKTSSR
3117 043642  016501  000002          22$:   MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
3118 043646  012702  000200                 MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
3119 043652  032701  000100                 BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
3120 043656  001402                         BEQ     25$                     ;BRANCH IF NOT OFF-LINE
3121 043660  052702  000100                 BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
3122 043664  020201                  25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
3123 043666  001404                         BEQ     30$                     ;OKAY IF MATCH
3127 043670                                 ERRHRD  ERRNO,T11NBA,PKTSSR     ;NBA NOT ZERO
     043670  104456                                                                                TRAP    C$ERHRD
     043672  002121                                                                                .WORD   1105
     043674  044752                                                                                .WORD   T11NBA
     043676  012046                                                                                .WORD   PKTSSR
3128 043700                          30$:
3129 043700  004737  011114          35$:   JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
3130 043704  103405                         BCS     59$                     ;RAM OK GO ON
3134 043706                                 ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
     043706  104456                                                                                TRAP    C$ERHRD
     043710  002122                                                                                .WORD   1106
     043712  004743                                                                                .WORD   PKTRAM
     043714  015510                                                                                .WORD   RAMERR
3135 043716                                 ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     043716                                                                          10000$:
     043716  104405                                                                                TRAP    C$ESEG
3136
3137
3138 043720                          59$:   ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\
     043720                                                                          L10103:
     043720  104403                                                                                TRAP    C$ESUB
3139
3140 043722  005737  002220                 TST     FATFLG                  ;ANY FATAL ERRORS ?
3141 043726  001402                         BEQ     60$                     ;BRANCH IF NOT
3142 043730  004737  017202                 JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
```

```
3143 043734                          60$:
3144
3145                                 ;+
3146                                 ;
3147                                 ;TEST 11, SUBTEST 2
3148                                 ;
3149                                 ;CHECK THAT NON-ZERO MODE BITS BEING SET CAUSES
3150                                 ;INITIALIZE COMMAND TO BE REJECTED
3151                                 ;
3152                                 ;-
3153
3154 043734                                  BGNSUB                          ;//////////// BEGIN SUBTEST ////////////
     043734                                                                          T11.2:
     043734  104402                                                                          TRAP    C$BSUB
3155
3156 043736                                  SETPRI  #PRIOO                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     043736  012700  000000                                                                  MOV     #PRIOO,RO
     043742  104441                                                                          TRAP    C$SPRI
3157 043744                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     043744  104404                                                                          TRAP    C$BSEG
3158
3159
3160 043746  004737  015774                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
3161 043752  103405                          BCS     3$                      ;BR IF SOFT INIT = OK
3165 043754  010001                          MOV     RO,R1                   ;SAVE CONTENTS OF TSSR
3166 043756                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     043756  104455                                                                          TRAP    C$ERDF
     043760  002123                                                                          .WORD   1107
     043762  003650                                                                          .WORD   SFIERR
     043764  012034                                                                          .WORD   SFIMSG
3167 043766                          3$:
3168 043766  012704  044710                  MOV     #T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
3169 043772  004737  010662                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
3170 043776  103404                          BCS     4$                      ;BR, IF COMMAND ISSUED OK
3174 044000                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     044000  104456                                                                          TRAP    C$ERHRD
     044002  002124                                                                          .WORD   1108
     044004  005054                                                                          .WORD   WRTMSG
     044006  012034                                                                          .WORD   SFIMSG
3175 044010                          4$:
3176 044010  004737  045514                  JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
3177 044014  012704  044640                  MOV     #T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3178 044020                          5$:
3179 044020  005037  002222          10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
3180 044024  052714  007400                  BIS     #P.MODE,(R4)            ;NON-ZERO COMMAND MODE BITS
3181 044030  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
3182 044034  004737  016336                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3183 044040  103405                          BCS     15$                     ;BR IF CARRY SET (COOD RETURN)
3184 044042  010001                          MOV     RO,R1                   ;SAVE CONTENTS OF TSSR
3188 044044                                  ERRDF   ERRNO,T11SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     044044  104455                                                                          TRAP    C$ERDF
     044046  002125                                                                          .WORD   1109
     044050  045204                                                                          .WORD   T11SSR
     044052  012046                                                                          .WORD   PKTSSR
3189 044054                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044054  104406                                                                          TRAP    C$CLP1
3190 044056                                  ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
```

```
        044056  104410                                                                  TRAP    C$ESCAPE
        044060  000074                                                                  .WORD   10000$ .
   3191 044062  005737  002222                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
   3192 044066  001004                        BNE     22$                     ;BRANCH IF YES
   3196 044070                                ERRHRD  ERRNO,T11NINT,PKTSSR
        044070  104456                                                                  TRAP    C$ERHRD
        044072  002126                                                                  .WORD   1110
        044074  045334                                                                  .WORD   T11NINT
        044076  012046                                                                  .WORD   PKTSSR
   3197 044100  016501  000002          22$:  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
   3198 044104  012702  100206                MOV     #SC!SSR!TSREJ,R2        ;EXPECTED CONTENTS OF TSSR
   3199 044110  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
   3200 044114  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
   3201 044116  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
   3202 044122  020201                  25$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
   3203 044124  001404                        BEQ     30$                     ;OKAY IF MATCH
   3207 044126                                ERRHRD  ERRNO,T112REJ,PKTSSR    ;COMMAND NOT REJECTED
        044126  104456                                                                  TRAP    C$ERHRD
        044130  002127                                                                  .WORD   1111
        044132  045012                                                                  .WORD   T112REJ
        044134  012046                                                                  .WORD   PKTSSR
   3208 044136                          30$:
   3209 044136  004737  011114          35$:  JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
   3210 044142  103405                        BCS     59$                     ;RAM OK GO ON
   3214 044144                                ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
        044144  104456                                                                  TRAP    C$ERHRD
        044146  002130                                                                  .WORD   1112
        044150  004743                                                                  .WORD   PKTRAM
        044152  015510                                                                  .WORD   RAMERR
   3215 044154                                ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        044154                                                                  10000$:
        044154  104405                                                                  TRAP    C$ESEG
   3216
   3217 044156                          59$:  ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
        044156                                                                  L10104:
        044156  104403                                                                  TRAP    C$ESUB
   3218
   3219                                 ;+
   3220                                 ;
   3221                                 ;TEST 11, SUBTEST 3
   3222                                 ;
   3223                                 ;CHECK THAT THE GET STATUS COMMAND IS ACCEPTED
   3224                                 ;
   3225                                 ;-
   3226
   3227 044160                                BGNSUB                          ;//////////// BEGIN SUBTEST ////////////
        044160                                                                  T11.3.
        044160  104402                                                                  TRAP    C$BSUB
   3228
   3229 044162                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
        044162  012700  000000                                                          MOV     #PRI00,R0
        044166  104441                                                                  TRAP    C$SPRI
   3230 044170                                BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
        044170  104404                                                                  TRAP    C$BSEG
   3231 044172  004737  015774                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
   3232 044176  103405                        BCS     3$                      ;BR IF SOFT INIT = OK
   3236 044200  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
```

```
3237 044202                          ERROF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     044202  104455                                                                       TRAP    C$EROF
     044204  002131                                                                       .WORD   1113
     044206  003650                                                                       .WORD   SFIERR
     044210  012034                                                                       .WORD   SFIMSG
3238 044212                     3$:
3239 044212  012704  044710          MOV     #T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
3240 044216  004737  010662          JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
3241 044222  103404                  BCS     4$                      ;BR. IF COMMAND ISSUED OK
3245 044224                          ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     044224  104456                                                                       TRAP    C$ERHRD
     044226  002132                                                                       .WORD   1114
     044230  005054                                                                       .WORD   WRTMSG
     044232  012034                                                                       .WORD   SFIMSG
3246 044234                     4$:
3247 044234  004737  045514          JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
3248 044240  012704  044640          MOV     #T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3249 044244                     5$:
3250 044244  005037  002222     10$: CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
3251 044250  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
3252 044254  004737  016336          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3253 044260  103405                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
3254 044262  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3258 044264                          ERROF   ERRNO,T11SR2,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     044264  104455                                                                       TRAP    C$EROF
     044266  002133                                                                       .WORD   1115
     044270  045260                                                                       .WORD   T11SR2
     044272  012046                                                                       .WORD   PKTSSR
3259 044274                     15$: CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044274  104406                                                                       TRAP    C$CLP1
3260 044276                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     044276  104410                                                                       TRAP    C$ESCAPE
     044300  000074                                                                       .WORD   10000$
3261 044302  005737  002222          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
3262 044306  001004                  BNE     22$                     ;BRANCH IF YES
3266 044310                          ERRHRD  ERRNO,T11NINT,PKTSSR
     044310  104456                                                                       TRAP    C$ERHRD
     044312  002134                                                                       .WORD   1116
     044314  045334                                                                       .WORD   T11NINT
     044316  012046                                                                       .WORD   PKTSSR
3267 044320  016501  000002     22$: MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
3268 044324  012702  000200          MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
3269 044330  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
3270 044334  001402                  BEQ     25$                     ;BRANCH IF NOT OFF-LINE
3271 044336  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
3272 044342  020201             25$: CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
3273 044344  001404                  BEQ     30$                     ;OKAY IF MATCH
3277 044346                          ERRHRD  ERRNO,T113REJ,PKTSSR    ;COMMAND NOT ACCEPTED
     044346  104456                                                                       TRAP    C$ERHRD
     044350  002135                                                                       .WORD   1117
     044352  045073                                                                       .WORD   T113REJ
     044354  012046                                                                       .WORD   PKTSSR
3278 044356                     30$:
3279 044356  004737  011114     35$: JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
3280 044362  103405                  BCS     59$                     ;RAM OK GO ON
3284 044364                          ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
     044364  104456                                                                       TRAP    C$ERHRD
```

```
        044366  002136                                                                    .WORD   1118
        044370  004743                                                                    .WORD   PKTRAM
        044372  015510                                                                    .WORD   RAMERR
   3285 044374                                ENDSEG                  ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        044374                                                                    100001:
        044374  104405                                                                    TRAP    C$ESEG
   3286
   3287
   3288 044376                        59$:    ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        044376                                                                    L10105:
        044376  104403                                                                    TRAP    C$ESUB
   3289
   3290                                ;+
   3291                                ;
   3292                                ;TEST 11, SUBTEST 4
   3293                                ;
   3294                                ;SUBTEST TO VERIFY THAT A GET STATUS COMMAND IS
   3295                                ;REJECTED IF A NON-ZERO COMMAND MODE FIELD IS USED
   3296                                ;
   3297                                ;-
   3298
   3299 044400                                BGNSUB                  ;///////////// BEGIN SUBTEST ///////////
        044400                                                                    T11.4:
        044400  104402                                                                    TRAP    C$BSUB
   3300
   3301 044402                                SETPRI  #PRI00          ;LOWER PRIORITY TO ALLOW INTERRUPTS
        044402  012700  000000                                                            MOV     #PRI00,R0
        044406  104441                                                                    TRAP    C$SPRI
   3302 044410                                BGNSEG                  ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
        044410  104404                                                                    TRAP    C$BSEG
   3303 044412  004737  015774                JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
   3304 044416  103405                        BCS     3$              ;BR IF SOFT INIT - OK
   3308 044420  010001                        MOV     R0,R1           ;SAVE CONTENTS OF TSSR
   3309 044422                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
        044422  104455                                                                    TRAP    C$ERDF
        044424  002137                                                                    .WORD   1119
        044426  003650                                                                    .WORD   SFIERR
        044430  012034                                                                    .WORD   SFIMSG
   3310 044432                        3$:
   3311 044432  012704  044710                MOV     #T11PK2,R4      ;WRITE CHARACTERISTICS PACKET
   3312 044436  004737  010662                JSR     PC,WRTCHR       ;ISSUE WRITE CHARACTERISTICS
   3313 044442  103404                        BCS     4$              ;BR, IF COMMAND ISSUED OK
   3317 044444                                ERRWRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
        044444  104456                                                                    TRAP    C$ERWRD
        044446  002140                                                                    .WORD   1120
        044450  005054                                                                    .WORD   WRTMSG
        044452  012034                                                                    .WORD   SFIMSG
   3318 044454                        4$:
   3319 044454  004737  045514                JSR     PC,T11REST      ;SET UP PACKET FOR COMMAND
   3320 044460  012704  044640                MOV     #T11PACKET,R4   ;GET THE ADDRESS OF COMMAND PACKET
   3321 044464                        5$:
   3322 044464  005037  002222        10$:    CLR     INTRCV          ;CLEAR INTERRUPT RECEIVED FLAG
   3323 044470  052714  007000                BIS     #007000,(R4)    ;SET TO NON-ZERO MODE
   3324 044474  010465  000000                MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
   3325 044500  004737  016336                JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
   3326 044504  103405                        BCS     15$             ;BR IF CARRY SET (GOOD RETURN)
   3327 044506  010001                        MOV     R0,R1           ;SAVE CONTENTS OF TSSR
```

```
3331 044510                                ERRDF   ERRNO,T11SR2,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     044510  104455                                                                          TRAP    C$ERDF
     044512  002141                                                                          .WORD   1121
     044514  045260                                                                          .WORD   T11SR2
     044516  012046                                                                          .WORD   PKTSSR
3332 044520                        15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044520  104406                                                                          TRAP    C$CLP1
3333 044522                                ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
     044522  104410                                                                          TRAP    C$ESCAPE
     044524  000076                                                                          .WORD   L10106 .
3334 044526  005737  002222                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
3335 044532  001004                        BNE     22$                     ;BRANCH IF YES
3339 044534                                ERRHRD  ERRNO,T11NINT,PKTSSR
     044534  104456                                                                          TRAP    C$ERHRD
     044536  002142                                                                          .WORD   1122
     044540  045334                                                                          .WORD   T11NINT
     044542  012046                                                                          .WORD   PKTSSR
3340 044544  016501  000002        22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
3341 044550  012702  100206                MOV     #SC!SSR!TSREJ,R2        ;EXPECTED CONTENTS OF TSSR
3342 044554  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
3343 044560  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
3344 044562  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
3345 044566  020201        25$:            CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
3346 044570  001404                        BEQ     30$                     ;OKAY IF MATCH
3350 044572                                ERRHRD  ERRNO,T114REJ,PKTSSR    ;COMMAND NOT REJECTED
     044572  104456                                                                          TRAP    C$ERHRD
     044574  002143                                                                          .WORD   1123
     044576  045123                                                                          .WORD   T114REJ
     044600  012046                                                                          .WORD   PKTSSR
3351 044602                        30$:
3352
3353 044602  004737  011114        35$:    JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
3354 044606  103404                        BCS     59$                     ;RAM OK GO ON
3358 044610                                ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
     044610  104456                                                                          TRAP    C$ERHRD
     044612  002144                                                                          .WORD   1124
     044614  004743                                                                          .WORD   PKTRAM
     044616  015510                                                                          .WORD   RAMERR
3359 044620                        59$:
3360 044620                                ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<
     044620                                                                         10000$:
     044620  104405                                                                          TRAP    C$ESEG
3361 044622                                ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\
     044622                                                                         L10106:
     044622  104403                                                                          TRAP    C$ESUB
3362
3363 044624                                EXIT    TST                     ;ALL DONE THIS TEST
     044624  104432                                                                          TRAP    C$EXIT
     044626  000772                                                                          .WORD   L10102-.
3364
3365                               ;+
3366                               ;LOCAL STORAGE FOR THIS TEST
3367                               ;-
3368
3370 044630                                .BLKB   10-<.-TSV267>
3372 044640                        T11PACKET:                              ;COMMAND PACKET FOR TEST
3373 044640  100204                        .WORD   100204                  ;WRITE CHAR COMMAND, WITH IE, ACK
```

```
044642  044650                          .WORD   T11DATA         ;ADDRESS OF CHARACTERISTICS BLOCK
044644  000000                          .WORD   0
044646  000010                          .WORD   8.              ;STARTING VALUE OF BLOCK SIZE

044650                          T11DATA:                                ;CHARACTERISTICS DATA BLOCK
044650  044662                          .WORD   T11BFR          ;ADDRESS OF MESSAGE BUFFER
044652  000000                          .WORD   0
044654  000016                          .WORD   14.             ;LENGTH OF MESSAGE BUFFER
044656  000000  000000                  .WORD   0,0

044662                          T11BFR: .BLKW   8.              ;MESSAGE BUFFER


044702                                  .BLKB   10 <.-TSV267>
044710                          T11PK2:                         ;COMMAND PACKET FOR TEST
044710  100204                          .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
044712  044720                          .WORD   T11DTA          ;ADDRESS OF CHARACTERISTICS BLOCK
044714  000000                          .WORD   0
044716  000010                          .WORD   8.              ;STARTING VALUE OF BLOCK SIZE

044720                          T11DTA:                         ;CHARACTERISTICS DATA BLOCK
044720  044732                          .WORD   T11BF2          ;ADDRESS OF MESSAGE BUFFER
044722  000000                          .WORD   0
044724  000016                          .WORD   14.             ;LENGTH OF MESSAGE BUFFER
044726  000000  000000                  .WORD   0,0

044732                          T11BF2: .BLKW   8.              ;MESSAGE BUFFER


                                ;+
                                ;LOCAL TEXT MESSAGES FOR TEST
                                ;-

044752     111     116     111  T11NBA: .ASCIZ  'INITIALIZE Command Not Accepted'
045012     111     116     111  T112REJ:        .ASCIZ  'INITIALIZE Not Rejected With Non Zero Mode Field'
045073     107     105     124  T113REJ:        .ASCIZ  'GET STATUS Not Accepted'
045123     107     105     124  T114REJ:        .ASCIZ  'GET STATUS Not Rejected With Non-Zero Mode Field'
045204     103     157     156  T11SSR: .ASCIZ  'Contents of TSSR Incorrect After INITIALIZE'
045260     103     157     156  T11SR2: .ASCIZ  'Contents of TSSR Incorrect After GET STATUS'
045334     105     170     160  T11NINT:        .ASCIZ  'Expected Interrupt Not Received On INITIALIZE
045412     111     156     143  T11TSBA:        .ASCIZ  'Incorrect TSBA Address After INITIALIZE'
045462     116     157     156  TST11ID:        .ASCIZ  'Non-Tape Motion Commands'
                                        .EVEN


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

045514                          T11REST:
045514                                  SAVREG                  ;SAVE THE REGISTERS
045520  012701  044640                  MOV     #T11PACKET,R1   ;START OF THE PACKET
045524  012721  100213                  MOV     #100213,(R1)+   ;INITIALIZE WITH ACK, IE
```

```
045530  005021                          CLR     (R1)+           ;ADDRESS OF CHAR DATA BLOCK
045532  005021                          CLR     (R1)+           ;EXTENDED ADDRESS
045534  005021                          CLR     (R1)+           ;SIZE OF DATA BLOCK IN BYTES
045536  005021                          CLR     (R1)+           ;ADDRESS OF MESSAGE BUFFER
045540  005021                          CLR     (R1)+
045542  005021                          CLR     (R1)+           ;LENGTH OF MESSAGE BUFFER
045544  005021                          CLR     (R1)+
045546  005011                          CLR     (R1)
045550  005037  044662                  CLR     T11BFR          ;CLEAR 1ST LOC IN MESSAGE BUFFER
045554  000207                          RTS     PC              ;RETURN
                                ;+
                                ;
                                ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
                                ;GET STATUS COMMAND
                                ;
                                ;-

045556                          T11RT2:
045556                                  SAVREG                  ;SAVE THE REGISTERS
045562  012701  044640                  MOV     #T11PACKET,R1   ;START OF THE PACKET
045566  012721  100217                  MOV     #100217,(R1)+   ;GET STATUS WITH ACK, IE
045572  005021                          CLR     (R1)+           ;ADDRESS OF CHAR DATA BLOCK
045574  005021                          CLR     (R1)+           ;EXTENDED ADDRESS
045576  005021                          CLR     (R1)+           ;SIZE OF DATA BLOCK IN BYTES
045600  005021                          CLR     (R1)+           ;ADDRESS OF MESSAGE BUFFER
045602  005021                          CLR     (R1)+
045604  005021                          CLR     (R1)+           ;LENGTH OF MESSAGE BUFFER
045606  005021                          CLR     (R1)+
045610  005011                          CLR     (R1)
045612  005037  044662                  CLR     T11BFR          ;CLEAR 1ST LOC IN MESSAGE BUFFER
045616  000207                          RTS     PC              ;RETURN
045620                                  ENDTST
045620                                                                  L10102:
045620  104401                                                                  TRAP    C$ETST
045622                                  ENDMOD
```

```
                                        .TITLE  TSV6 - PARAMETER CODING
                                        BGNMOD  TSV6
045622                          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.
                                ;--
045622                                  BGNHRD
045622  000010                          .WORD L10107-L$HARD/2
045624                          L$HARD::
045624                                  GPRMA   HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
045624  000031                          .WORD   T$CODE
045626  045644                          .WORD   HPM1
045630  160010                          .WORD   T$LOLIM
045632  177776                          .WORD   T$HILIM
045634                                  GPRMA   HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
045634  001031                          .WORD   T$CODE
045636  045700                          .WORD   HPM2
045640  000000                          .WORD   T$LOLIM
045642  000776                          .WORD   T$HILIM
                                        ;GPRMC  HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
045644                                  ENDHRD
                                        .EVEN
045644                          L10107:
045644     104     105     126  HPM1:   .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
045700     111     116     124  HPM2:   .ASCIZ  'INTERRUPT VECTOR    '
045724     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.
                                ;--
045754                                  BGNSFT
045754  000003                          .WORD L10110-L$SOFT/2
045756                          L$SOFT::
                                ;       GPRML   SPM1,0, 1,YES                   ; GET TRANSPORT TEST FLAG.
045756                                  GPRML   SPM4,2,-1,YES                   ; GET ITERATION CONTROL.
045756  001130                          .WORD   T$CODE
045760  046014                          .WORD   SPM4
045762  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
045764                                  ENDSFT
                                        .EVEN
045764                          L10110:

045764     105     116     101  SPM1:   .ASCIZ  'ENABLE TRANSPORT TESTS '
046014     111     116     110  SPM4:   .ASCIZ  'INHIBIT ITERATIONS     '
046044     120     105     122  SPM6:   .ASCIZ  'PER TEST ERROR LIMIT   '
046074     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).
                                ;

046124                          PATCH::

046124                                  .BLKW   32.

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

SYMBOL TABLE

```
ADDSSR  012126 G     C$AU  = 000052      DEVDRO  023332      FRESIZ  003124 G     INTFLA  016145
ADR   = 000020 G     C$AUTO= 000061      DEVNRD  023251      FUSI    004115      INTMAS  016144
AMBTSS  006633       C$BRK = 000022      DEVNXR  023167      F$AU  = 000015      INTR    016216 G
ASSEMB= 000010       C$BSEG= 000004      DEVONL  023117      F$AUTO= 000020      INTREC  002222 G
A1716 = 000003       C$BSUB= 000002      DEVSUM  023062      F$BGN = 000040      INTVEC  016146
BADDAT  003154 G     C$CEFG= 000045      DFPTBL  002154 G    F$CLEA= 000007      INTX    004276
BADSSR  015700 G     C$CLCK= 000062      DIAGMC= 000000      F$DU  = 000016      INVERT  021142 G
BDVPCR= 177520 G     C$CLEA= 000012      DICEA = 000001      F$END = 000041      IOKCKI= 000200
BENBSW  002226 G     C$CLOS= 000035      DSBINT  016204      F$HARD= 000004      IOKSTP= 000001
BIE   = 040000       C$CLP1= 000006      DUAD12  004641      F$HW  = 000013      IPRI    002210 G
BIT0  = 000001 G     C$CVEC= 000036      DUFLG   003110 G    F$INIT= 000006      ISR   = 000100 G
BIT00 = 000001 G     C$DCLN= 000044      DUMMY   003060      F$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$ERHR= 000056      EMAXDU  016777      F$SEG = 000003      I$HRD = 000041
BIT07 = 000200 G     C$ERRO= 000060      EN    = 000000      F$SOFT= 000005      I$INIT= 000041
BIT08 = 000400 G     C$ERSF= 000051      ENAINT  016152      F$SRV = 000010      I$MOD = 000041
BIT09 = 001000 G     C$ERSO= 000057      ENVIRN  020630      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   006356      F$TEST= 000001      I$PTAB= 000041
BIT11 = 004000 G     C$ESUB= 000003      EPRT2   006356      GDDAT   003156 G    I$PWR = 000041
BIT12 = 010000 G     C$ETST= 000001      ERCM    011733      GERRMA  002172 G    I$RPT = 000041
BIT13 = 020000 G     C$EXIT= 000032      ERRHI   002234 G    GETPAT  020174 G    I$SEG = 000041
BIT14 = 040000 G     C$GETB= 000026      ERRK    016756      GETSEL  020256 G    I$SETU= 000041
BIT15 = 100000 G     C$GETW= 000027      ERRLO   002236 G    G$CNTO= 000200      I$SFT = 000041
BIT2  = 000004 G     C$GMAN= 000043      ERRNU = 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  003374      G$EXCP= 000400      I$TST = 000041
BIT5  = 000040 G     C$GPRI= 000040      ERTABL  003174      G$HILI= 000002      J$JMP = 000167
BIT6  = 000100 G     C$INIT= 000011      ESUM    016760      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  015502 G    G$OFSI= 000376      KIPAR3= 172346
BOE   = 000400 G     C$MSG = 000023      EXPD    002230 G    G$PRMA= 000001      KIPAR4= 172350
BRINIT  004455       C$OPEN= 000034      EXPGOT  004531      G$PRMD= 000002      KIPAR5= 172352
BSEL0 = 000000       C$PNTB= 000014      EXPGT2  004565      G$PRML= 000000      KIPAR6= 172354
BSEL1 = 000001       C$PNTF= 000017      EXPMSG  002320 G    G$RADA= 000140      KIPAR7= 172356
CHKAMB  016044       C$PNTS= 000016      EXPREC  015474 G    G$RADB= 000000      KIPDR0= 172300
CHKMAN  020500 G     C$PNTX= 000015      EXTA    005770      G$RADD= 000040      KIPDR1= 172302
CHKTSS  016336       C$QIO = 000377      EXTEND  005766      G$RADL= 000120      KIPDR2= 172304
CKDROP  017202       C$RDBU= 000007      EXTFEA  002224 G    G$RADO= 000020      KIPDR3= 172306
CKEMAX  017102       C$REFG= 000047      E$END = 002100      G$XFER= 000004      KIPDR4= 172310
CKMSG   011360 G     C$RESE= 000033      E$LOAD= 000035      G$YES = 000010      KIPDR5= 172312
CKMSG2  011500 G     C$REVI= 000003      FATERR= 000060      HIADDR= 001400      KIPDR6= 172314
CKRAM   011114 G     C$RFLA= 000021      FATFLG  002220 G    HOE   = 100000 G    KIPDR7= 172316
CKRAM2  011224 G     C$RPT = 000025      FERCM   011722      HPM1    045644      KTENAB  003132 G
CMDPKT  021214 G     C$SEFG= 000046      FIFEXP  012170 G    HPM2    045700      KTFLG   003130 G
CMPMEM  017660       C$SPRI= 000041      FIF1MS  012242      HPM3    045724      KTINIT  021010
CONFIG  017250       C$SVEC= 000037      FIF2MS  012311      IBE   = 010000 G    KTOFF   017274
COUNT   002306 G     C$TPRI= 000013      FILLME  017422      IDU   = 000040 G    KTON    017256
CSRADD  002204 G     DATA    002310 G    FNOINT  004213      IER   = 020000 G    LERRMA  002170 G
CTAB    003162 G     DATASC  020232      FORCER  002174 G    IFAULT  004254      LISTAL= 000001
CTABE   003174 G     DEBUGM  011632      FREE    003122 G    INCERK  017044      LOE   = 040000 G
CTABM   003162 G     DEVCNT  002216 G    FREEHI  003126      INTCPC  016150      LOOPCN  002214 G
```

SYMBOL TABLE

```
LOOPCO  013126          L10001  002174          L10073  036746          NXR     003736          PRIO5 = 000240 G
LOOPFL  003160 G        L10002  005764          L10074  037152          NXRERR  005734 G        PRIO6 = 000300 G
LOT   = 000010 G        L10003  012044          L10075  043452          NXRX    003775          PRIO7 = 000340 G
L$ACP   002110 G        L10004  012062          L10076  040742          NXTU    021756          PRMESS  014242
L$APT   002036 G        L10005  012100          L10077  041334          OFL   = 000100          PRMNO   002316 G
L$AU    022306 G        L10006  012106          L10100  042224          ONEFIL= 000000          PRMSGE  014552 G
L$AUT   002070 G        L10007  012124          L10101  042452          O$APTS= 000000          PRMSG0  014732
L$AUTO  022512 G        L10010  012142          L10102  045620          O$AU  = 000001          PRMSG1  014777
L$CCP   002106 G        L10011  012156          L10103  043720          O$BGNR= 000001          PRMSG2  015035
L$CLEA  022572 G        L10012  012240          L10104  044156          O$BGNS= 000001          PROFSC  014420
L$CO    002032 G        L10013  012410          L10105  044376          O$DU  = 000001          PR1ASC  014465
L$DEPO  002011 G        L10014  013124          L10106  044622          O$ERRT= 000000          PST32W  003150 G
L$DESC  003406 G        L10015  013752          L10107  045644          O$GNSW= 000001          PUNIT   022240
L$DESP  002076 G        L10016  013774          L10110  045764          O$POIN= 000001          PW.D11= 000021
L$DEVP  002060 G        L10017  015500          MEMADD  013754 G        O$SETU= 000000          PW.D13= 000022
L$DISP  002124 G        L10020  015506          MEMCK   021232 G        PASRPT  022010          PW.D22= 000020
L$DLY   002116 G        L10021  015514          MENASC  020447          PATCH   046124 G        PW.NOP= 000000
L$DTP   002040 G        L10022  015526          MENERR  020374          PATDAT  020230          PW.NO1= 000023
L$DTYP  002034 G        L10023  015550          MENRES  020476          PC.ERA= 002400          PW.RDE= 000024
L$DU    022404 G        L10024  015576          MMVEC = 000250          PC.IER= 002000          PW.RDR= 000001
L$DUT   002072 G        L10025  015736          MSA.FR= 000006          PC.NOO= 001000          PW.RDS= 000005
L$DVTY  003400 G        L10026  016246          MSA.NO= 000000          PC.REL= 000000          PW.RFI= 000003
L$EF    002052 G        L10030  022236          MSA.NR= 000004          PC.REW= 000400          PW.WCT= 000006
L$ENVI  002044 G        L10031  022402          MSA.VO= 000002          PKBCNT= 000006          PW.WFI= 000004
L$ETP   002102 G        L10032  022510          MSGEXP  012144 G        PKHI  = 000004          PW.WFM= 000007
L$EXP1  002046 G        L10033  022570          MSGLOO  013064 G        PKLOW = 000002          PW.WMI= 000010
L$EXP4  002064 G        L10034  022616          MSGSTA  012350 G        PKTADD  007552          PW.WNP= 000011
L$EXP5  002066 G        L10035  023060          MSGSUB  013742 G        PKTFRM  007514          PW.WTR= 000002
L$HARD  045624 G        L10036  023620          MS.ATT= 000006          PKTGET  012064 G        P.ACK = 100000
L$HIME  002120 G        L10037  023466          MS.EXT= 000200          PKTMES  012110 G        P.CMD = 000037
L$HPCP  002016 G        L10040  023550          MS.RSD= 000001          PKTRAM  004743 G        P.CONT= 000012
L$HPTP  002022 G        L10041  024316          MS.RSF= 000020          PKTSSR  012046 G        P.CVC = 040000
L$HW    002154 G        L10042  025010          MS.RST= 000010          PNT   = 001000 G        P.FMT = 000140
L$ICP   002104 G        L10043  026344          M2901   026070          PRAMPK  013776          P.FORM= 000011
L$INIT  021512 G        L10044  025252          M8186   005552          PRASC   014523          P.GETS= 000017
L$LADP  002026 G        L10045  025552          M8189   005643          PRBEXP  015470          P.IE  = 000200
L$LAST  046404 G        L10046  026050          NBA   = 002000          PRBMSG  015336          P.INIT= 000013
L$LOAD  002100 G        L10047  027450          NEWPAS  021744          PRBREC  015472          P.MODE= 007400
L$LUN   002074 G        L10050  026716          NODEV   003112 G        PRBTOT  015423          P.OPP = 020000
L$MREV  002050 G        L10051  027266          NOINIT  004333          PRBYTE  015122 G        P.POSI= 000010
L$NAME  002000 G        L10052  030722          NOINTR  004217          PRI   = 002000 G        P.READ= 000001
L$PRIO  002042 G        L10053  030000          NOITS   002166 G        PRIADD  010156          P.SWB = 010000
L$PROT  021502 G        L10054  030324          NOMAN   020534          PRIAO   010226          P.WRIT= 000005
L$PRT   002112 G        L10055  034310          NOMEM   005456          PRIBXO  007610 G        P.WRTC= 000004
L$REPP  002062 G        L10056  031254          NP.IR = 000200          PRIEQU  010056          P.WRTS= 000006
L$REV   002010 G        L10057  031540          NP.LOO= 000040          PRIPKT  007366 G        QVP     002202 G
L$RPT   022620 G        L10060  032004          NP.OUT= 000100          PRIRAM  010064          RAMASC  014156
L$SOFT  045756 G        L10061  032232          NP.WRP= 000020          PRITAD  010272          RAMDAT  002240 G
L$SPC   002056 G        L10062  032476          NSI     004150          PRITSS  006022          RAMERR  015510 G
L$SPCP  002020 G        L10063  033036          NSINIT  004405          PRITO   010354          RAMEXP  015530 G
L$SPTP  002024 G        L10064  035214          NUL     004525          PRIT1   010417          RAMFOR  010114
L$STA   002030 G        L10065  040340          NULCR   004526          PRIXOR  007740 G        RAMSIZ  002300 G
L$SW    002164 G        L10066  035456          NXM   = 004000          PRIO0 = 000000 G        RAMTAD  015516 G
L$TEST  002114 G        L10067  035716          NXMFLG  003134 G        PRIO1 = 000040 G        RCVHIA  002302 G
L$TIML  002014 G        L10070  036122          NXMHI   003140 G        PRIO2 = 000100 G        RCVLOA  002304 G
L$UNIT  002012 G        L10071  036310          NXMLO   003136 G        PRIO3 = 000140 G        RDERR   005204
L10000  002162          L10072  036514          NXMTST  021406          PRIO4 = 000200 G        RECMSG  002464 G
```

SYMBOL TABLE

RECV	002232 G	S1.IEO=	010000	TST6ID	030703	T1.1	023422	T4	025012 G
REGSAV	020140	S1.IFM=	001000	TST7ID	034213	T1.2	023502	T4LOOP	025032
RETERR	005370	S1.IHE=	000400	TST8ID	035177	T10	040342 G	T4.1	025012
REWIND	011014 G	S1.IID=	004000	TST9ID	040241	T10BFR	042512	T4.2	025254
RMCHBE=	000167	S1.I1R=	020000	TSV2	002000 G	T10BUF	042554	T4.3	025554
RMCHEN=	000200	S1.I2R=	040000	TSV3	002174 G	T10DAT	042500	T5	026346 G
RMMSGB=	000215	S1.PAR=	100000	TSV4	021502 G	T10DTA	042542	T5ADDR	027406
RMMSGE=	000234	S2.ATI=	000010	TSV5	023402 G	T10INT	043210	T5LOOP	026364
RMPKTB=	000201	S2.BTI=	000004	TSV6	045622 G	T10LOO	040360	T5MEM	027350
RMPKTE=	000210	S2.DIM=	000200	TTIBFR=	177562 G	T10MBF	042574	T5.1	026370
RMR =	010000	S2.ILW=	000100	TTICSR=	177560 G	T10NBA	042671	T5.2	026736
RWPACK	011110	S2.INR=	000020	TTIVEC=	000060 G	T10NIN	043117	T6	027452 G
SC =	100000	S2.OUT=	000040	T$ARGC=	000003	T10NNB	042753	T6INT	030473
SCE =	020000	S2.UND=	000003	T$CODE=	001130	T10PAC	042470	T6LOOP	027470
SCHERR	005276	TBLEND=	003060 G	T$ERRN=	002144	T10PKT	042532	T6NBA	030370
SCME	005011	TCOASC	006474	T$EXCP=	000000	T10RST	043326	T6NINT	030551
SDELAY	010660	TCOCOC	006674	T$FLAG=	000040	T10RT2	043400	T6PACK	030360
SELASC	020442	TEMP1	003114 G	T$GMAN=	000000	T10SSR	043030	T6SSR	030415
SELDAT=	000004	TEMP2	003116 G	T$HILI=	000776	T10.1	040360	T6TSBA	030631
SEL2 =	000002	TERCLS=	000016	T$LAST=	000001	T10.2	040744	T6.1	027470
SETMAP	017316	TESTNO=	000013	T$LOLI=	000000	T10.3	041336	T6.2	030014
SETU	022042	TEXASC	006433	T$LSYM=	010000	T10.4	042226	T7	030724 G
SFFMSG	012102 G	TFCASC	006535	T$LTNO=	000013	T11	043454 G	T7BFR	033106
SFHERR	003703	TIMEXP	015552 G	T$NEST=	177777	T11BFR	044662	T7DATA	033070
SFIERR	003650	TIMSGO	015600	T$NS0 =	000000	T11BF2	044732	T7INT	034041
SFIMSG	012034 G	TINERR	012021	T$NS1 =	000005	T11DAT	044650	T7LOOP	030742
SFPTBL	002164 G	TMPBFR	002630 G	T$NS2 =	000002	T11DTA	044720	T7NBA	033220
SIFLAG	003152 G	TNAM	016704	T$NS3 =	000003	T11LOO	043472	T7NINT	033750
SIMSG	011766	TRANST	002164 G	T$PTNU=	000000	T11NBA	044752	T7PACK	033060
SKIPT	003376	TSBA =	000000 G	T$SAVL=	177777	T11NIN	045334	T7REST	034242
SOFINI	015774 G	TSBAH =	000001 G	T$SEGL=	177777	T11PAC	044640	T7SP	033100
SPACE	010464 G	TSBAM2	026160	T$SEKO=	010000	T11PK2	044710	T7SSR	033661
SPM1	045764	TSBAM3	026242	T$SUBN=	000004	T11RES	045514	T7TSBA	034130
SPM4	046014	TSDB =	000000 G	T$TAGL=	177777	T11RT2	045556	T7.1	030742
SPM6	046044	TSDBH =	000001 G	T$TAGN=	010111	T11SR2	045260	T7.2	031270
SPM7	046074	TSFCOD	007234	T$TEMP=	000000	T11SSR	045204	T7.3	031542
SR0 =	177572	TSREJ =	000006	T$TEST=	000013	T11TSB	045412	T7.4	032006
SR1 =	177574	TSSDEF	006604	T$TSTM=	177777	T11.1	043472	T7.5	032234
SR2 =	177576	TSSR =	000002 G	T$TSTS=	000001	T11.2	043734	T7.6	032500
SR3 =	172516	TSSRBI	003500 G	T$$AU =	010031	T11.3	044160	T72DAT	033126
SSR =	000200	TSSRFO	006413	T$$AUT=	010033	T11.4	044400	T72DON=	033142
STATCO	012412	TSSRH =	000003 G	T$$CLE=	010034	T112RE	045012	T72NBA	033142
SVCGBL=	000000	TSSX	003016	T$$DU =	010032	T113RE	045073	T72REJ	033273
SVCINS=	000000	TSTBLK	002750 G	T$$HAR=	010107	T114RE	045123	T73REJ	033372
SVCSUB=	000001	TSTCNT	002212 G	T$$HW =	010000	T2	023622 G	T74REJ	033465
SVCTAG=	000000	TSTEND	016720	T$$INI=	010030	T2LOOP	023640	T75REJ	033563
SVCTST=	000001	TSTFLA	002312 G	T$$MSG=	010025	T2SSR	024216	T8	034312 G
S$LSYM=	010000	TSTLOO	016456 G	T$$PRO=	010027	T2TSBA	024104	T8BFR	034742
SO.IDB=	000010	TSTPTR	002314 G	T$$RPT=	010035	T2TSSR	024151	T8DATA	034730
SO.IFB=	000002	TSTSET	016510 G	T$$SEG=	010000	T23A	003142 G	T8LOOP	034330
SO.IFP=	000001	TST1ID	023600	T$$SOF=	010110	T23B	003144 G	T8NVCK	035017
SO.ILD=	000020	TST10I	043277	T$$SRV=	010026	T3	024320 G	T8PACK	034720
SO.ION=	000040	TST11I	045462	T$$SUB=	010106	T3BFLG	003146 G	T8SSR	035110
SO.IRD=	000100	TST2ID	024270	T$$SW =	010001	T3LOOP	024336	T8VCK	034762
SO.IRW=	000004	TST3ID	024763	T$$TES=	010102	T3SSR	024712	T9	035216 G
SO.ISP=	000200	TST4ID	026322	T1	023402 G	T3TSBA	024602	T9BFR	037212
S1.ICE=	002000	TST5ID	027320	T1LOOP	023420	T3TSSR	024646	T9DATA	037200

```
TSV6   PARAMETER CODING MACRO M1113  14-JUN-84 15:15                                                    SEQ 0180
SYMBOL TABLE

T9INT   040067          UAM    = 000200 G       WRTCHR  010662 G        XSOONL= 000100          X2.EXT= 000200
T9LOOP  035240          UNITN    002200 G       WRTERR  005111          XSOPED= 000010          X2.OPM= 100000
T9NBA   037246          UNREC  = 000006         WRTMSG  005054          XSORLL= 010000          X2.RCE= 040000
T9NINT  037776          USI      004121         WSMBK   021224 G        XSORLS= 040000          X2.REV= 000077
T9PACK  037170          WAITF    016250 G       XFERAS  015740          XSOTMK= 100000          X2.SPA= 035400
T9REST  040266          WC.IFA= 000200          XNXM    016376          XSOVCK= 000020          X2.UNI= 000007
T9SSR   037707          WC.IFE= 000002          XORBFO  007672          XSOWLE= 004000          X2.WCF= 002000
T9TSBA  040156          WC.IGO= 000001          XORFOR  010010          XSOWLK= 000004          X3.DCK= 000010
T9.1    035240          WC.IRE= 000010          XSTO   = 000006 G       XXCOMM  003120 G        X3.MBZ= 000006
T9.2    035506          WC.IRW= 000004          XST1   = 000010 G       X$ALWA= 000000          X3.MDE= 177400
T9.3    035720          WC.IOT= 000100          XST2   = 000012 G       X$FALS= 000040          X3.OPI= 000100
T9.4    036124          WC.I1T= 000040          XST3   = 000014 G       X$OFFS= 000400          X3.REV= 000040
T9.5    036312          WC.I5R= 000020          XST4   = 000016 G       X$TRUE= 000020          X3.RIB= 000001
T9.6    036516          WF.IED= 000010          XSOBOT= 000002          X1.COR= 020000          X3.SPA= 000200
T9.7    036750          WF.IER= 000004          XSOEOT= 000001          X1.DLT= 100000          X3.TRF= 000020
T92DAT  037232          WF.IHI= 000200          XSOIE  = 000040         X1.MBZ= 017375          X4.HSP= 100000
T92DON= 037246          WF.IRE= 000040          XSOILA= 000400          X1.RBP= 000400          X4.MBZ= 017400
T92REJ  037321          WF.IWF= 000020          XSOILC= 001000          X1.SPA= 040000          X4.RCE= 040000
T93REJ  037420          WF.IWR= 000100          XSOLET= 020000          X1.UNC= 000002          X4.TSM= 020000
T94REJ  037513          WF.I3R= 000002          XSOMOT= 000200          X2.BUF= 000100          X4.WRC= 000377
T95REJ  037611          WF.I4R= 000001          XSONEF= 002000

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

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