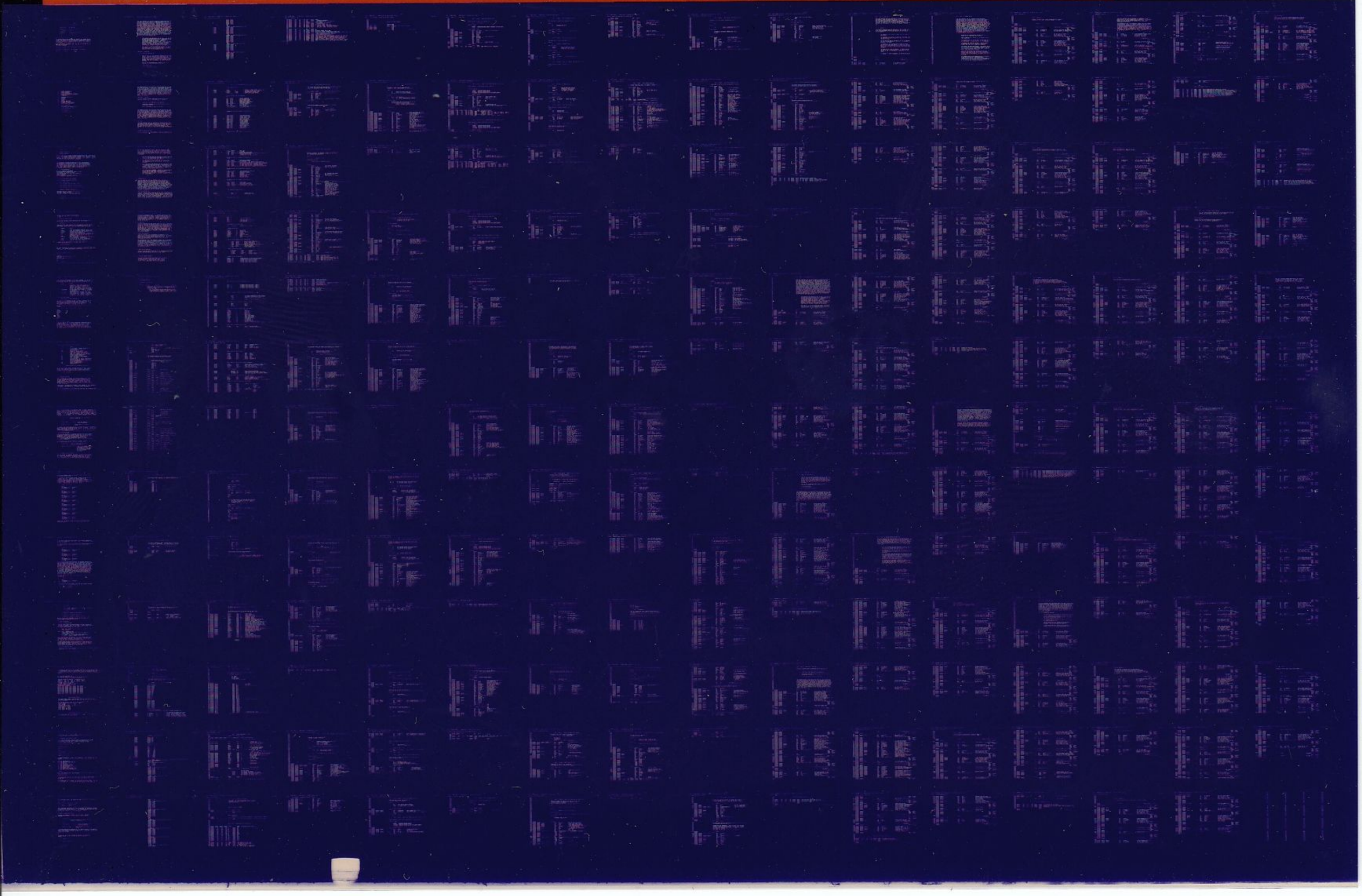
11/21+ TSV05 CTRL LT1
TSV05 CNTSAA0

AH-T815A-MC
FICHE 01 OF 02

JUL 1984
digital
Made In USA



11/Z1+ TSV05 CTRL LT1
TSV05 CNTSAA0

AH-T815A-MC
FICHE 0Z OF 0Z

JUL 1984
digital
Made In USA

.REM_

IDENTIFICATION

PRODUCT ID: AC T814A MC

PRODUCT TITLE: CNTSAAO TSV05 CTRL LT1

DECO/DEPO: 1.0

DEPARTMENT: ISS/DIAGNOSTIC SERVICES

DATE: APRIL 09, 1984

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS IS A SBC-11/21+ RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF A TSV05 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A SBC-11/21+ SYSTEM (Q BUS). THE PROGRAM PROVIDES ERROR MESSAGES WHICH IDENTIFY FAILING FUNCTIONS THAT AID IN THE REPAIR OF THE DEVICE. THIS DIAGNOSTIC CONSISTS 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

SBC-11/21+ PROCESSOR AND MEMORY
CAUTION:DIAGNOSTIC REQUIRES 32K WORDS OF MEMORY
(28K USEABLE I.E. 4K FOR I/O PAGE)
TSV05 MAGTAPE SUBSYSTEM (DRIVE AND CONTROLLER)
CONSOLE TERMINAL
PDP 11 DIAGNOSTIC SUPERVISOR (MSAAA.SYS VERSION 34 OR LATER)
PDP 11 DIAGNOSTIC LOADER/MONITOR (XXDP+)

1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

1. XXDP+ USERS GUIDE
2. TSV05 TRANSPORT SUBSYSTEM USER'S GUIDE
3. TSV05 TRANSPORT SUBSYSTEM TECHNICAL MANUAL
4. TSV05 TRANSPORT SUBSYSTEM INSTALLATION MANUAL

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

FUNCTIONAL SBC-11/21+ CENTRAL PROCESSOR AND MEMORY
FUNCTIONAL CONSOLE TERMINAL
FUNCTIONAL STANDALONE DIAGNOSTIC SUPERVISOR
FUNCTIONAL DIAGNOSTIC LOADER/MONITOR (XXDP+)

1.5 ASSUMPTIONS

ALL HARDWARE EXCEPT THE HARDWARE UNDER TEST IS ASSUMED TO WORK

PROPERLY OR FALSE ERRORS CAN BE REPORTED.
THE TAPE BEING USED ON THE TS05 TRANSPORT IS A KNOWN GOOD REEL
OF TAPE.

2.0 OPERATING INSTRUCTIONS

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

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 SBC-11/21+ DIAGNOSTIC SUPERVISOR COMPATIBLE
PROGRAM. ALL LOADING AND RUNTIME INSTRUCTIONS CAN BE REFERENCED IN THE
XXDP+ USERS GUIDE. THE USER ENTRY IS IN QUOTES.

BOOT THE DIAGNOSTIC MEDIA

```
.R NTSA??
DIAG. RUN-TIME SERVICES REV D. APR 79
CNTSA-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 = 176000, 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) 176000 ? <ENTER THE ADDRESS OF THE
                              TSBA/TSDB REGISTER>

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

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

2.5 SOFTWARE QUESTIONS

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

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

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

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

2.6 EXTENDED P-TABLE DIALOGUE

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

DEVICE WITH SEVERAL LINES, THIS BECOMES TEDIOUS SINCE MOST OF
THE ANSWERS ARE REPETITIOUS.

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

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

UNIT 1
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE @ (O) ?  0<CR>
Q-FACTOR (O) 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 MFSSAGE 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
CNTSA HRD ERR 01610 ON UNIT 00 TST 016 SUB 002 PC: 040624
FIFO STATUS (IN WORD  9) INCORRECT AFTER WRITE FIFO

 TAPE BUS SIGNALS IN WORD #8: - DESIGNATOR <BIT #>
PARERR<15>  IEOT  <12>  IFMK  <9>  IRDY<6>  IRWD<2>
IRESV2<14>  IIDENT<11>  IHER  <8>  IONL<5>  IFBY<1>
IRESV1<13>  ICER  <10>  ISPEED<7>  ILDP<4>  IFPT<0>
 TAPE BUS SIGINALS IN WORD #9:
DATMIS<7>  ILW<6>  OUTRDY<5>  INRDY<4>
 MESSAGE BUFFER ADDRESS = 047352
 MESSAGE BUFFER CONTENTS:
WORD #0 EXPD: 100020  RECV: 100020  XOR: 000000
WORD #1 EXPD: 000012  RECV: 000012  XOR: 000000
WORD #2 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #3 EXPD: 000010  RECV: 000010  XOR: 000000
WORD #4 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #5 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #6 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #7 EXPD: 000000  RECV: 000000  XOR: 000000
WORD #8 EXPD: 070217  RECV: 070217  XOR: 000000
WORD #9 EXPD: 000074  RECV: 000034  XOR: 000040
```

ERROR MESSAGE EXAMPLE 2

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

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

ERROR MESSAGE EXAMPLE 3

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


CNTS HRD ERR 00121 ON UNIT 00 TST 001 SUB 002 PC: 023306
```

```
MOT BIT (XSTO) NOT SET DURING REWIND (EXTENDED FEATURES MODE)
 EXPD: 000312 RECV: 000112 XOR: 000200
```

4.0 PERFORMANCE AND PROGRESS REPORTS

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

SUCCESSFUL RUN EXAMPLE (PDP 11/23)

```
 DR>STA/FLA:PNT:HOE

 UNITS (D) ?  1

 UNIT 0

 DEVICE ADDRESS  (O) 176000 ? <CR>

 VECTOR (O) 224 ? <CR>

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

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

```
TST: 001 INITIALIZE #1
TST: 002 WRAP DATA HIGH BYTE TEST
TST: 003 WRAP DATA LOW BYTE TEST
TST: 004 RAM TEST
TST: 005 I.ITIALIZE  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 THE FALCON+ PROCESSOR.

THE PROGRAM RUNS IN TWO MODES; NO ITERATIONS AND DEFAULT MODE. IN THE NO ITERATIONS MO , EACH TEST IS RUN ONCE, WITH NO ITERATIONS. IN THE DEFAULT MODE EACH ST IS REPEATED BY THE NUMBER OF TIMES INDICATED Y

THE ITERATION COUNT. NO ITERATIONS MODE IS SELECTED BY ANSWERING THE INHIBIT ITERATIONS QUESTION WITH A "Y" (YES).

TIMES REQUIRED TO RUN TESTS 1 THROUGH 12:

Q.V.	2 MIN 15 SECONDS
DEFAULT	16 MINS

MORE EXHAUSTIVE CHECKS ARE AVAILABLE BY ALLOWING THE DIAGNOSTIC PROGRAMS TO RUN FOR MORE THAN ONE PASS. THE SECOND PASS OF THE PROGRAM IS MORE COMPREHENSIVE THAN THE FIRST PASS. ALL ITERATIONS AFTER THE FIRST PASS ARE THE SAME, HOWEVER, THEY ARE SUBSTANTIALLY LONGER.

5.0 DEVICE INFORMATION TABLES

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

CHANGE HW (L) ?

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

UNIT 0

DEVICE ADDRESS (O) 176000 ? <ENTER THE ADDRESS OF THE TSBA/TSDB REGISTER>

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

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

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

CHANGE SW (L) ?

INHIBIT ITERATIONS (L) N ?

6.0 TEST SUMMARIES

TEST 1: BUS RESET TEST

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

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

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

TEST 2: WRAP DATA - HIGH BYTE

THIS TEST VERIFIES OPERATION OF:

1. PART OF THE 11/21+ BUS INTERFACE SECTION OF THE M7196 MODULE: PART OF THE INPUT FILE (TSDB HIGH BYTE), PART OF THE OUTPUT FILE (TSSR HIGH BYTE AND TSBA, BOTH BYTES), PART OF THE DC005 TRANSCEIVER CIRCUITS (ADDRESS DECODER, BDAL DRIVERS, HIGH BYTE OF INTERNAL DAL BUS DRIVERS), AND BASIC PROGRAMMED I/O CONTROL SEQUENCES AND LOGIC;

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

3. Y AND SOURCE BUSES;

4. BASIC MICROPROGRAM SEQUENCES.

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

TEST 3: WRAP DATA - LOW BYTE

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

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

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

TEST 4: RAM TEST

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

TEST 5: SECOND INITIALIZATION TEST

THIS TEST VERIFIES THE SAME ELEMENTS AS DID INITIALIZATION TEST #1 AND ALSO CHECKS THAT CERTAIN PARTS OF RAM IS CLEARED TO ZERO

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

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

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

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

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

TEST 6: COMMAND REJECT

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

TEST 7: WRITE CHARACTERISTICS

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

VERIFIED IN SUBSEQUENT TESTS. ALL COMMANDS EXECUTED IN THIS TEST HAVE THE INTERRUPT ENABLE (IE) BIT CLEARED TO ZERO, SO NO INTERRUPTS SHOULD BE GENERATED. HOWEVER, THE PROGRAM RUNS AT PROCESSOR PRIORITY 0, WITH THE INTERRUPT SERVICE ROUTINE SET UP TO FLAG UNEXPECTED INTERRUPTS. IF AN INTERRUPT OCCURS, A PROBLEM EXISTS IN EITHER THE 11/21+ 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

CVTSAAO => CNTSAAO JAKI BERG 9-APR-1984

CHANGES WERE MADE TO CVTSAAO TO PRODUCE CNTSAAO FOR THE FALCON-PLUS PROJECT (SBC-11/21+). CHANGES, MARKED BY ";JB REV A-0", ARE:

- SET THE ODT BREAK VECTOR (LOCATION 140) TO THE STARTING ADDRESS OF FALCON'S ODT ROM (170000-OCTAL).
- LOWER THE GENERAL INTERRUPT PRIORITY FROM 7 TO 6.
  CHANGE DEFAULT CSR ADDRESS FROM 172540 TO 176000.

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

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

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


002000                                  POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
002000                                  HEADER  CNTSA,A,0,655.,0
002000                          L$NAME::                ;DIAGNOSTIC NAME
002000  103                             .ASCII /C/
002001  116                             .ASCII /N/
002002  124                             .ASCII /T/
002003  123                             .ASCII /S/
002004  101                             .ASCII /A/
002005  000                             .BYTE   0
002006  000                             .BYTE   0
002007  000                             .BYTE   0
002010
002010  101                     L$REV::                 ;REVISION LEVEL
                                        .ASCII  /A/
002011
002011  060                     L$DEPO::                ;0
                                        .ASCII  /0/
002012
002012  000000                  L$UNIT::                ;NUMBER OF UNITS
                                        .WORD   0
002014
002014  001217                  L$TIML::                ;LONGEST TEST TIME
                                        .WORD   655.
002016
002016  045654                  L$HPCP::                ;POINTER TO H.W. QUES.
                                        .WORD   L$HARD
002020
002020  046006                  L$SPCP::                ;POINTER TO S.W. QUES.
                                        .WORD   L$SOFT
002022
002022  002154                  L$HPTP::                ;PTR  TO DEF. H.W. PTABLE
                                        .WORD   L$HW
002024
002024  002164                  L$SPTP::                ;PTR. TO S.W. PTABLE
                                        .WORD   L$SW
002026
002026  046404                  L$LADP::                ;DIAG. END ADDRESS
                                        .WORD   L$LAST
002030
002030  000000                  L$STA::                 ;RESERVED FOR APT STATS
                                        .WORD   0
002032                          L$CO::
002032  000000                            .WORD   0
002034
002034  000000                  L$DTYP::                ;DIAGNOSTIC TYPE
                                        .WORD   0
002036
002036  000000                  L$APT::                 ;APT EXPANSION
                                        .WORD   0
002040
002040  002124                  L$DTP::                 ;PTR. TO DISPATCH TABLE
                                        .WORD   L$DISPATCH
```

SEQ 0020

PROGRAM HEADER

```
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$DEVP::                ; POINTER TO DEVICE TYPE LIST
002060  003376                          .WORD   L$DVTYP
002062                          L$REPP::                ;PTR. TO REPORT CODE
002062  022702                          .WORD   L$RPT
002064                          L$EXP4::
002064  000000                          .WORD   0
002066                          L$EXP5::
002066  000000                          .WORD   0
002070                          L$AUT::                 ;PTR. TO ADD UNIT CODE
002070  022370                          .WORD   L$AU
002072                          L$DUT::                 ;PTR. TO DROP UNIT CODE
002072  022466                          .WORD   L$DU
002074                          L$LUN::                 ;LUN FOR EXERCISERS TO FILL
002074  000000                          .WORD   0
002076                          L$DESP::                ;POINTER TO DIAG. DESCRIPTION
002076  003404                          .WORD   L$DESC
002100                          L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
002100  104035                          EMT     E$LOAD
002102                          L$ETP::                 ;POINTER TO ERRTBL
002102  000000                          .WORD   0
002104                          L$ICP::                 ;PTR. TO INIT CODE
002104  021546                          .WORD   L$INIT
002106                          L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  022654                          .WORD   L$CLEAN
002110                          L$ACP::                 ;PTR. TO AUTO CODE
002110  022574                          .WORD   L$AUTO
002112                          L$PRT::                 ;PTR. TO PROTECT TABLE
002112  021536                          .WORD   L$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
31
```

DISPATCH TABLE

```
                                        .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  023464                          .WORD   T1
002126  023704                          .WORD   T2
002130  024402                          .WORD   T3
002132  025074                          .WORD   T4
002134  026430                          .WORD   T5
002136  027534                          .WORD   T6
002140  031004                          .WORD   T7
002142  034372                          .WORD   T8
002144  035276                          .WORD   T9
002146  040412                          .WORD   T10
002150  043514                          .WORD   T11
```

DEFAULT HARDWARE P TABLE

```
                                        .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   DF'TBL          ;DEFAULT HARD-P TABLE
002152  000003                          .WORD   L10000-L$HW/2
002154                          L$HW::
002154                          DFPTBL::

002154  176000                          .WORD   176000          ; 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
```

SOFTWARE P-TABLE

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


002174                                  BGNMOD  TSV3
002174                          TSV3::


                                        .SBTTL  GLOBAL EQUATES SECTION

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

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

GLOBAL EQUATES SECTION

```
                        ;
                        ; PRIORITY LEVEL DEFINITIONS
                        ;
000340                  PRI07== 340
000300                  PRI06== 300
000240                  PRI05== 240
000200                  PRI04== 200
000140                  PRI03== 140
000100                  PRI02== 100
000040                  PRI01== 40
000000                  PRI00== 0
                        ;
                        ;OPERATOR FLAG BITS
                        ;
000004                  EVL==     4
000010                  LOT==    10
000020                  ADR==    20
000040                  IDU==    40
000100                  ISR==   100
000200                  UAM==   200
000400                  BOE==   400
001000                  PNT==  1000
002000                  PRI==  2000
004000                  IXE==  4000
010000                  IBE== 10000
020000                  IER== 20000
040000                  LOE== 40000
100000                  HOE== 100000
34
35 002174                       KT11    ,,              ;DEFINE MEMORY MANAGEMENT REGISTERS
                        .SBTTL  MEMORY MANAGEMENT DEFINITIONS
                        ;*KT11 VECTOR ADDRESS
000250                  MMVEC=  250
                        ;*KT11 STATUS REGISTER ADDRESSES
177572                  SR0=    177572
177574                  SR1=    177574
177576                  SR2=    177576
172516                  SR3=    172516
                        .IF NB
                        ;*USER "I" PAGE DESCRIPTOR REGISTERS
                        UIPDR0= 177600
                        UIPDR1= 177602
                        UIPDR2= 177604
                        UIPDR3= 177606
                        UIPDR4= 177610
                        UIPDR5= 177612
                        UIPDR6= 177614
                        UIPDR7= 177616
                        .IF NB
                        ;*USER "D" PAGE DESCRIPTOR REGISTORS
                        UDPDR0= 177620
                        UDPDR1= 177622
                        UDPDR2= 177624
                        UDPDR3= 177626
                        UDPDR4= 177630
                        UDPDR5= 177632
                        UDPDR6= 177634
```

```
UDPDR7= 177636
.ENDC
;*USER "I" PAGE ADDRESS REGISTERS
UIPAR0= 177640
UIPAR1= 177642
UIPAR2= 177644
UIPAR3= 177646
UIPAR4= 177650
UIPAR5= 177652
UIPAR6= 177654
UIPAR7= 177656
.IF NB
;*USER "D" PAGE ADDRESS REGISTERS
UDPAR0= 177660
UDPAR1= 177662
UDPAR2= 177664
UDPAR3= 177666
UDPAR4= 177670
UDPAR5= 177672
UDPAR6= 177674
UDPAR7= 177676
.ENDC
.ENDC
.IF NB
;*SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS
SIPDR0= 172200
SIPDR1= 172202
SIPDR2= 172204
SIPDR3= 172206
SIPDR4= 172210
SIPDR5= 172212
SIPDR6= 172214
SIPDR7= 172216
.IF NB
;*SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS
SDPDR0= 172220
SDPDR1= 172222
SDPDR2= 172224
SUPDR3= 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
SUPAR1= 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                  MIADDR= 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)
```

TSV05 REGISTER AND PACKET DEFINITIONS

```
                                ;
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
```

TSV05 REGISTER AND PACKET DEFINITIONS

```
                                ;
                                ;

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

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

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

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

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


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

```
                                ;*
                                ;
                                ;REGISTER DEFINITIONS IN THE MESSAGE BUFFER
                                ;
                                ;-

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


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

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

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

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

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

                                ;*
                                ;BSEL1 CODES FOR WRITE TAPE CONTROL
                                ;
000200                          WC.IFAD         = BIT7          ;IFAD - FORMATTER ADDRESS
000100                          WC.IOTAD        = BIT6          ;ITADO  - TRANSPORT ADDRESS BIT 0
```

```
                        WC.I1TAD        = BIT5          ;ITAD1  - TRANSPORT ADDRESS BIT 1
                        WC.I5RESV       = BIT4          ;IRESV5 - RESERVED #5
                        WC.IREW         = BIT3          ;IREW   - REWIND
                        WC.IRWU         = BIT2          ;IRWU   - REWIND AND UNLOAD
                        WC.IFEN         = BIT1          ;IFEN   - FORMATTER ENABLE
                        WC.IGO          = BIT0          ;GO

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


                        ;*
                        ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
                        ;-
                        MS.EXT          = BIT7          ;INVERT SENSE OF EXTENDED FEATURES SWITCH
                        MS.RSFIFO       = BIT4          ;RESET FIFO AND INPUT PARITY ERRORR
                        MS.RSTAPE       = BIT3          ;RESET TAPE STATUS IN 2 FLIP-FLOPS
                        MS.ATTN         = BIT2!BIT1     ;ATTENTION TRIGGER FIELD
                        MS.RSD          = BIT0          ;RESET TIMER A,B THEN DELAY TIMES IN SEL2
                        ;*
                        ; MS.ATTN SUBCODES
                        ;-
                                MSA.NOP = 0*2           ;NO OP (NOTHING TRIGGERED)
                                MSA.VOL = 1*2           ;SIMULATE ON-LINE/OFF-LINE TRANSISTION
                                MSA.NRAM= 2*2           ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
                                MSA.FRAM= 3*2           ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
                        ;*
                        ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
                        ;
                        NP.IR           = BIT7          ;INTERRUPT REQUEST (0-1 TRANSITION)
                        NP.OUT          = BIT6          ;TAPE DATA DIRECTION OUT (0= IN)
                        NP.LOOP         = BIT5          ;ENABLE TRANSPORT LOOPBACK
                        NP.WRP          = BIT4          ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
                        ;*
                        ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
                        ;-

                        S2.DIM          = BIT7          ;WORD #9 BYTE 2 DATA IN MISS
                        S2.ILW          = BIT6          ;               ILW H
                        S2.OUTRDY       = BIT5          ;               OUT RDY H
                        S2.INRDY        = BIT4          ;               IN RDY H
                        S2.ATIMR        = BIT3          ;               TIMER A FLAG H
                        S2.BTIMR        = BIT2          ;               TIMER B FLAG H
                        S2.UNDEF        = BIT1+BIT0     ;(UNDEFINED)
                        S1.PARIN        = BIT15         ;WORD #8 BYTE 1 PARIN H
                        S1.I2RESV       = BIT14         ;               IRESV2
                        S1.I1RESV       = BIT13         ;               IRESV1
                        S1.IEOT         = BIT12         ;               IEOT L
```

TSV05 REGISTER AND PACKET DEFINITIONS

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

SPECIAL MACROS AND OPDEFS.

```
                        .SBTTL  SPECIAL MACROS AND OPDEFS.


                ;+
                ;SAVE GENERAL REGS 1 TO 5
                ;
                        .MACRO  SAVREG
                        JSR     R5,REGSAV
                        .ENDM

                ;+
                ; MACRO TO FORCE AN ERROR
                ;-
                .MACRO  FORCERROR       TAG,NOTSSR
                .NLIST
                .IIF NDF LISTALL, .NLIST
                .LIST
                .IF B NOTSSR
                        MOV     TSSR(R5),R1     ;READ TSSR
                .ENDC
                        MOV     FORCER,FORCER   ;IS FORCER SET? (LEAVE C BIT ALONE)
                        BNE     TAG             ;BR IF YES
                .NLIST
                .IIF NDF LISTALL, .LIST
                .LIST
                .ENDM
                ;+
                ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
                ;       WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
                ;       SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
                ;       FORCER TO 177777
                ;       TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
                ;-
                .MACRO  FORCEXIT        TAG
                .NLIST
                .IIF NDF LISTALL, .NLIST
                .LIST
                        MOV     FORCER,FORCER   ;IS FORCER NEGATIVE?
                        BMI     TAG             ;BR IF YES
                .NLIST
                .IIF NDF LISTALL, .LIST
                .LIST
                .ENDM
                ;+
                ; MACRO TO INCREMENT ERROR COUNTS
                ;
                .MACRO  NEXT.ERRNO
                .NLIST
                ;;;.IIF NDF LISTALL, .NLIST
                        ERRNO=ERRNO+1
                ;;;.IIF NDF LISTALL, .LIST
                .LIST
                .ENDM

                ;+
```

SPECIAL MACROS AND OPDEFS.

```
                                ;MACRO TO PERFORM XOR
                                ;
                                        .MACRO  XOR     A,B
                                        MOV     A,-(SP)
                                        BIC     B,(SP)
                                        BIC     A,B
                                        BIS     (SP)+,B
                                        .ENDM

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

                                ;
                                ; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
                                ; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
                                ;
002174  000000                 FORCER::        0       ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED
                                                ; - BY THE MACRO "IFERROR").  AN ERROR NEED NOT -
                                                ; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.
```

GLOBAL DATA SECTION

```
                                        .SBTTL  GLOBAL DATA SECTION

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

                                ;
                                ;THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
                                ;SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.
                                ;
002176  000000                  EPRTSW::        .WORD   0       ;PRINT SWITCH
002200  000000                  UNITN::         .WORD   0       ;UNIT # UNDER TEST.
002202  000000                  QVP::           .WORD   0       ;QUICK VERIFY FLAG.
002204  000000                  CSRADDR::       .WORD   0       ;ADDRESS OF CSR FOR CURRENT DEVICE
002206  000224                  IVEC::          .WORD   224     ;INTERRUPT VECTOR
002210  000200                  IPRI::          .WORD   PRI04   ;INTERRUPT PRIORITY.
002212  000000                  TSTCNT::        .WORD   0       ;NUMBER OF TESTS RUN IN THIS PASS
002214  000000                  LOOPCNT::       .WORD   0       ;REMAINING ITERATION COUNT FOR TEST
002216  000000                  DEVCNT::        .WORD   0       ;NUMBER OF DEVICE UNDER TEST
002220  000000                  FATFLG::        .WORD   0       ;SET IF FATAL ERROR IS DETECTED IN TEST
002222  000000                  INTRECV::       .WORD   0       ;SET IF TAPE INTERRUPT WAS RECEIVED
002224  000000                  EXTFEA::        .WORD   0       ;EXTENDED FE TURES 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.    ;E..PECTED MESSAGE BUFFER DATA
002464                          RECMSG::        .BLKB   100.    ;RECEIVED MESSAGE BUFFER DATA
002630                          TMPBFR::        .BLKB   80.     ;TEMPORARY STORAGE FOR PRINT
```

```
                                        .SBTTL  TSTBLK  - TEST DATA TABLE

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

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

GLOBAL ENVIRONMENT STORAGE

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


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

003112  000000                  TEMP1::         .WORD   0               ;SOME TEMP LOCATIONS.
003114  000000                  TEMP2::         .WORD   0
003116  000000                  XXCOMM::        .WORD   0               ;XXDP+ COMM BLOCK POINTER.
003120  000000                  FREE::          .WORD   0               ;1ST FREE MEMORY ADDRESS...
003122  000000                  FRESIZ::        .WORD   0               ;...AND SIZE (IN WORDS).
003124  000000                  FREEHI: .WORD   0               ;LAST WORD IN FREE SPACE
003126  000000                  KTFLG::         .WORD   0               ;KT11, MEM AVAIL FLAG -
                                                                        ;- .WORD        0 = <24K OR NO KT -
                                                                        ;- NZ = >24K AND KT.
003130  000000                  KTENABLE::      .WORD   0               ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
003132  000000                  NXMFLG::        .WORD   0               ;SET IF WE CAN TEST CLEARED OTHERWISE
003134  000000                  NXMLO::         .WORD   0               ;NXM LO ADDRESS BITS
003136  000000                  NXMHI::         .WORD   0               ;NXM HI ADDRESS BITS FOR DAL'S 16-21
003140  000000                  T23A::          .WORD   0               ;11/23A FLAG
003142  000000                  T23B::          .WORD   0               ;11/23B FLAG
003144  000000                  T38FLG::        .WORD   0               ;TEST 38 FLAG ↑O
003146  002000                  PST32W::        .WORD   2000            ;32W BLOCK ADDRESS FOR 32K START
003150  000000                  SIFLAG::        .WORD   0               ;
003152  000000                  BADDAT::        .WORD   0               ;ACTUAL DATA
003154  000000                  GDDAT::         .WORD   0               ;EXPECTED DATA
003156  000000                  LOOPFL::        .WORD   0
003160                          CTAB::                                  ;CONFIGURATION TABLES.
003160  000000                  CTABM::         .WORD   0               ;CONFIG WORK.
003162  000000                                  .WORD   0
003164  000000                                  .WORD   0
003166  000000                                  .WORD   0
003170  177777                                  .WORD   -1              ;END OF MEM TABLE.
003172                          CTABE::
                                ;ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
                                ;
                                ;       0       =       UNIT NOT TESTED
                                ;       100000  =       UNIT ONLINE, NO ERRORS
                                ;       10XXXX  =       UNIT ONLINE, ENCOUNTERED XXXX ERRORS
                                ;       160000  =       UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
                                ;       160001  =       UNIT DROPPED, NOT IDLE AT START
                                ;       14XXXX  =       UNIT DROPPED, ENCOUNTERED XXXX ERRORS
                                ;
003172                          ERTABL:         .BLKW   64.
003372  000000                  ERTABE:         .WORD   0

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

GLOBAL TEXT MESSAGES

```
                                        .SBTTL  GLOBAL TEXT MESSAGES
                                ;++
                                ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
                                ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
                                ; MORE THAN ONE TEST.
                                ;-



                                ;+
                                ;NAMES OF DEVICES SUPPORTED
                                ;-

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

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


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

003476  003536  003541  003545  TSSRBIT::       .WORD   1$,2$,3$,4$,5$,6$,7$,8$
003516  003577  003603  003607                  .WORD   9$,10$,11$,12$,13$,14$,15$,16$
003536     123     103     000  1$:     .ASCIZ  'SC'
003541     102     111     105  2$:     .ASCIZ  'BIE'
003545     123     103     105  3$:     .ASCIZ  'SCE'
003551     122     115     122  4$:     .ASCIZ  'RMR'
003555     116     130     115  5$:     .ASCIZ  'NXM'
003561     116     102     101  6$:     .ASCIZ  'NBA'
003565     102     111     124  7$:     .ASCIZ  'BIT9'
003572     102     111     124  8$:     .ASCIZ  'BIT8'
003577     123     123     122  9$:     .ASCIZ  'SSR'
003603     117     106     114  10$:    .ASCIZ  'OFL'
003607     102     111     124  11$:    .ASCIZ  'BIT5'
003614     102     111     124  12$:    .ASCIZ  'BIT4'
003621     102     111     124  13$:    .ASCIZ  'BIT3'
003626     102     111     124  14$:    .ASCIZ  'BIT2'
003633     102     111     124  15$:    .ASCIZ  'BIT1'
003640     102     111     124  16$:    .ASCIZ  'BIT0'
                                .EVEN
003646     124     123     123  SFIERR: .ASCIZ  'TSSR ERROR AFTER SOFT INIT'
003701     124     123     123  SFHERR: .ASCIZ  'TSSR ERROR AFTER BUS RESET'
003734     040     040     116  NXR:    .ASCIZ  /  NON-EXISTANT DEVICE REGISTER/
003773     045     101     040  NXRX:   .ASCIZ  /%A  ADDRESS: %06/
004014     045     101     040  TSSX:   .ASCII  /%A  TSBA,TSSR EXP'D: %06%A,%06%N/
004054     045     101     040          .ASCIZ  /%A  TSBA,TSSR REC'D: %06%A,%06/
```

GLOBAL TEXT MESSAGES

```
644 004113  045  116  045  FUSI:   .ASCII  /%N%A/
645 004117  040  040  125  USI:    .ASCIZ  /  UNEXPECTED INTERRUPT/
646 004146  040  040  111  NSI:    .ASCIZ  /  INTERRUPT EXPECTED, NOT RECEIVED/
647 004211  045  116  045  FNOINTR:        .ASCII  /%N%A/
648 004215  040  040  116  NOINTR: .ASCIZ  /  NO INTERRUPT WAS GENERATED/
649 004252  040  040  111  IFAULT: .ASCIZ  /  INTERRUPT FAULT/
650 004274  045  101  040  INTX:   .ASCIZ  /%A  CPU PC: %O6%A  TSBA: %O6/
651 004331  040  040  042  NOINIT: .ASCIZ  /  "BUS INIT" DIDN'T INITIALIZE CONTROLLER/
652 004403  040  040  042  NSINIT: .ASCIZ  /  "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
653 004453  040  040  042  BRINIT: .ASCIZ  /  "BUS-RESET" DIDN'T INITIALIZE THE DPU/
654
655 004523  000            NUL:    .ASCIZ  //
656 004524  045  116  000  NULCR:  .ASCIZ  /%N/
657 004527  045  101  040  EXPGOT: .ASCIZ  /%A EXP'D: %O6%A, REC'D: %O6/
658 004563  045  116  045  EXPGT2: .ASCIZ  /%N%A EXP'D: %O6%A, %O6%N%A REC'D: %O%A, %O6/
659 004637  045  101  040  DUAD12: .ASCIZ  /%A REG(W) WRITTEN TO: %O6%A REG(R) READ; EXP'D: %O6%A, REC'D: %O6/
660 004741  122  101  115  PKTRAM::        .ASCIZ  'RAM Contents Do Not Match Packet Sent'
661 005007  040  040  103  SCME:   .ASCIZ  /  CONFIG DOESN'T MATCH MFG. MASTER/
662 005052  127  122  111  WRTMSG: .ASCIZ  'WRITE CHARACTERISTICS Failed'
663 005107  124  123  123  WRTERR: .ASCIZ  'TSSR Incorrect After WRITE Command, More Bits Set Than SSR
664 005202  124  123  123  RDERR:  .ASCIZ  'TSSR Incorrect After READ Command, More Bits Set Than SSR'
665 005274  106  101  124  SCHERR: .ASCIZ  'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc.'
666 005366  105  122  122  RETERR: .ASCIZ  'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
667 005454  045  116  045  NOMEM:  .ASCIZ  '%N%A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. ******%N'
668 005550  045  116  045  M8186:  .ASCIZ  '%N%A ****************** 11/23A SYSTEM ******************%N'
669 005641  045  116  045  M8189:  .ASCIZ  '%N%A ****************** 11/23B SYSTEM ******************%N'
670                          .EVEN
671
672
673
```

GLOBAL ERROR REPORT SECTION

```
                                        .SBTTL  GLOBAL ERROR REPORT SECTION

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

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

                                ;
                                ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
                                ; TO ANY OF THE ABOVE ERROR SIGNATURES.
                                ;
005764  005727                  EXTEND: TST     (PC)+
005766  000000                  EXTA:   0                       ; 0 = NO EXTENSION.
005770  001402                          BEQ     1$
005772  004777  177770                  JSR     PC,@EXTA        ; APPEND EXTENSION TEXT.
005776                          1$:     PRINTX  #NULCR          ; PRINT A BLANK LINE
005776  012746  004524                  MOV     #NULCR,-(SP)
006002  012746  000001                  MOV     #1,-(SP)
006006  010600                          MOV     SP,R0
006010  104415                          TRAP    C$PNTX
006012  062706  000004                  ADD     #4,SP
006016  000207                          RTS     PC
```

```
                                        .SBTTL  PRITSSR - PRINT TSSR CONTENTS

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

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

```
                006164  012746  000002                  MOV     #2,-(SP)
                006170  010600                          MOV     SP,R0
                006172  104415                          TRAP    C$PNTX
                006174  062706  000006                  ADD     #6,SP
          745
          746   006200  010403                  20$:    MOV     R4,R3           ;GET THE TSSR CONTENTS
          747   006202  042703  177761                  BIC     #^CTERCLS,R3    ;CLEAR ALL BUT TERMINATION
          748   006206  016303  006672                  MOV     TCOCOD(R3),R3   ;GET THE TERMINATION CODE MEANING
          749   006212                                  PRINTX  #TCOASC,R3      ;PRINT THE TERMINATION CODE
                006212  010346                          MOV     R3,-(SP)
                006214  012746  006472                  MOV     #TCOASC,-(SP)
                006220  012746  000002                  MOV     #2,-(SP)
                006224  010600                          MOV     SP,R0
                006226  104415                          TRAP    C$PNTX
                006230  062706  000006                  ADD     #6,SP
          750   006234  010403                          MOV     R4,R3           ;TSSR CONTENTS AGAIN
          751   006236  042703  177717                  BIC     #^CFATERR,R3    ;CLEAR ALL BUT FATAL TERMINATION
          752   006242  001416                          BEQ     25$             ;DON'T PRINT IF ZERO
          753   006244  006203                          ASR     R3
          754   006246  006203                          ASR     R3
          755   006250  006203                          ASR     R3              ;ALINE TERMINATION CODE FOR INDEX
          756   006252  016303  007232                  MOV     TSFCOD(R3),R3   ;GET THE FATAL TERMINATION CODE
          757   006256                                  PRINTX  #TFCASC,R3      ;PRINT THE FATAL TERMINATION CODE
                006256  010346                          MOV     R3,-(SP)
                006260  012746  006533                  MOV     #TFCASC,-(SP)
                006264  012746  000002                  MOV     #2,-(SP)
                006270  010600                          MOV     SP,R0
                006272  104415                          TRAP    C$PNTX
                006274  062706  000006                  ADD     #6,SP
          758   006300  042704  176377          25$:    BIC     #^CHIADDR,R4    ;CLEAR ALL BUT EXTENDED ADDRESS
          759   006304  001411                          BEQ     30$             ;DON'T PRINT IF ZERO
          760   006306                                  PRINTX  #TEXASC,R4      ;PRINT THE EXTENDED ADDRESS BITS
                006306  010446                          MOV     R4,-(SP)
                006310  012746  006431                  MOV     #TEXASC,-(SP)
                006314  012746  000002                  MOV     #2,-(SP)
                006320  010600                          MOV     SP,R0
                006322  104415                          TRAP    C$PNTX
                006324  062706  000006                  ADD     #6,SP
          761   006330  013703  002176          30$:    MOV     EPRTSW,R3               ;PRINT MEAASGE BUFFER ADDRESS
          762   006334                                  PRINTX  R3                      ;PRINT PROPER MESSAGE
                006334  010346                          MOV     R3,-(SP)
                006336  012746  000001                  MOV     #1,-(SP)
                006342  010600                          MOV     SP,R0
                006344  104415                          TRAP    C$PNTX
                006346  062706  000004                  ADD     #4,SP
          763   006352  000207                          RTS     PC              ;RETURN TO CALLER
          764
          766   006354                          EPRT2:
          767   006354      045     116     045 EPRT1:  .ASCIZ   '%N%A *****REPLACE M7196*****'
          782   006411      045     116     045 TSSRFOR:        .ASCIZ   '%N%A TSSR = %06'
          783   006431      045     116     045 TEXASC:         .ASCIZ   '%N%A Extended Address Bits = %06'
          784   006472      045     116     045 TCOASC:         .ASCIZ   '%N%A Termination Class Code = %T'
          785   006533      045     116     045 TFCASC:         .ASCIZ   '%N%A Fatal Termination Class Code = %T'
          786   006602      045     116     045 TSSDEF:         .ASCIZ   '%N%A TSSR Bits Set: %T'
          787   006631      045     116     045 AMBTSSR:        .ASCIZ   '%N%A TSSR Contents Are Ambiguous'
          788                                         .EVEN
          789   006672  006712  006735  006763  TCOCOD: .WORD   1$,2$,3$,4$,5$,6$,7$,8$
```

PRITSSR - PRINT TSSR CONTENTS

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

PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

```
                                        .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 ICNORED IF THE KTENABLE FLAG IS CLEAR.
                                ;

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

007512      045     116     045 PKTFRM: .ASCIZ  '%N%A Packet Word #%D1%A = %O6'
007550      045     116     045 PKTADD: .ASCIZ  '%N%A Packet Address = %O1%O5'
                                        .EVEN
```

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

                                ;+
                                ;
                                ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA
                                ;       R2      EXPECTED DATA
                                ;
                                ;OUTPUT:
                                ;
                                ;       R0      XOR OF EXPECTED/RECEIVED DATA
                                ;
                                ;-

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

007670      045     116     045  XORBFOR:          .ASCIZ  '%N%A EXPD: %03%A RECV: %03%A XOR: %03'
                                        .EVEN
```

```
                                        .SBTTL  PRIXOR  - PRINT EXPD, RECV AND XOR

                                ;*
                                ;
                                ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA
                                ;       R2      EXPECTED DATA
                                ;
                                ;OUTPUT:
                                ;
                                ;       R0      XOR OF EXPECTED/RECEIVED DATA
                                ;
                                ;-

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

010006  045  116  045  XORFOR: .ASCIZ  '%N%A EXPD: %06%A RECV: %06%A XOR: %06'
                                .EVEN
```

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

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

010054                          PRIEQU:
010054                                  SAVREG                          ;SAVE THE REGISTERS
010060  000207                          RTS     PC                      ;RETURN TO CALLER


                                        .SBTTL  PRIRAM  - PRINT RAM ADDRESS
                                ;+
                                ;
                                ;PRINT CONTROLLER RAM ADDRESS.
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                                ;
                                ;INPUTS:
                                ;
                                ;       R4      RAM ADDRESS
                                ;
                                ;
010062                          PRIRAM:
010062                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
010066                                  PRINTB  @RAMFOR,R4              ;PRINT RAM ADDRESS IN ERROR
010066  010446                          MOV     R4,-(SP)
010070  012746  010112                  MOV     @RAMFOR,-(SP)
010074  012746  000002                  MOV     #2,-(SP)
010100  010600                          MOV     SP,R0
010102  104414                          TRAP    C$PNTB
010104  062706  000006                  ADD     #6,SP
010110  000207                          RTS     PC                      ;RETURN

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

```
PRIADD    PRINT MEMORY ERROR ADDRESS

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

010224      045     116     045  PRIA0:  .ASCIZ  '%N%A MEMORY ERROR ADDRESS = %01%05'
                                        .EVEN


                                        .SBTTL  PRITADD - PRINT MEMORY TEST ADDRESS
                                        ;+
                                        ;
                                        ;PRINT MEMORY ADDRESS
                                        ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                                        ;
                                        ; IMPLICIT INPUTS
                                        ;
                                        ;       ERRHI   - HIGH ORDER ADDRESS
                                        ;       ERRLO   - LOW ORDER ADDRESS
                                        ;
                                        ;-
010270                                  PRITADD:
010270                                          SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
010274  013702  002234                          MOV     ERRHI,R2        ;GET HIGH ADDRESSS
010300  013701  002236                          MOV     ERRLO,R1        ;GET LOW ADDRESS
                                                ;MOV    R1,R2           ;COPY LOW ADDRESS
                                                ;ROL    R1              ;SHIFT BIT 15 TO C BIT
                                                ;ROL    R0              ;SHIFT INTO HIGH ORDER
010304                                          PRINTB  #PRIT0,R1       ;PRINT MEMORY ADDRESS LOW IN ERROR
010304  010146                                  MOV     R1,-(SP)
010306  012746  010352                          MOV     #PRIT0,-(SP)
010312  012746  000002                          MOV     #2,-(SP)
010316  010600                                  MOV     SP,R0
010320  104414                                  TRAP    C$PNTB
010322  062706  000006                          ADD     #6,SP
010326                                          PRINTB  #PRIT1,R2       ;PRINT MEMORY ADDRESS HIGH IN ERROR
010326  010246                                  MOV     R2,-(SP)
010330  012746  010415                          MOV     #PRIT1,-(SP)
010334  012746  000002                          MOV     #2,-(SP)
010340  010600                                  MOV     SP,R0
010342  104414                                  TRAP    C$PNTB
010344  062706  000006                          ADD     #6,SP
010350  000207                                  RTS     PC              ;RETURN
```

PRITADD - PRINT MEMORY TEST ADDRESS

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

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

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

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

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

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

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

```
                                        .SBTTL  WRTCHR  - WRITE CHARACTERISTICS COMMAND

                                ;+
                                ;
                                ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
                                ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
                                ;
                                ;INPUT:
                                ;
                                ;       R4      ADDRESS OF PACKET FROM TEST
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;       REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
                                ;
                                ;OUTPUT:
                                ;
                                ;       R0      TSSR CONTENTS
                                ;       CARRY   SET - WRITE CHARACTERISTICS COMMAND OK
                                ;               CLR - WRITE CHARACTERISTICS FAILED
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
                                ;       SOFTWARE SWITCHES SET AS FOLLOWS:
                                ;               EXTFEA = EXTENDED FEATURES PRESENT
                                ;               BENBSW = BUFFER ENABLE SWITCH ON OR OFF
                                ;
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;
                                ;-
010652                          WRTCHR::
010652                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
010656  005037  002226                  CLR     BENBSW          ;CLEAR BUFFER ENABLE SWITCH
010662  005037  002224                  CLR     EXTFEA                  ;CLEAR EXTENDED FEATURES SW SWITCH
010666  010465  000000          10$:    MOV     R4,TSDB(R5)             ;SEND OUT COMMAND
010672  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
010676  103401                          BCS     20$                     ;BR, IF SSR IS SET AND OK
010700  000435                          BR      60$                     ;BR IF TROUBLE CARRY = CLEAR
010702  016501  000002          20$:    MOV     TSSR(R5),R1             ;READ TSSR
010706  012702  000200                  MOV     #SSR,R2                 ;SET UP EXPECTED
010712  032701  000100                  BIT     #OFL,R1                 ;WAS OFF LINE SET IN TSSR
010716  001402                          BEQ     25$                     ;BR, IF NO OFL SET
010720  052702  000100                  BIS     #OFL,R2                 ;MAKE THEM LOOK ALIKE
010724  020201          25$:    CMP     R2,R1                   ;ARE THEY OK
010726  001401                          BEQ     40$                     ;BR, IF EQUAL = OK
010730  000421                          BR      60$                     ;TROUBLE EXIT
010732  062704  000010          40$:    ADD     #8.,R4                  ;POINT TO WRT CHARA DATA PACKET
010736  011403                          MOV     (R4),R3                 ;GET ADDRESS OF MESSAGE BUFFER
010740  032763  000200  000012          BIT     #X2.EXTF,XST2(R3)       ;EXTENDED FEATURES BIT SET?
010746  001402                          BEQ     45$                     ;BR IF NO
010750  005237  002224                  INC     EXTFEA                  ;SET EXTENDED FEATURES SW SWITCH
010754                          45$:
010754  032763  000100  000012          BIT     #X2.BUFE,XST2(R3)                       ;BUFFER ENABLE SWITCH SET
010762  001402                          BEQ     50$                     ;BR, IF SWITCH NOT SET
010764  005237  002226                  INC     BENBSW                  ;SET SOFTWARE SWITCH FOR ENABLED
```

WRTCHR WRITE CHARACTERISTICS COMMAND

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

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

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


                        .=<.+10>&177770
011100          RWPACK:
011100  102010                .WORD   102010          ;POSTION COMMAND (REWIND)
011102  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
                                ;
                                ;-

011104                          CKRAM::
011104                                  SAVREG                  ;SAVE THE GENERAL REGISTERS
011110  012701  002240                  MOV     #RAMDATA,R1     ;ADDRESS TO SAVE THE RAM DATA
011114  012702  000201                  MOV     #RMPKTBEG,R2    ;BYTE ADDRESS OF FIRST RAM DATA
011120  005003                          CLR     R3              ;CLEAR THE ERROR FLAG
011122  004737  016326                  JSR     PC,CHKTSSR      ;WAIT FOR SSR
011126  112765  000000  000000          MOVB    #0,TSDB(R5)     ;SET MAINTENANCE MODE
011134  004737  016326          10$:    JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
011140  010265  000000                  MOV     R2,TSDB(R5)     ;SELECT NEXT RAM ADDRESS
011144  004737  016326                  JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
011150  116511  000000                  MOVB    TSBA(R5),(R1)   ;READ THE RAM DATA
011154  122124                          CMPB    (R1)+,(R4)+     ;COMPARE TO EXPECTED
011156  001401                          BEQ     20$             ;BRANCH IF OK
011160  005203                          INC     R3              ;SET ERROR FLAG
011162  005202                  20$:    INC     R2              ;ADDRESS OF NEXT RAM LOCATION
011164  020227  000210                  CMP     R2,#RMPKTEND    ;REACHED END YET ?
011170  003761                          BLE     10$             ;BRANCH TILL ALL READ
011172  005703                          TST     R3              ;WAS AN ERROR FOUND ?
011174  001402                          BEQ     30$             ;BRANCH IF NOT
011176  000241                          CLC                     ;CLEAR CARRY TO SHOW ERROR
011200  000401                          BR      50$             ;AND EXIT
011202  000261                  30$:    SEC                     ;SHOW GOOD COMPARE
011204  012737  000010  002300  50$:    MOV     #8.,RAMSIZ      ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
011212  000207                          RTS     PC              ;RETURN
```

```
                                        .SBTTL  CKRAM2    COMPARE RAM TO I/O CHARACTERISTICS DATA
                                ;+
                                ;
                                ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
                                ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
                                ;
                                ;INPUT:
                                ;
                                ;       R4      ADDRESS OF THE CHARACTERISTICS DATA
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;
                                ;OUTPUT:
                                ;
                                ;       CARRY   SET - RAM MATCHES PACKET
                                ;               CLR - RAM DOES NOT MATCH PACKET
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       THE TABLE RAMDATA IS FILLED WITH THE
                                ;       DATA HELD IN RAM.
                                ;       RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
                                ;
                                ;-

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

H5

CKRAM2 COMPARE RAM TO I/O CHARACTERISTICS DATA SEQ 0059

```
1324 011346  000207                  50$:    RTS     PC                  ;RETURN
1325
```

```
1327                                    .SBTTL  CKMSG   - COMPARE WRITE CHAR. MESSAGE BUFFERS
1328                                ;+
1329                                ;
1330                                ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
1331                                ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1332                                ;ERROR PRINT ROUTINES.
1333                                ;
1334                                ;INPUT:
1335                                ;
1336                                ;       R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1337                                ;       R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
1338                                ;       R2      EXPD MESSAGE BUFFER ADDRESS
1339                                ;OUTPUT:
1340                                ;
1341                                ;       CARRY   SET - MESSAGE BUFFERS MATCH
1342                                ;               CLR -MESSAGE BUFFERS DON'T MATCH
1343                                ;
1344                                ;IMPLICIT OUTPUT:
1345                                ;
1346                                ;       EXPMSG          BUFFER IS SET TO EXPD DATA
1347                                ;       RECMSG          BUFFER IS SET TO RECV DATA
1348                                ;       RCVHIADD        SET TO HIGH ORDER ADDRESS OF RECV
1349                                ;       RCVLOADD        SET TO LOW ORDER ADDRESS OF RECV
1350                                ;
1351                                ;-
1352 011350                        CKMSG::
1353 011350                                SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
1354 011354  010037  002302               MOV     R0,RCVHIADD     ;SAVE RECV HIGH ADDRESS
1355 011360  010137  002304               MOV     R1,RCVLOAD      ;SAVE RECV LOW ADDRESS
1356 011364  005737  003130               TST     KTENABLE        ;TESTING ABOVE 28K?
1357 011370  001403                       BEQ     .0$             ;BR IF NO
1358 011372  004737  017306               JSR     PC,SETMAP       ;RETURN ADDRESS BIASED TO PAR6 IN R0
1359 011376  010001                       MOV     R0,R1           ;GET RETURNED ADDRESS BIASED TO PAR6
1360 011400  005004                10$:   CLR     R4              ;WORD IN BUFFER
1361 011402  005003                       CLR     R3              ;CLEAR ERROR SEEN FLAG
1362 011404  010205                       MOV     R2,R5           ;GET EXPD BUFFER ADDRESS
1363 011406  011264  002320        15$:   MOV     (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
1364 011412  011164  002464               MOV     (R1),RECMSG(R4) ;SAVE RECV FOR ERROR REPORT
1365 011416  022221                       CMP     (R2)+,(R1)+     ;EXPD EQUAL RECV?
1366 011420  001401                       BEQ     25$             ;BR IF YES
1367 011422  005203                       INC     R3              ;SET ERROR SEEN FLAG
1368 011424  062704  000002        25$:   ADD     #2,R4           ;POINT TO NEXT WORD ADDRESS
1369 011430  020427  000014               CMP     R4,#14          ;DONE FIRST 7 WORDS?
1370 011434  003764                       BLE     15$             ;BR IF NO
1371 011436  032765  000200  000012       BIT     #X2.EXTF,XST2(R5);IS EXTENDED FEATURES SET IN EXPD?
1372 011444  001403                       BEQ     50$             ;BR IF NO
1373 011446  020427  000016               CMP     R4,#16          ;DONE EXTENDED FEATURES WORD?
1374 011452  003755                       BLE     15$             ;BR IF NO
1375 011454  005703                50$:   TST     R3              ;ANY ERRORS SEEN?
1376 011456  001402                       BEQ     55$             ;BR IF NO
1377 011460  000241                       CLC                     ;SET FAILURE
1378 011462  000401                       BR      60$             ;
1379 011464  000261                55$:   SEC                     ;SET SUCCESS
1380 011466  000207                60$:   RTS     PC               RETURN
1381
```

```
                                        .SBTTL  CKMSG2  - COMPARE EXPD RECV MESSAGE BUFFERS
                                ;+
                                ;
                                ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
                                ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
                                ;ERROR PRINT ROUTINES.
                                ;
                                ;INPUT:
                                ;
                                ;       R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
                                ;       R2      EXPD MESSAGE BUFFER ADDRESS
                                ;       R3      NUMBER OF BYTES TO COMPARE
                                ;
                                ;OUTPUT:
                                ;
                                ;       CARRY   SET - MESSAGE BUFFERS MATCH
                                ;               CLR - MESSAGE BUFFERS DON'T MATCH
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       EXPMSG          BUFFER IS SET TO EXPD DATA
                                ;       RECMSG          BUFFER IS SET TO RECV DATA
                                ;       RCVHIADD        SET TO HIGH ORDER ADDRESS OF RECV
                                ;       RCVLOADD        SET TO LOW ORDER ADDRESS OF RECV
                                ;
                                ;-
                                CKMSG2::
                                        SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
                                        CMP     R3,#RECMSG-EXPMSG;&&D IS COUNT ABOVE MAX ALLOWED?
                                        BLE     5$                      ;&&D BR IF NO
                                        MOV     #RECMSG-EXPMSG,R3;&&D
                                        PRINTF  #DEBUGMSG               ;&&D
                                        MOV     #DEBUGMSG,-(SP)
                                        MOV     #1,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTF
                                        ADD     #4,SP
                                5$:     MOV     R0,RCVHIADD             ;SAVE RECV HIGH ADDRESS
                                        MOV     R1,RCVLOAD              ;SAVE RECV LOW ADDRESS
                                        TST     KTENABLE                ;TESTING ABOVE 28K?
                                        BEQ     10$                     ;BR IF NO
                                        JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
                                        MOV     R0,R1                   ;GET RETURNED ADDRESS BIASED TO PAR6
                                10$:    CLR     R4                      ;WORD IN BUFFER
                                        CLR     R5                      ;CLEAR ERROR SEEN FLAG
                                15$:    MOVB    (R2),EXPMSG(R4)         ;SAVE EXPD FOR ERROR REPORT
                                        MOVB    (R1),RECMSG(R4)         ;SAVE RECV FOR ERROR REPORT
                                        CMPB    (R2)+,(R1)+             ;EXPD EQUAL RECV?
                                        BEQ     25$                     ;BR IF YES
                                        INC     R5                      ;SET ERROR SEEN FLAG
                                25$:    ADD     #1,R4                   ;POINT TO NEXT BYTE
                                        CMP     R4,R3                   ;DONE ALL BYTES?
                                        BGE     50$                     ;BR IF YES
                                        BR      15$                     ;DO NEXT BYTE
                                50$:    TST     R5                      ;ANY ERRORS SEEN?
                                        BEQ     55$                     ;BR IF NO
```

CKMSG2 COMPARE EXPD RECV MESSAGE BUFFERS

```
1435 011612  000241                          CLC                     ;SET FAILURE
1436 011614  000401                          BR      60$             ;
1437 011616  000261                  55$:    SEC                     ;SET SUCCESS
1438 011620  000207                  60$:    RTS     PC              ;RETURN
1439
1440 011622     120     122     117  DEBUGMSG:       .ASCIZ  'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED ';@@D
1441 011712     045     116     045  FERCM:  .ASCII  /%N%A  %%%/
1442 011723     040     040     124  ERCM:   .ASCIZ  /  TSSR ERROR CODE REC'D = /
1443 011756     056     056     056  SIMSG:  .ASCIZ  /.... AFTER DOING SOFT INIT/
1444 012011     124     105     123  TINERR: .ASCIZ  /TEST: .../
1445                                  .EVEN
```

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

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

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

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

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

                                        BGNMSG  PKTGETS
012054                          PKTGETS::
```

```
1497 012054  004737  006020                    JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1498 012060  012700  000002                    MOV     #2,R0           ;NO. OF WORDS IN GET STATUS PACKET
1499 012064  004737  007364                    JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
1500 012070                                    ENDMSG
     012070                              L10005:
     012070  104423                            TRAP    C$MSG
1501
1502
1503                                    ;+
1504                                    ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
1505                                    ;
1506                                    ;INPUTS:
1507                                    ;
1508                                    ;       R1      TSSR CONTENTS
1509                                    ;       R4      ADDRESS OF COMMAND PACKET
1510                                    ;-
1511
1512 012072                                    BGNMSG  SFFMSG
     012072                             SFFMSG::
1513 012072  004737  006020                    JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1514 012076                                    ENDMSG
     012076                              L10006:
     012076  104423                            TRAP    C$MSG
1515
1516
1517                                           .SBTTL  PKTMES  - PRINT TSSR AND MESSAGE BUFFER
1518                                    ;+
1519                                    ;
1520                                    ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
1521                                    ;BUFFER FOR ERROR REPORTS
1522                                    ;
1523                                    ;INPUTS:
1524                                    ;
1525                                    ;       R1      CONTENTS OF TSSR
1526                                    ;       R2      LOW ORDER MESSAGE BUFFER
1527                                    ;       R3      HIGH ORDER MESSAGE BUFFER ADDRESS
1528                                    ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
1529                                    ;-
1530 012100                                    BGNMSG  PKTMES
     012100                             PKTMES::
1531 012100  004737  006020                    JSR     PC,PRITSSR              ;PRINT CONTENTS OF TSSR
1532 012104  010200                            MOV     R2,R0                   ;LOW ORDER ADDRESS
1533 012106  010301                            MOV     R3,R1                   ;HIGH ORDER ADDRESS
1534 012110  004737  014232                    JSR     PC,PRMESS               ;PRINT THE MESSAGE BUFFER
1535 012114                                    ENDMSG
     012114                              L10007:
     012114  104423                            TRAP    C$MSG
1536
```

```
1538                                            .SBTTL  ADDSSR  - PRINT TEST ADDRESS AND TSSR
1539                                    ;+
1540                                    ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1541                                    ;TSSR AND A MEMORY TEST ADDRESS
1542                                    ;
1543                                    ;INPUTS:
1544                                    ;
1545                                    ;       R5      FIRST DEVICE UNIBUS ADDRESS
1546                                    ;       ERRHI   HIGH ORDER MEMORY TEST ADDRESS
1547                                    ;       ERRLO   LOW ORDER MEMORY TEST ADDRESS
1548                                    ;
1549
1550 012116                                            BGNMSG  ADDSSR
     012116                             ADDSSR::
1551 012116  004737  010270                           JSR     PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
1552 012122  016501  000002                           MOV     TSSR(R5),R1     ;GET CURRENT TSSR
1553 012126  004737  006020                           JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1554 012132                                            ENDMSG
     012132                             L10010:
     012132  104423                                   TRAP    C$MSG
1555
1556
1557                                            .SBTTL  MSGEXP - PRINT WRITE CHAR. EXPD-RECV MESSAGE BUFFERS
1558                                    ;+
1559                                    ;
1560                                    ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
1561                                    ;
1562                                    ;IMPLICIT INPUTS:
1563                                    ;
1564                                    ;       EXPMSG  - EXPECTED MESSAGE BUFFER
1565                                    ;       RECMSG  - RECEIVED MESSAGE BUFFER
1566                                    ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1567                                    ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1568                                    ;-
1569 012134                                            BGNMSG  MSGEXP
     012134                             MSGEXP::
1570 012134  012700  000007                           MOV     #7,R0           ;ASSUME NO EXT FEATURES
1571 012140  005737  002224                           TST     EXTFEA          ;EXT FEATURES SET?
1572 012144  001402                                   BEQ     5$              ;BR IF NO
1573 012146  012700  000010                           MOV     #8.,R0          ;EXT FEATURE BUFFER IS 8 WORDS
1574 012152  004737  014542             5$:           JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
1575 012156                                            ENDMSG
     012156                             L10011:
     012156  104423                                   TRAP    C$MSG
1576
1577
```

```
                                        .SBTTL  FIFEXP  - PRINT FIFO EXP/RECV DATA
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT FIFO EXP/RECV DATA
                                ;
                                ;       R1      - BYTE COUNT
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
                                ;-
                                        BGNMSG  FIFEXP
                                FIFEXP::
                                        PRINTX  #FIF1MSG,R1     ;PRINT BYTES TRANSFERRED
012160  010146                          MOV     R1,-(SP)
012162  012746  012232                  MOV     #FIF1MSG,-(SP)
012166  012746  000002                  MOV     #2,-(SP)
012172  010600                          MOV     SP,R0
012174  104415                          TRAP    C$PNTX
012176  062706  000006                  ADD     #6,SP
                                        PRINTX  #FIF2MSG        ;PRINT HEADER MSG
012202  012746  012301                  MOV     #FIF2MSG,-(SP)
012206  012746  000001                  MOV     #1,-(SP)
012212  010600                          MOV     SP,R0
012214  104415                          TRAP    C$PNTX
012216  062706  000004                  ADD     #4,SP
012222  010100                          MOV     R1,R0           ;GET BYTE COUNT
012224  004737  015112                  JSR     PC,PRBYTEXP     ;PRINT FIFO BYTES IN ERROR
                                        ENDMSG
                                L10012:
                                        TRAP    C$MSG
012232  045  116  045  FIF1MSG:         .ASCIZ  '<CR><LF>A NUMBER OF BYTES TRANSFERRED = %D2'
012301  045  116  045  FIF2MSG:         .ASCIZ  '<CR><LF>A FIFO DATA BYTES IN ERROR:'
                                        .EVEN
```

```
                                        .SBTTL  MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
                                ;
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
012340                                  BGNMSG  MSGSTAT
012340                          MSGSTAT::
012340  012701  012402                   MOV     #STATCOD,R1     ;ASCII ADDRESS TABLE
012344  012100                  10$:     MOV     (R1)+,R0        ;DONE ALL MSG LINES?
012346  001410                           BEQ     20$             ;BR IF YES
012350                                   PRINTX  R0              ;PRINT STATUS BIT NAMES
012350  010046                           MOV     R0,-(SP)
012352  012746  000001                   MOV     #1,-(SP)
012356  010600                           MOV     SP,R0
012360  104415                           TRAP    C$PNTX
012362  062706  000004                   ADD     #4,SP
012366  000766                           BR      10$             ;DO ANOTHER MSG LINE
012370  012700  000012          20$:     MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
012374  004737  014542                   JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012400                                   ENDMSG
012400                          L10013:
012400  104423                           TRAP    C$MSG
                                
012402  012420  012462  012553  STATCOD:        .WORD   1$,2$,3$,4$,5$,6$,0
012420    045     116     045   1$:.ASCIZ  '<NL>A Tape Bus Signals in Word #8:'
012462    045     116     045   2$:.ASCIZ  '<NL>A         PARERR<15>  IEOT  <12>  IFMK  <9>  IRDY<6>  IRWD<2>'
012553    045     116     045   3$:.ASCIZ  '<NL>A         IRESV2<14>  IIDENT<11>  IHER  <8>  IONL<5>  IFBY<1>'
012644    045     116     045   4$:.ASCIZ  '<NL>A         IRESV1<13>  ICER  <10>  ISPEED<7>  ILDP<4>  IFPT<0>'
012735    045     116     045   5$:.ASCIZ  '<NL>A Tape Bus Signals in Word #9:'
012777    045     116     045   6$:.ASCIZ  '<NL>A         DATMIS<7>  ILW<6>  OUTRDY<5>  INRDY<4>'
                                        .EVEN
                                
                                
                                
                                        .SBTTL  MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;
013054                                  BGNMSG  MSGLOOP
013054                          MSGLOOP::
013054  012701  013116                   MOV     #LOOPCOD,R1     ;ASCII ADDRESS TABLE
```

```
MSGLOOP   PRINT LOOPBACK HEADER AND MESSAGE BUFFERS

 1650 013060  012100                  10$:    MOV     (R1)+,R0        ;DONE ALL MSG LINES?
 1651 013062  001410                          BEQ     20$             ;BR IF YES
 1652 013064                                  PRINTX  R0              ;PRINT STATUS BIT NAMES
      013064  010046                          MOV     R0,-(SP)
      013066  012746  000001                  MOV     #1,-(SP)
      013072  010600                          MOV     SP,R0
      013074  104415                          TRAP    C$PNTX
      013076  062706  000004                  ADD     #4,SP
 1653 013102  000766                          BR      10$             ;DO ANOTHER MSG LINE
 1654 013104  012700  000012          20$:    MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
 1655 013110  004737  014542                  JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
 1656 013114                                  ENDMSG
      013114                          L10014:
      013114  104423                          TRAP    C$MSG
 1657
 1658 013116  013136  013211  013310  LOOPCOD:        .WORD   1$,2$,3$,4$,5$,6$,7$,0
 1659 013136     045     116     045  1$:.ASCIZ  '<CR><LF>A Tape Bus Loopback Signals in Word #8:'
 1660 013211     045     116     045  2$:.ASCIZ  '<CR><LF>A      PARERR<15>          IRESV2<14>          IRESV1<13>'
 1661 013310     045     116     045  3$:.ASCIZ  '<CR><LF>A IHISP=>IEOT<12>    IWRT=>IIDENT<11>    IREV  =>ICER  <10>'
 1662 013407     045     116     045  4$:.ASCIZ  '<CR><LF>A IWFM =>IFMK<09>    IEDIT=>IHER <08>    IFAD  =>ISPEED<07>'
 1663 013506     045     116     045  5$:.ASCIZ  '<CR><LF>A ITAD0=>IRDY<06>    ITAD1=>IONL <05>    IERASF=>ILDP  <04>'
 1664 013605     045     116     045  6$:.ASCIZ  '<CR><LF>A IREW =>IDBY<03>    IRWU =>IRWD <02>    IFEN  =>IFBY  <01>'
 1665 013704     045     116     045  7$:.ASCIZ  '<CR><LF>A IGO  =>IFPT<00>'
 1666                                          .EVEN
 1667
```

```
1669                                            .SBTTL  MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
1670                                    ;+
1671                                    ;
1672                                    ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
1673                                    ;
1674                                    ;
1675                                    ;IMPLICIT INPUTS:
1676                                    ;
1677                                    ;       EXPMSG  - EXPECTED MESSAGE BUFFER
1678                                    ;       RECMSG  - RECEIVED MESSAGE BUFFER
1679                                    ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1680                                    ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1681                                    ;-
1682 013732                                         BGNMSG  MSGSUB
     013732                             MSGSUB::
1683 013732  012700  000012                         MOV     #10.,R0         ;SIZE OF WRITE SUBSYSTEM BUFFER
1684 013736  004737  014542                         JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
1685 013742                                         ENDMSG
     013742                             L10015:
     013742  104423                             TRAP    C$MSG
1686
1687
1688
1689
1690
1691                                            .SBTTL  MEMADD - PRINT MEMORY ADDRESS DATA ERROR
1692                                    ;+
1693                                    ;
1694                                    ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
1695                                    ;
1696                                    ;IMPLICIT INPUTS:
1697                                    ;
1698                                    ;       ERRHI   - MEMORY ERROR HIGH ORDER ADDRESS
1699                                    ;       ERRLO   - MEMORY ERROR LOW ORDER ADDRESS
1700                                    ;       EXP     - EXPECTED DATA
1701                                    ;       RECV    - RECEIVED DATA
1702                                    ;-
1703 013744                                         BGNMSG  MEMADD
     013744                             MEMADD::
1704 013744  004737  010154                         JSR     PC,PRIADD       ;PRINT MEMORY ADDRESS IN ERROR
1705 013750  013701  002230                         MOV     EXPD,R1         ;GET EXPD DATA
1706 013754  013702  002232                         MOV     RECV,R2         ;GET RECEIVED DATA
1707 013760  004737  007736                         JSR     PC,PRIXOR       ;PRINT EXPD/RECV
1708 013764                                         ENDMSG
     013764                             L10016:
     013764  104423                             TRAP    C$MSG
1709
```

```
                        .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
                ;-
                
013766          PRAMPKT:
013766                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
013772  012701  002240          MOV     #RAMDATA,R1     ;DATA FROM THE RAM
014000  005002                  CLR     R2              ;INIT BYTE NUMBER
014002  122124          5$:     CMPB    (R1)+,(R4)+     ;COMPARE EXPECTED, RECEIVED
014004  001005                  BNE     7$              ;BR IF NO MATCH
014014                          FORCERROR       7$,NOTSSR
014016  000436                  BR      10$             ;OOD
014022  116105  177777  002232  7$:     MOVB    -1(R1),R5       ;GET RECV RAM DATA
014026  116403  177777  002230          MOVB    -1(R4),R3       ;GET EXPD PACKET DATA
014036                          XOR     R5,R3           ;XOR EXPD/RECV
014042  042703  177400          BIC     #177400,R3      ;LOW BYTE ONLY
014050  116137  177777  002232  MOVB    -1(R1),RECV     ;GET RECEIVED RAM DATA
014056  116437  177777  002230  MOVB    -1(R4),EXPD     ;GET EXPECTED RAM DATA
014056                          PRINTB  #RAMASC,R2,RECV,EXPD,R3
014056  010346                  MOV     R3,-(SP)
014060  013746  002230          MOV     EXPD,-(SP)
014064  013746  002232          MOV     RECV,-(SP)
014070  010246                  MOV     R2,-(SP)
014072  012746  014146          MOV     #RAMASC,-(SP)
014076  012746  000005          MOV     #5,-(SP)
014102  010600                  MOV     SP,R0
014104  104414                  TRAP    C$PNTB
014106  062706  000014          ADD     #14,SP
014112  005202          10$:    INC     R2              ;UPDATE BYTE COUNT
014114  005737  002300          TST     RAMSIZ          ;DEFAULT TO 8.?
014120  001404                  BEQ     15$             ;BR IF YES
014122  020237  002300          CMP     R2,RAMSIZ       ;DONE ALL BYTES?
014126  003724                  BLE     5$              ;BR IF NO
014130  000403                  BR      25$             ;
014132  020227  000010  15$:    CMP     R2,#8.          ;DONE DEFAULT NUMBER OF BYTES?
014136  002720          20$:    BLT     5$              ;BR IF NO
014140  005037  002300  25$:    CLR     RAMSIZ          ;SET DEFAULT RAMSIZ
014144  000207                  RTS     PC              ;RETURN
                
014146  045  116  045  RAMASC: .ASCIZ  '%N%A  BYTE: %D2%A RAM: %03%A Packet: %03%A XOR:%03'
```

PRAMPKT PRINT RAM AND PACKET DATA

```
1759                                            .EVEN
1760
1761
```

```
                        .SBTTL  PRMESS  - PRINT CONTENTS OF MESSAGE BUFFER
                ;*
                ;
                ;THIS ROUTINE PRINTS THE CONTENTS OF
                ;THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE
                ;TSV-05.
                ;
                ;INPUT:
                ;
                ;       R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
                ;       R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
                ;       NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
                ;
                ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
                ;
                ;-

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

```
TSV3   GLOBAL AREAS     MACRO M1200  23 MAR-84 09:44  PAGE 41 1                                  SEQ 0073

PRMESS  - PRINT CONTENTS OF MESSAGE BUFFER

   1801 014374  002761                          BLT     20$                ;PRINT FIRST 7 WORDS
   1802 014376  032763  000200  000012          BIT     #X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
   1803 014404  001355                          BNE     20$                ;PRINT EXTENDED STATUS WORD
   1804 014406  000207                  50$:    RTS     PC                 ;RETURN
   1805
   1806 014410     045     116     045  PROASC: .ASCIZ  '#N#A Message Buffer Address = #01#05'
   1807 014455     045     116     045  PR1ASC: .ASCIZ  '#N#A Message Buffer Contents:'
   1808 014513     045     116     045  PRASC:  .ASCIZ  '#N#A    Word#D1#A: #O'
   1809                                          .EVEN
```

```
                        .SBTTL  PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
                ;+
                ;
                ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
                ;
                ;       R0          NUMBER OF WORDS IN BUFFER
                ;
                ;IMPLICIT INPUTS:
                ;
                ;       EXPMSG   - EXPECTED MESSAGE BUFFER
                ;       RECMSG   - RECEIVED MESSAGE BUFFER
                ;       RCVHIADD - RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                ;       RCVLOADD - RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                ;-
014542          PRMSGEXP::
                        SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
014542 010005           MOV     R0,R5                   ;SAVE NUMBER OF WORDS
014546 013700 002304    MOV     RCVLOADD,R0             ;GET RECV LOW ADDRESS
014550 010004           MOV     R0,R4                   ;COPY LOW ADDRESS
014554 013701 002302    MOV     RCVHIADD,R1             ;GET RECV HIGH ADDRESS
014560 006100           ROL     R0                      ;SHIFT BIT15 TO C BIT
014562 006101           ROL     R1                      ;SHIFT TO HIGH ORDER FOR PRINTOUT
014564                  PRINTX  #PRMSG0,R1,R4           ;PRINT MESSAGE BUFFER ADDRESS
014564 010446           MOV     R4,-(SP)
014566 010146           MOV     R1,-(SP)
014570 012746 014722    MOV     #PRMSG0,-(SP)
014574 012746 000003    MOV     #3,-(SP)
014600 010600           MOV     SP,R0
014602 104415           TRAP    C$PNTX
014604 062706 000010    ADD     #10,SP
014610                  PRINTX  #PRMSG1                 ;PRINT HEADER FOR CONTENTS
014610 012746 014767    MOV     #PRMSG1,-(SP)
014614 012746 000001    MOV     #1,-(SP)
014620 010600           MOV     SP,R0
014622 104415           TRAP    C$PNTX
014624 062706 000004    ADD     #4,SP
014630 005004           CLR     R4                      ;NUMBER OF THE CURRENT WORD
014632 012701 002320    MOV     #EXPMSG,R1              ;GET EXPD BUFFER ADDRESS
014636 012702 002464    MOV     #RECMSG,R2              ;GET RECV BUFFER ADDRESS
014642 011100   20$:     MOV     (R1),R0                 ;GET EXPD
014644 011203           MOV     (R2),R3                 ;GET RECV
014646 040300           XOR     R0,R3                   ;XOR EXPD/RECV
014650                  PRINTX  #PRMSG2,R4,(R1)+,(R2)+,R3
014650 010346           MOV     R3,-(SP)
014652 012246           MOV     (R2)+,-(SP)
014654 012146           MOV     (R1)+,-(SP)
014656 010446           MOV     R4,-(SP)
014660 012746 015025    MOV     #PRMSG2,-(SP)
014664 012746 000005    MOV     #5,-(SP)
014670 010600           MOV     SP,R0
014672 104415           TRAP    C$PNTX
014674 062706 000014    ADD     #14,SP
014700 005204           INC     R4                      ;NUMBER OF THE NEXT
014702 020405           CMP     R4,R5                   ;DONE ALL YET?
014704 002001           BGE     50$                     ;BR IF YES
014706 000752           BR      20$                     ;DO ANOTHER
014710 000207   50$:    RTS     PC                      ;RETURN
```

PRMSGEXP PRINT EXPD/RECV MESSAGE BUFFERS

```
1847
1848 014722  045  116  045  PRMSG0: .ASCIZ  '%N%A Message Buffer Address = %01%05'
1849 014767  045  116  045  PRMSG1: .ASCIZ  '%N%A Message Buffer Contents:'
1850 015025  045  116  045  PRMSG2: .ASCIZ  '%N%A   WORD #%D2%A  EXPD: %06%A  RECV: %06%A  XOR: %06'
1851                                .EVEN
1852
```

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

PRBYTEXP PRINT FRROR BYTES IN EXP/REC MESSAGE BUFFER

```
        015320  062706  000006                  ADD     #6,SP
   1897 015324  000207                          RTS     PC              ;RETURN
   1898
   1899 015326     045     116     045  PRBMSG: .ASCIZ  '<N><A>  BYTE #<D2><A>  EXPD: <03><A>  RECV: <03><A>  XOR: <03>'
   1900 015413     045     116     045  PRBTOT: .ASCIZ  '<N><A> NUMBER OF BYTES IN ERROR = <D2>'
   1901                                  .EVEN
   1902 015460  000000                  PRBEXP: .WORD   0               ;EXPD
   1903 015462  000000                  PRBREC: .WORD   0               ;RECV
   1904
```

,

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

```
                                        .SBTTL  EXPBREC - PRINT EXPD/RECV BYTE DATA
                                ;*
                                ;
                                ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
                                ;
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA BYTE
                                ;       R2      EXPECTED DATA BYTE
                                ;
                                ;

                                        BGNMSG  EXPBREC
                                EXPBREC::
015472  004737  007606                  JSR     PC,PRIBXOR              ;PRINT THE DATA
                                        ENDMSG
                                L10020:
015476  104423                          TRAP    C$MSG



                                        .SBTTL  RAMERR  - PRINT RAM AND PACKET DATA
                                ;*
                                ;
                                ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
                                ;
                                ;INPUTS:
                                ;
                                ;       R4      POINTER TO COMMAND PACKET
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       RAMDATA         DATA AS READ FROM THE RAM
                                ;       RAMSIZ          NUMBER OF BYTES IN PACKET
                                ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
                                ;
                                ;IMPLICIT OUTPUTS:
                                ;
                                ;       RAMSIZ  SET TO 0
                                ;-

                                        BGNMSG  RAMERR
                                RAMERR::
015500  004737  013766                  JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
                                        ENDMSG
                                L10021:
015504  104423                          TRAP    C$MSG


                                        .SBTTL  RAMTADD   PRINT TEST ADDRESS, RAM AND PACKET DATA
                                ;*
                                ;
                                ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
                                ;
                                ;INPUTS:
```

```
                        ;
                        ;       R4      POINTER TO COMMAND PACKET
                        ;
                        ;IMPLICIT INPUTS:
                        ;
                        ;       RAMDATA         DATA AS READ FROM THE RAM
                        ;       RAMSIZ          NUMBER OF BYTES IN PACKET
                        ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
                        ;       ERRHI           HIGH ORDER TEST ADDRESS
                        ;       ERRLO           LOW ORDER TEST ADDRESS
                        ;
                        ;IMPLICIT OUTPUTS:
                        ;
                        ;       RAMSIZ  SET TO 0
                        ;-
                        
                                BGNMSG  RAMTADD
                        RAMTADD::
015506  004737  010270          JSR     PC,PRITADD      ;PRINT TEST ADDRESS
015512  004737  013766          JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
                                ENDMSG
                        L10022:
015516  104423                  TRAP    C$MSG
                        
                                .SBTTL  RAMEXP  - PRINT RAM EXPD/RECV DATA
                        ;*
                        ;
                        ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                        ;
                        ;INPUTS:
                        ;
                        ;       R1      RECEIVED DATA
                        ;       R2      EXPECTED DATA
                        ;       R4      CONTROLLER RAM ADDRESS
                        ;-
                        
                                BGNMSG  RAMEXP
                        RAMEXP::
015520  042701  177400          BIC     #^C<377>,R1             ;SAVE EXPD RAM DATA BYTE
015524  042702  177400          BIC     #^C<377>,R2             ;SAVE EXPD RAM DATA BYTE
015530  004737  010062          JSR     PC,PRIRAM               ;PRINT THE RAM ADDRESS
015534  004737  007736          JSR     PC,PRIXOR               ;PRINT THE DATA
                                ENDMSG
                        L10023:
015540  104423                  TRAP    C$MSG
                        
                                .SBTTL  TIMEXP  - PRINT TIMER A,B AND EXP/REC
                        ;*
                        ;
                        ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                        ;AND TIMER A,B HEADER MESSAGE
                        ;
                        ;INPUTS:
                        ;
                        ;       R1      RECEIVED DATA
                        ;       R2      EXPECTED DATA
```

```
                                        ;
                                        ;
                                        BGNMSG  TIMEXP
                                TIMEXP::
                                        PRINTX  #TIMSGO                 ;PRINT HEADER
015542  012746  015570                  MOV     #TIMSGO,-(SP)
015546  012746  000001                  MOV     #1,-(SP)
015552  010600                          MOV     SP,R0
015554  104415                          TRAP    C$PNTX
015556  062706  000004                  ADD     #4,SP
015562  004737  007736                  JSR     PC,PRIXOR               ;PRINT THE DATA
                                        ENDMSG
                                L10024:
015566  104423                          TRAP    C$MSG

                                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)
                                ;
                                ;-

015670                                  BGNMSG  BADSSR
015670                          BADSSR::
015670  010246                          MOV     R2,-(SP)                ;SAVE DATA TRANSFERRED
015672  042702  177400                  BIC     #177400,R2              ;GET JUST ONE BYTE
015676                                  PRINTB  #XFERASC,R2
015676  010246                          MOV     R2,-(SP)
015700  012746  015730                  MOV     #XFERASC,-(SP)
015704  012746  000002                  MOV     #2, (SP)
015710  010600                          MOV     SP,R0
015712  104414                          TRAP    C$PNTB
015714  062706  000006                  ADD     #6,SP
015720  012602                          MOV     (SP)+,R2                ;RESTORE R2
015722  004737  006020                  JSR     PC,PRITSSR              ;DECODE TSSR CONTENTS
015726                                  ENDMSG
015726                          L10025:
015726  104423                          TRAP    C$MSG
015730    045   116   045     XFERASC:        .ASCIZ  '%N%A Data Transferred = %O3'
```

```
                                .SBTTL  GLOBAL SUBROUTINES SECTION

                        ;**
                        ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
                        ; THAT ARE USED IN MORE THAN ONE TEST.
                        ;
```

```
                                        .SBTTL  SOFINIT - SOFT INITIALIZE OF CONTROLLER

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

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

```
                                        .SBTTL  CHKAMB  - CHECK TSSR FOR AMBIGUITY

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

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

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

                                ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
                                ;
  000200                                IOKCKIN=BIT7    ; DON'T CHECK FOR BAD INTERRUPTS    TEST WILL.
  000001                                IOKSTP=BIT0     ; EXPECT "STOP" INTERRUPT.
                                ;
                                ;INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
016134      000               INTMASK:        .BYTE   0
                                ;INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
016135      000               INTFLAG:        .BYTE   0

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

                                ;SUBROUTINE TO ENABLE INTERRUPTS:
016142  010046                ENAINT: MOV     R0,-(SP)        ;SAVE R0
016144  013700  002206                MOV     IVEC,R0         ;GET POINTER TO VECTORS
016150  012720  016206                MOV     #INTR,(R0)+     ;SET UP INTERRUPT VECTOR
016154  012720  000300                MOV     #PRI06,(R0)+
016160  012600                        MOV     (SP)+,R0        ;RESTORE R0
016162  011646                        MOV     (SP),-(SP)
016164  012766  000000  000002        MOV     #0,2(SP)        ;SET CPU TO LEVEL 0
016172  000002                        RTI

                                ;SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 6)
016174  011646                DSBINT: MOV     (SP),-(SP)
016176  012766  000300  000002        MOV     #PRI06,2(SP)
016204  000002                        RTI
```

```
2195
2196
2197 016206                                    .SBTTL  INTR    - INTERRUPT HANDLERS
     016206
2198 016206  012737  000001  002222  INTR::    BGNSRV  INTR                ;DEFINE INTERRUPT ENTRY
2199 016214  105037  016135                    MOV     #1,INTRECV          ;SET FLAG TO SHOW INTERRUPT RECEIVED
2200 016220  132737  000001  016134            CLRB    INTFLAG             ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2201 016226  001003                            BITB    @IOKSTP,INTMASK     ;EXPECTING STOP INTERRUPT?
2202 016230  152737  000001  016135            BNE     1$                  ;BR IF YES
2203                                           BISB    @IOKSTP,INTFLAG     ;NO. SET THE ERROR FLAG.
2204
2205 016236                                    ;SAVE REGISTERS, MSG BUFFER, ETC.
2206 016236                                    1$:
     016236                                            ENDSRV
     016236  000002                    L10026:
                                               RTI
2207
2208
```

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

```
                                        .SBTTL  CHKTSSR   CHECK TSSR FOR READY

                                ;*
                                ;
                                ;THIS ROUTINE WAITS FOR READY IN THE TSSR
                                ;AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
                                ;
                                ;INPUT:
                                ;
                                ;       R5      ADDRESS OF CSR REGISTERS
                                ;
                                ;OUTPUT:
                                ;
                                ;       R0      CONTENTS OF TSSR
                                ;       CARRY   SET - OKAY
                                ;               CLR - NOT READY AMBIGUOUS, OR SC SET
                                ;
                                ;-

016326                          CHKTSSR:
016326  004737  016240                  JSR     PC,WAITF                ;WAIT FOR READY
016332  103014                          BCC     20$                     ;BRANCH IF TIME OUT
016334  004737  016034                  JSR     PC,CHKAMB               ;TSSR AMBIGOUS?
016340  103006                          BCC     10$                     ;BR IF YES
016342  032700  100000                  BIT     #SC,R0                  ;SPECIAL CONDITION SET?
016346  001405                          BEQ     15$                     ;BR IF NO
016350  032700  074000                  BIT     #<SCE!BIE!RMR!NXM>,R0   ;ANY ERROR BITS SET?
016354  001402                          BEQ     15$                     ;BR IF NO
016356  000241                  10$:    CLC                             ;SET FAILURE
016360  000401                          BR      20$                     ;
016362  000261                  15$:    SEC                             ;SET SUCCESS
016364  000207                  20$:    RTS     PC                      ;RETURN TO CALLER
```

```
                                        .SBTTL  XNXM    - CHECK FOR NONEXISTENT MEMORY
                                ;+
                                ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
                                ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
                                ;               "C" = 0, ALL ADDRESSES OK.
                                ;
                                ;CALL:  MOV ADR1,R1
                                ;       MOV ADR2,R2
                                ;       JSR PC,NXM
                                ;       RETURN          ;TEST "C" AND PROCEED.
                                ;
016366  012737  016420  000004  XNXM:   MOV     @#2$,@#4        ; SET BUSERR VECTOR.
016374  012737  000200  000006          MOV     @#PRI04,@#6
016402  005003                          CLR     R3              ;FLAG.
016404  005711                  1$:     TST     (R1)            ;TEST THE ADDRESS(ES).
                                                                ;IF ANY TRAP, CONTINUE AT 2$.
016406  020102                          CMP     R1,R2           ;OTHERWISE, CONTINUE HERE.
016410  001407                          BEQ     3$              ;BR IF FINISHED (NO NEXM'S).
016412  062701  000002                  ADD     #2,R1           ;SET NEXT ADDRESS...
016416  000772                          BR      1$              ;...AND CONTINUE.

016420  005103                  2$:     COM     R3              ;GOT ONE, SET FLAG...
016422  012716  016430                  MOV     #3$,(SP)
016426  000002                          RTI                     ;...AND DISMISS INTERRUPT...
016430                          3$:     CLRVEC  #4              ;...AND GIVE BACK THE VECTOR.
016430  012700  000004                  MOV     #4,R0
016434  104436                          TRAP    C$CVEC
016436  005703                          TST     R3              ;DID WE CATCH ONE ??
016440  001401                          BEQ     .+4             ;NO, "C" = 0, SKIP NEXT.
016442  000261                          SEC                     ;YES, "C" = 1, (R1) = NEXM ADDR.
016444  000207                          RTS     PC
```

```
                                        .SBTTL  TSTLOOP   CHECK ITERATION COUNT
                                ;+
                                ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
                                ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
                                ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
                                ;
                                ; CALL: LOOPTO  ARG
                                ;
016446                          TSTLOOP::
016446  005737  002166                  TST     NOITS           ; ITERATIONS INHIBITED?
016452  001006                          BNE     1$              ; YES.
016454  005737  002202                  TST     QVP             ; NO.
016460  100403                          BMI     1$              ;LOOPS DISALLOWED IN QUICK PASS.
016462  005337  002214                  DEC     LOOPCNT         ; BUMP LOOP COUNTER.
016466  001002                          BNE     2$
016470  000241                  1$:     CLC                     ;LOOP DISALLOWED, OR DONE.
016472  000401                          BR      3$
016474  000261                  2$:     SEC                     ;LOOP ENABLED.
016476  000207                  3$:     RTS     PC
```

TSTLOOP - CHECK ITERATION COUNT

```
                                        .SBTTL  TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
                                ;+
                                ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
                                ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
                                ; IN THE CURRENT RUN SEQUENCE.
                                ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
                                ;
                                ;INPUT:
                                ;
                                ;       R0      POINTER TO TEST ID ASCIZ STRING
                                ;
                                ;OUTPUT:
                                ;
                                ;       R5      ADDRESS OF FIRST DEVICE REGISTER
                                ;
                                ;IMPLICIT OUTPUTS:
                                ;
                                ;       TSTCNT  UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;       INTERRUPT LEVEL IS RASIED TO LEVEL OF
                                ;       THE DEVICE UNDER TEST
                                ;
                                ;-
016500                          TSTSETUP::
016500  010046                          MOV     R0,-(SP)        ;SAVE THE TEST ID MESSAGE
016502  005037  003150                  CLR     SIFLAG          ; CLEAR "SOFT INIT" FLAG
016506  005037  016746                  CLR     ERRK            ; CLEAR LOCAL ERROR COUNTER.
016512  005037  005766                  CLR     EXTA            ; CLEAR ERROR EXTENSION FLAG.
016516  105037  016134                  CLRB    INTMASK         ; CLEAR INTERRUPT MASK (CHECK ERROR)
016522  013700  002200                  MOV     UNITN,R0        ; GET THE UNIT NUMBER,
016526  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
016530  005737  003110                  TST     NODEV           ; DID STARTUP FIND THE DEVICE?
016534  001430                          BEQ     4$              ; BR IF YES
016536  100010                          BPL     3$              ; BR IF NOT IDLE
016540  052760  160000  003172          BIS     #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
016546                                  ERRDF   1,NXR,NXRERR    ; NO DEVICE HERE -- PRINT IT
016546  104455                          TRAP    C$ERDF
016550  000001                          .WORD   1
016552  003734                          .WORD   NXR
016554  005732                          .WORD   NXRERR
016556  000407                          BR      2$
016560  052760  160001  003172  3$:     BIS     #160001,ERTABL(R0)      ; FLAG ERROR IN THE ERROR TABLE
016566                                  ERRDF   2,NOINIT        ; DEVICE NOT IDLE
016566  104455                          TRAP    C$ERDF
016570  000002                          .WORD   2
016572  004331                          .WORD   NOINIT
016574  000000                          .WORD   0
016576  012737  177777  003106  2$:     MOV     #-1,DUFLG       ; DROP THE UNIT
016604                                  DODU    UNITN
016604  013700  002200                  MOV     UNITN,R0
016610  104451                          TRAP    C$DODU
016612                                  DOCLN                   ; ABORT THE PASS
```

TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS

```
      016612  104444                          TRAP    C$DCLN
 2373 016614  000423                          BR      5$
 2374
 2375 016616                          4$:     RFLAGS  R0              ; GET THE OPERATOR FLAGS.
      016616  104421                          TRAP    C$RFLA
 2376 016620  032700  001000                  BIT     #PNT,R0         ; PRINT THE TEST NUMBERS?
 2377 016624  001412                          BEQ     1$              ; BR IF NO
 2378 016626  011600                          MOV     (SP),R0         ;GET THE ID MESSAGE
 2379 016630                                  PRINTF  #TNAM,R0                ;DISPLAY THE TEST ID
      016630  010046                          MOV     R0,-(SP)
      016632  012746  016674                  MOV     #TNAM,-(SP)
      016636  012746  000002                  MOV     #2,-(SP)
      016642  010600                          MOV     SP,R0
      016644  104417                          TRAP    C$PNTF
      016646  062706  000006                  ADD     #6,SP
 2380 016652  005237  002212          1$:     INC     TSTCNT          ; BUMP TEST COUNTER.
 2381 016656                                  SETPRI  IPRI            ;PRIORITY THAT OF DEVICE
      016656  013700  002210                  MOV     IPRI,R0
      016662  104441                          TRAP    C$SPRI
 2382 016664  005726                  5$:     TST     (SP)+           ;FIX UP THE STACK
 2383 016666  013705  002204                  MOV     CSRADDR,R5      ; ADDRESS OF TSV REGISTERS ON UNIBUS
 2384 016672  000207                          RTS     PC
 2385 016674     045     123     045  TNAM:   .ASCIZ  '%S%T%A Test'
 2386                                          .EVEN
```

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

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

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

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

INCERK - INCREMENT LOCAL ERROR COUNT

```
        017154  000000                          .WORD   0
   2434 017156                                  DODU    UNITN
        017156  013700  002200                  MOV     UNITN,R0
        017162  104451                          TRAP    C$DODU
   2435 017164                                  DOCLN
        017164  104444                          TRAP    C$DCLN
   2436 017166  012600                  2$:     MOV     (SP)+,R0        ; RESTORE R0
   2437 017170  000207                          RTS     PC              ; RETURN TO CALLER
   2438
```

```
2440                                            .SBTTL  CKDROP    CHECK IF UNIT SHOULD BE DROPPED
2441                                    ;+
2442                                    ; CHECK IF UNIT SHOULD BE DROPPED
2443                                    ;-
2444 017172  010046                    CKDROP: MOV     R0,-(SP)
2445 017174                                    FORCERROR       1$,NOTSSR
2446 017204                                    RFLAGS  R0
     017204  104421                            TRAP    C$RFLA
2447 017206  032700  000040                    BIT     #IDU,R0
2448 017212  001010                            BNE     1$
2449 017214  011600                            MOV     (SP),R0
2450 017216  012737  177777  003106            MOV     #-1,DUFLG
2451 017224                                    DODU    UNITN
     017224  013700  002200                    MOV     UNITN,R0
     017230  104451                            TRAP    C$DODU
2452 017232                                    DOCLN                   ;ABORT THE PASS
     017232  104444                            TRAP    C$DCLN
2453 017234  012600                    1$:     MOV     (SP)+,R0
2454 017236  000207                            RTS     PC
2455
2456
2457
2458
2459                                            .SBTTL  CONFIG    DETERMINE CONFIGURATION OF SYSTEM
2460                                    ;
2461                                    ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
2462                                    ;
2463 017240                            CONFIG:
2464 017240  004737  015764                    JSR     PC,SOFINIT
2465 017244  000207                            RTS     PC
2466
2467
2468
```

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


                                        ;
                                        ; SUBROUTINE - DISABLE MEM MGT.
                                        ;
017264  005737  003126          KTOFF:  TST     KTFLG           ; GOT KT11?
017270  001405                          BEQ     1$              ; NO.
017272  000240                          NOP
017274  000240                          NOP
017276  012737  000000  177572          MOV     #0,SR0          ; DISABLE KT.
017304  000207                  1$:     RTS     PC
```

```
                        .SBTTL  SETMAP  - SETUP PAR6 MAPPING

                ;*
                ;
                ;THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
                ;AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
                ;IS RETURNED BIASED TO PAR6.
                ;
                ;INPUTS:
                ;
                ;       R0      HIGH ORDER ADDRESS BITS
                ;       R1      LOW ORDER ADDRESS BITS
                ;
                ;OUTPUTS:
                ;
                ;       R0      OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
                ;       CARRY   SET IF SUCCESS
                ;               CLR IF ERROR
                ;-
017306          SETMAP:
017306                  SAVREG                          ;SAVE R1-R4 UNTIL NEXT RETURN
017312  005737  003126          TST     KTFLG           ;SYSTEM HAVE ABOVE 28K?
017316  001433                  BEQ     10$             ;BR IF NO
017320  010102                  MOV     R1,R2           ;SAVE LOW ORDER BITS
        000006                  .REPT   6
                                ASR     R0              ;CONVERT WORD ADDRESS TO 32W BLOCKS
                                ROR     R1              ;MAKE IT DOUBLE PRECISION
                                .ENDR
017352  042701  000177          BIC     #177,R1         ;ALINE FOR LOWER 4K BOUNDARY
017356  020137  003126          CMP     R1,KTFLG        ;HIGHER THAN EXISTING MEMORY?
017362  103011                  BHIS    10$             ;BR IF YES
017364  010137  172354          MOV     R1,@#KIPAR6     ;SETUP MAPPING REGISTER PAR6
017370  042702  160000          BIC     #160000,R2      ;SETUP DISPLACEMENT IN PAGE
017374  062702  140000          ADD     #140000,R2      ;ADD IN PAR6 BIAS
017400  010200                  MOV     R2,R0           ;RETURN IN R0
017402  000261                  SEC                     ;SET SUCCESS
017404  000401                  BR      15$             ;
017406  000241          10$:    CLC                     ;SET FAILURE
017410  000207          15$:    RTS     PC              ;RETURN
```

```
                                        .SBTTL  FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN
                                ;*
                                ;  FILL MEMORY WITH A BACKGROUND PATTERN
                                ;
                                ; INPUTS:
                                ;
                                ;       R0 = BACKGROUND PATTERN
                                ;       FREE    = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
                                ;       KTFLG   = SET TO HIGHEST MEMORY LOCATION IF > 28K.
                                ;
                                ; OUTPUTS:
                                ;       NONE
                                ;-
017412                          FILLMEM:
017412                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
017416  004737  017264                  JSR     PC,KTOFF        ;DISABLE KT.
017422  010003                          MOV     R0,R3           ;COPY TEST PATTERN
017424  013701  003120                  MOV     FREE,R1         ;GET FIRST FREF LOCATION
017430  013702  003122                  MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
017434  010321                  10$:    MOV     R3,(R1)+        ;STORE A BACKGROUND WORD
017436  005302                          DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
017440  003375                          BGT     10$             ;BR IF NO
017442  005737  003126                  TST     KTFLG           ; GOT KT?
017446  001477                          BEQ     55$             ; NO. GET OUT.
017450  004737  017246                  JSR     PC,KTON         ; YES. ENABLE KT.
017454  005000                          CLR     R0              ;HIGH ORDER ADDRESS START
017456  013701  003146                  MOV     PST32W,R1       ;GET >28K START ADDRESS (IN 32W BLOCKS)
        000006                          .REPT   6
                                        CLC                     ;CLEAR C BIT
                                        ROL     R1              ;CONVERT BLOCKS TO WORDS
                                        ROL     R0              ;MAKE IT DOUBLE PRECISION
                                        .ENDR
017526  004737  017306                  JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
017532  010320                  30$:    MOV     R3,(R0)+        ;STORE TEST PATTERN IN >28K ADDRESS
017534  020027  160000                  CMP     R0,#160000      ;END OF PAR6 MAPPING AREA?
017540  103774                          BLO     30$             ;BR IF NO
017542  162700  020000                  SUB     #20000,R0       ;BACKUP INTO PAR6 MAPPING BEGIN
017546  062737  000200  172354          ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
017554  023737  172354  003126          CMP     @#KIPAR6,KTFLG  ;END OF MEMORY?
017562  001427                          BEQ     50$             ;BR IF YES
017564  005737  003140                  TST     T23A            ;11/23A?
017570  001407                          BEQ     35$             ;NO KEEP GOING
017572  013704  177572                  MOV     SR0,R4          ;GET SR0 CONTENTS
017576  042704  177761                  BIC     #1^7761,R4      ;CLEAR ALL BUT PAGE NUMBER
017602  022704  000016                  CMP     #16,R4          ;SEE IF PAGE 7
017606  001415                          BEQ     50$             ;EXIT IF THERE
017610  005737  003142          35$:    TST     T23B            ;11/23B?
017614  001410                          BEQ     45$             ;NO KEEP GOING
017616  023727  172354  007600          CMP     @#KIPAR6,#7600  ;REACHED 18 BITS?
017624  103001                          BHIS    40$             ;YES
017626  000403                          BR      45$             ;NO KEEP GOING
017630  012737  000020  172516  40$:    MOV     #20,SR3         ;SET 22 BIT RELOCATION
017636  000137  017532          45$:    JMP     30$             ;KEEP GOING ON ETC.
017642  004737  017264          50$:    JSR     PC,KTOFF        ; DISABLE KT.
017646  000207                  55$:    RTS     PC
```

```
                        .SBTTL  CMPMEM  - COMPARE MEMORY TO BACKGROUND PATTERN
                ;*
                ; COMPARE MEMORY WITH A BACKGROUND PATTERN
                ;
                ; INPUTS:
                ;
                ;       R0 = BACKGROUND PATTERN
                ;       FREE    = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
                ;       KTFLG   = SET TO HIGHEST MEMORY LOCATION IF > 28K.
                ;
                ; OUTPUTS:
                ;
                ;       CARRY   - SET IF NO ERROR
                ;       CARRY   - CLR IF ERROR
                ;
                ; IMPLICIT OUTPUTS:
                ;
                ;       ERRHI   - ERROR HIGH ADDRESS
                ;       ERRLO   - ERROR LOW ADDRESS
                ;       EXPD    - EXPECTED DATA
                ;       RECV    - RECEIVED DATA
                ;-
017650          CMPMEM:
017650                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
017654  010003          MOV     R0,R3           ;COPY TEST PATTERN
017656  004737  017264  JSR     PC,KTOFF        ;DISABLE KT.
017662  013701  003120  MOV     FREE,R1         ;GET FIRST FREE LOCATION
017666  013702  003122  MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
017672  020311          10$:    CMP     R3,(R1)         ;FREE SPACE LOCATION EQUAL TO EXPD?
017674  001411          BEQ     15$             ;BR IF YES
017676  010137  002236  MOV     R1,ERRLO        ;SAVE ADDRESS IN ERROR
017702  005037  002234  CLR     ERRHI           ;NO HIGH ADDRESS
017706  010337  002230  MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
017712  011137  002232  MOV     (R1),RECV       ;SAVE RECV FOR ERROR REPORT
017716  000474          BR      50$             ;
017720  005721          15$:    TST     (R1)+           ;POINT TO NEXT ADDRESS
017722  005302          DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
017724  003362          BGT     10$             ;BR IF NO
017726  005737  003126  TST     KTFLG           ; GOT KT?
017732  001472          BEQ     55$             ; NO. GET OUT.
017734  004737  017246  JSR     PC,KTON         ; YES. ENABLE KT.
017740  005000          CLR     R0              ;HIGH ORDER ADDRESS START
017742  013701  003146  MOV     PST32W,R1       ;GET >28K START ADDRESS (IN 32W BLOCKS)
                000006          .REPT   6
                                ROL     R1              ;CONVERT BLOCKS TO WORDS
                                ROL     R0              ;MAKE IT DOUBLE PRECISION
                                .ENDR
017776  042701  000177  BIC     #177,R1         ;ALINE 4K BOUNDARY
020002  010046          MOV     R0,-(SP)        ;SAVE HIGH ORDER
020004  010146          MOV     R1,-(SP)        ;SAVE LOW ORDER
020006  004737  017306  JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
020012  010004          MOV     R0,R4           ;COPY ADDRESS BIASED TO PAR6
020014  012601          MOV     (SP)+,R1        ;RESTORE LOW ORDER IN NON PAR6 FORMAT
020016  012600          MOV     (SP)+,R0        ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
020020  020314          30$:    CMP     R3,(R4)         ;ABOVE 28K LOCATION EQUAL EXPD?
020022  001411          BEQ     32$             ;BR IF YES
020024  010037  002234  MOV     R0,ERRHI        ;SAVE HIGH ORDER IN ERROR
```

CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN

```
2648 020030  010137  002236                    MOV     R1,ERRLO        ;SAVE LOW ORDER IN ERROR
2649 020034  010337  002230                    MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
2650 020040  011437  002232                    MOV     (R4),RECV       ;SAVE RECV FOR ERROR REPORT
2651 020044  000421                            BR      50$             ;
2652 020046  062701  000002            32$:    ADD     #2,R1           ;UPDATE NON PAR6 ADDRESS
2653 020052  005500                            ADC     R0              ;MAKE IT DOUBLE PRECISION ADD
2654 020054  062704  000002                    ADD     #2,R4           ;UPDATE PAR FORMAT ADDRESS
2655 020060  020427  160000                    CMP     R4,#160000      ;END OF PAR6 MAPPING AREA?
2656 020064  103755                            BLO     30$             ;BR IF NO
2657 020066  162704  020000                    SUB     #20000,R4       ;BACKUP INTO PAR6 MAPPING BEGIN
2658 020072  062737  000200  172354            ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
2659 020100  023737  172354  003126            CMP     @#KIPAR6,KTFLG  ;END OF MEMORY?
2660 020106  101744                            BLOS    30$             ;BR IF NO
2661 020110  004737  017264            50$:    JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
2662 020114  000241                            CLC                     ;SET FAILURE
2663 020116  000403                            BR      60$             ;
2664 020120  004737  017264            55$:    JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
2665 020124  000261                            SEC                     ;SET SUCCESS
2666 020126  000207            60$:    RTS     PC
2667
```

```
                                        .SBTTL  REGSAV  - SAVE R1 R5 ON STACK
                                ;+
                                ;
                                ;ROUTINE TO
                                ;SAVE R1 THROUGH R5 ON THE STACK
                                ;
                                ;CALLING SEQUENCE:
                                ;
                                ;       JSR     R5,REGSAV
                                ;
                                ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
                                ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
                                ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
                                ;REGISTERS.
                                ;
                                ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
                                ;CALLED VIA A JSR PC INSTRUCTION
                                ;
                                ;

020130                          REGSAV:
020130  010446                          MOV     R4,-(SP)
020132  010346                          MOV     R3,-(SP)
020134  010246                          MOV     R2,-(SP)
020136  010146                          MOV     R1,-(SP)
020140  010546                          MOV     R5,-(SP)
020142  016605  000012                  MOV     10.(SP),R5
020146  004736                          JSR     PC,@(SP)+
020150  012601                          MOV     (SP)+,R1
020152  012602                          MOV     (SP)+,R2
020154  012603                          MOV     (SP)+,R3
020156  012604                          MOV     (SP)+,R4
020160  012605                          MOV     (SP)+,R5
020162  000207                          RTS     PC
```

```
                                        .SBTTL  GETPAT  - GET 8 BIT PATTERN FROM OPERATOR
                                ;+
                                ;
                                ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
                                ;
                                ;INPUTS:
                                ;
                                ;       NONE.
                                ;
                                ;OUTPUTS:
                                ;
                                ;       R0      OCTAL NUMBER FROM THE OPERATOR
                                ;
                                ;CALLING SEQUENCE:
                                ;
                                ;       JSR     PC,GETPAT
                                ;
                                ;-

020164                          GETPAT::
                                        SAVREG                          ;SAVE THE GENERAL REGISTERS
020170                          1$:     GMANID  DATASC,PATDAT,0,377,0,377,NO
020170  104443                          TRAP    C$GMAN
020172  000406                          BR      10000$
020174  020220                          .WORD   PATDAT
020176  000022                          .WORD   T$CODE
020200  020222                          .WORD   DATASC
020202  000377                          .WORD   377
020204  000000                          .WORD   T$LOLIM
020206  000377                          .WORD   T$HILIM
020210                          10000$:
020210                                  BNCOMPLETE      1$      ;RETRY IF ERROR
020210  103367                          BCC     1$
020212  013700  020220                  MOV     PATDAT,R0       ;DATA PATTERN FROM OPERATOR
020216  000207                          RTS     PC              ;RETURN TO CALLER

                                ;+
                                ;LOCAL DATA AREA
                                ;-

020220  000000                  PATDAT: .WORD   0               ;TEMPORARY STORAGE FOR DATA
020222     105     116     124  DATASC: .ASCIZ  'ENTER DATA PATTERN'
                                        .EVEN
```

```
                                        .SBTTL  GETSEL  - ISSUE MENU AND GET OPERATOR RESPONSE
                                ;+
                                ;
                                ;ROUTINE TO ISSUE A MENU AND GET
                                ;THE OPERATOR'S RESPONSE.
                                ;
                                ;INPUTS:
                                ;
                                ;       R0      ADDRESS OF ASCIZ STRING OF MENU
                                ;       R1      MAXIMUM ALLOWABLE OPERATOR RESPONSE
                                ;
                                ;OUTPUTS:
                                ;
                                ;       R0      NUMBER OF THE OPERATOR'S SELECTION
                                ;
                                ;-
                                GETSEL::
020246                                  SAVREG                          ;SAVE GENERAL REGISTERS
020246  010002                          MOV     R0,R2                   ;SAVE THE MENU ADDRESS
020252  010203                  1$:     MOV     R2,R3                   ;START OF MENU STRING
020254  005713                  2$:     TST     (R3)                    ;END OF ASCII ?
020256  001412                          BEQ     3$                      ;BRANCH IF ALL LINES DISPLAYED
020260                                  PRINTF  @SELASC,(R3)+           ;DISPLAY THE MENU
020260  012346                          MOV     (R3)+,-(SP)
020262  012746  020432                  MOV     @SELASC,-(SP)
020266  012746  000002                  MOV     #2,-(SP)
020272  010600                          MOV     SP,R0
020274  104417                          TRAP    C$PNTF
020276  062706  000006                  ADD     #6,SP
020302  000764                          BR      2$
020304                          3$:     GMANID  MENASC,MENRES,D,-1,0, 1,NO
020304  104443                          TRAP    C$GMAN
020306  000406                          BR      10001$
020310  020466                          .WORD   MENRES
020312  000042                          .WORD   T$CODE
020314  020437                          .WORD   MENASC
020316  177777                          .WORD   -1
020320  000000                          .WORD   T$LOLIM
020322  177777                          .WORD   T$HILIM
020324                          10001$:
020324                                  BNCOMPLETE      1$              ;RETRY IF ERROR
020324  103352                          BCC     1$
020326  013700  020466                  MOV     MENRES,R0               ;GET THE OPERATOR'S REPLY
020332  020001                          CMP     R0,R1                   ;COMPARE TO MAXIMUM ALLOWED
020334  101411                          BLOS    5$                      ;BRANCH IF OK
020336                                  PRINTF  @MENERR                 ;DISPLAY ERROR MESSAGE
020336  012746  020362                  MOV     @MENERR,-(SP)
020342  012746  000001                  MOV     #1,-(SP)
020346  010600                          MOV     SP,R0
020350  104417                          TRAP    C$PNTF
020352  062706  000004                  ADD     #4,SP
020356  000735                          BR      1$                      ;RETRY
020360  000207                  5$:     RTS     PC                      ;RETURN TO CALLER
020362     045     116     045  MENERR: .ASCIZ  '%N%A *** Menu Selection Too Large ***'
020430     045     116     045  SELASC: .ASCIZ  '%N%T'
020435     105     156     164  MENASC: .ASCIZ  'Enter Menu Selection: '
```

```
   2775
   2776 020466  000000                    MENRES:  .EVEN
                                                  .WORD   0
```

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

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

020524      045     116     045 NOMAN:  .ASCIZ  '%N%A *** Manual Intervent'on not Allowed - Test Aborted ***'
                                        .even
```

ENVIRN - SETUP FREE DIAGNOSTIC SPACE

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

```
                                        .SBTTL  KTINIT  - SETUP KT11 MEMORY MANAGEMENT REGISTERS
                                ;+
                                ;
                                ;ROUTINE TO INIT KT-11
                                ;
                                ;-
021044                          KTINIT:
021044  005037  003126                CLR     KTFLG           ; INIT >28K MEMORY FLAG
021050  005037  003130                CLR     KTENABLE        ; INIT TEST >28K FLAG
021054  023727  002120  001577        CMP     L$HIME,#1577    ; GOT ENOUGH MEMORY (>28K)?
021062  101444                        BLOS    9$              ; NO.
021064  013700  000004                MOV     @#ERRVEC,R0     ; SAVE OLD ERR VEC PTR.
021070  012737  021162  000004        MOV     #2$,@#ERRVEC    ; SET ERR VEC PTR.
021076  005737  177572                TST     @#SR0           ; GOT KT11?
021102  000240                        NOP                     ;  (TRAP IF NO).
021104  013737  002120  003126        MOV     L$HIME,KTFLG    ; YES. SET KT FLAG.
021112  042737  000177  003126        BIC     #177,KTFLG      ;
021120  010037  000004                MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
021124  005000                        CLR     R0              ; R0 = AR DATA.
021126  012701  172340                MOV     #KIPAR0,R1      ; R1 = KI REGS PTR.
021132  012761  077406  177740  1$:   MOV     #77406,-40(R1)  ; SET DESCRIPTOR REG.
021140  010021                        MOV     R0,(R1)+        ; SET KIPAR REG.
021142  062700  000200                ADD     #200,R0         ; BUMP AR DATA BY "4K".
021146  020027  002000                CMP     R0,#2000        ; AT "I/O"?
021152  001367                        BNE     1$              ; NO.
021154  012741  177600                MOV     #177600,-(R1)   ; YES. SET KTPAR7 FOR I/O.
021160  000405                        BR      9$

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

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

021174  000207                9$:     RTS     PC
```

KTINIT - SETUP KT11 MEMORY MANAGEMENT REGISTERS

```
                                        ;*
                                        ;       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.
                                        ;
                                        ;-

021176                                  INVERT::

021176  005737  002224                          TST     EXTFEA                  ; IS SWITCH SET?
021202  001020                                  BNE     1$                      ; YES,EXIT STAGE RIGHT!(or the next one outa town!)
021204  012737  100206  021250                  MOV     #100206,CMDPKT          ; WRT SUB-SYS MEM CMD
021212  012737  021260  021252                  MOV     #WSMBK,CMDPKT+2         ; MSG BUF ADDR
021220  012737  000006  021256                  MOV     #6,CMDPKT+6             ; BYTE COUNT
021226  012737  100010  021260                  MOV     #100010,WSMBK           ; INVERT THE SWITCH
021234  012704  021250                          MOV     #CMDPKT,R4              ; SET CMDPKT INTO R4
021240  004737  010652                          JSR     PC,WRTCHR               ; DO IT
021244  000207                          1$:     RTS     PC                      ; RETURN

                                        ;       COMMAND PACKET.

        021250                          .       =       <.+3>&177774    ;MUST BE ON MOD 4 BOUNDRY.

021250  000000                          CMDPKT:: 0                      ;1ST WORD IS TSO5 COMMAND.
021252  000000                                   0                      ;2ND WORD IS THE BUFFER LOW ADDRESS.
021254  000000                                   0                      ;3RD WORD IS THE BUFFER HIGH ADDRESS.
021256  000000                                   0                      ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.

                                        ;       WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.

021260  000000                          WSMBK:: 0                       ;1ST WORD:: SEL 0
021262  000000                                  0                       ;2ND WORD:: SEL 2
021264  000000                                  0                       ;3RD WORD:: SEL 4
                                                .EVEN
```

```
2934                                    ;*
2935                                    ;       SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
2936                                    ;
2937                                    ;
2938                                    ;INPUTS:
2939                                    ;OUTPUTS:
2940                                    ;       The NXMFLG is set if we can test.
2941                                    ;       The NXMLO and NXMHI addresses are setup.
2942                                    ;-
2943
2944 021266                             MEMCK::
2945
2946 021266                                     SAVREG                          ;SAVE THE REGISTERS
2947 021272  005037  003132                     CLR     NXMFLG                  ;CLEAR THE FLAG
2948 021276  005037  003134                     CLR     NXMLO                   ;CLEAR THE TEST ADDRESS LO
2949 021302  005037  003136                     CLR     NXMHI                   ;CLEAR THE TEST ADDRESS HI
2950 021306  005737  003142                     TST     T23B                    ;IS IT A 11/23B?
2951 021312  001407                           BEQ     1$                      ;NO
2952 021314  023727  002120  007777           CMP     L$HIME,#7777            ; GREATER THAN 128K
2953 021322  103406                           BLO     2$                      ; NO
2954 021324  004737  021442                   JSR     PC,NXMTST               ;SETUP THE ADDRESS
2955 021330  000427                           BR      13$                     ;SET THE FLAG AND EXIT
2956 021332  005737  003140           1$:     TST     T23A                    ;IS IT A 11/23A?
2957 021336  001413                           BEQ     4$                      ;NO
2958 021340  023727  002120  005777   2$:     CMP     L$HIME,#5777            ;GREATER THAN 96K
2959 021346  101023                           BHI     14$                     ;YES,23A/23B WITH 128K MEMORY
2960 021350  023727  002120  003777           CMP     L$HIME,#3777            ;GREATER THAN 64K BUT LESS THAN 92K?
2961 021356  103403                           BLO     4$                      ;NO, CHECK 24K
2962 021360  004737  021442                   JSR     PC,NXMTST               ;SETUP THE ADDRESS
2963 021364  000411                           BR      13$                     ;SET THE FLAG AND EXIT
2964 021366  023727  002120  001577   4$:     CMP     L$HIME,#1577            ;GREATER THAN 24K BUT LESS THAN 64K?
2965 021374  103410                           BLO     14$                     ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
2966 021376  004737  021442                   JSR     PC,NXMTST               ;SETUP THE ADDRESS
2967 021402  062737  000077  003136           ADD     #77,NXMHI               ;FOOL THE 11/02 & 11/03
2968 021410  005237  003132           13$:    INC     NXMFLG                  ;SET THE FLAG
2969 021414  000411                           BR      15$                     ;EXIT
2970 021416  000410                   14$:    BR      15$                     ;NOP FOR PRINTOUT
2971 021420                                     PRINTF  #NOMEM                  ;TELL THEM & EXIT ***NO PRINT*****
     021420  012746  005454                     MOV     #NOMEM,-(SP)
     021424  012746  000001                     MOV     #1,-(SP)
     021430  010600                             MOV     SP,R0
     021432  104417                             TRAP    C$PNTF
     021434  062706  000004                     ADD     #4,SP
2972 021440  000207                   15$:    RTS     PC                      ;RETURN
2973
2974
2975                                    ;*
2976                                    ;       SUBROUTINE TO SETUP THE NXM ADDRESS FOR TESTING
2977                                    ;
2978                                    ;OUTPUTS:NXMLO,NXMHI            ;SETUP WITH NXM ADDRESS
2979                                    ;
2980                                    ;
2981
2982 021442  013701  002120           NXMTST: MOV     L$HIME,R1               ;GET TOP OF MEMORY
2983 021446  062701  000200                   ADD     #200,R1                 ;MAKE IT I/O BLOCK OR OTHER NXM
2984 021452  042701  000177                   BIC     #177,R1
2985 021456  01010c                           MOV     R1,R2                   ;RESAVE RESULTS
```

KTINIT SETUP KT11 MEMORY MANAGEMENT REGISTERS SEQ 0110

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


021536                                  ENDMOD
```

```
                                                .TITLE  TSV4 - MISCELLANEOUS SECTIONS
                                                BGNMOD  TSV4
021536                                  TSV4::
021536
```

PROTECTION TABLE

```
                                                .SBTTL  PROTECTION TABLE
                                                BGNPROT
                                        L$PROT::
                                                .WORD   1.  1.  1.  1           ;NO DEVICE PROTECTION REQUIRED.
                                                ENDPROT
```

INITIALIZE SECTION

```
                                        .SBTTL  INITIALIZE SECTION

                                ;++
                                ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ;AT THE BEGINNING OF EACH PASS.
                                ;
                                ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS INIT.
                                ;IF "CONTINUE", NOTHING IS REQUIRED.
                                ;
                                ;--
                                ;+
                                ;INSERT TEMPORARY JUMP TO ODT
                                ;-
021546                                  BGNINIT
021546                          L$INIT::

021546                                  SETVEC  #140,#170000,#340       ;ODT ROM ADDRESS                ;JB REV A 0
021546  012746  000340                  MOV     #340,-(SP)
021552  012746  170000                  MOV     #170000,-(SP)
021556  012746  000140                  MOV     #140,-(SP)
021562  012746  000003                  MOV     #3,-(SP)
021566  104437                          TRAP    C$SVEC
021570  062706  000010                  ADD     #10,SP

021574  005037  002224          40$:    CLR     EXTFEA
021600  005037  003132                  CLR     NXMFLG
021604  012737  006354  002176          MOV     #EPRT1,EPRTSW           ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
021612  005037  003150                  CLR     SIFLAG                  ;CLEAR "SOFT INIT" FLAG
021616  005037  003130                  CLR     KTENABLE                ;CLEAR TEST ABOVE 28K FLAG
021622  005037  002300                  CLR     RAMSIZ                  ;CLEAR RAM SIZE FOR RAMERR ROUTINE
021626                                  READEF  #EF.CONTINUE
021626  012700  000036                  MOV     #EF.CONTINUE,R0
021632  104447                          TRAP    C$REFG
021634                                  BNCOMPLETE 1$
021634  103023                          BCC     1$
021636  023737  002200  002012          CMP     UNITN,L$UNIT            ;UNIT IN RANGE?
021644  103070                          BHIS    4$                      ;BR IF NO.
021646  005737  003106                  TST     DUFLG                   ;DROPPED UNIT?
021652  100472                          BMI     NXTU                    ;BR IF YES
021654  013701  002200                  MOV     UNITN,R1
021660  006301                          ASL     R1
021662  005761  003172                  TST     ERTABL(R1)
021666  001516                          BEQ     SETU
021670  032761  040000  003172          BIT     #BIT14,ERTABL(R1)       ;DROPPED?
021676  001060                          BNE     NXTU
021700                                  EXIT    INIT                    ;DO NOTHING IF "CONTINUE".
021700  104432                          TRAP    C$EXIT
021702  000416                          .WORD   L10030-.
021704                          1$:     READEF  #EF.NEW
021704  012700  000035                  MOV     #EF.NEW,R0
021710  104447                          TRAP    C$REFG
021712                                  BNCOMPLETE NXTU                 ;TAKE NEXT UNIT IF NOT NEW PASS.
021712  103052                          BCC     NXTU
021714                                  READEF  #EF.START
021714  012700  000040                  MOV     #EF.START,R0
021720  104447                          TRAP    C$REFG
021722                                  BCOMPLETE 2$
```

INITIALIZE SECTION

```
                021722  103404                          BCS     2$
     64 021724                                          READEF  #EF.RESTART
                021724  012700  000037                  MOV     #EF.RESTART,R0
                021730  104447                          TRAP    C$REFG
     65 021732                                          BNCOMPLETE 31$
                021732  103031                          BCC     31$
     66 021734                          2$:                                  ;1ST PASS, BUS-INIT...
     67 021734                                          BRESET                ;BUS RESET.
                021734  104433                          TRAP    C$RESET
     68 021736  005037  002212                          CLR     TSTCNT        ;NUMBER OF TESTS RUN IN PASS
     69 021742  005037  002220                          CLR     FATFLG        ;CLEAR FATAL ERROR COUNT
     70 021746  005037  003140                          CLR     T23A          ;CLEAR 11/23A FLAG
     71 021752  005037  003142                          CLR     T23B          ;CLEAR 11/23B FLAG
     72                                     ;           MOV     #340,-(SP)
     73                                     ;           MOV     #20$,-(SP)    ;RETURN TO DEBUGGER
     74                                     ;           JMP     0.ODT       ;;@@ENTER THE DEBUGGER
     75 021756  005037  003374                          CLR     SKIPT       ;CLEAR THE SUBTEST "SKIPPER"
     76 021762                          20$:
     77 021762  012737  177777  002202                  MOV     #-1,QVP               ;...QUICK VERIFY...
     78 021770  004737  020620                          JSR     PC,ENVIRN             ;SET ENVIRONMENT.
     79 021774  004737  021044                          JSR     PC,KTINIT             ;INITIALIZE KT MEMORY MANAGEMENT
     80 022000  012700  003172                          MOV     #ERTABL,R0
     81 022004  005020                          30$:    CLR     (R0)+                 ;CLEAR THE ERROR TABLE
     82 022006  020027  003372                          CMP     R0,#ERTABE
     83 022012  103774                                  BLO     30$
     84 022014  000404                                  BR      4$
     85 022016  005037  002202                  31$:    CLR     QVP
     86 022022  000137  022072                          JMP     PASRPT                ;GO REPORT THE STATUS
     87
     88 022026                          4$:
     89 022026  012737  177777  002200  NEWPAS: MOV     #-1,UNITN             ;INIT UNIT NUMBER...
     90 022034  005037  002216                          CLR     DEVCNT        ;CLEAR COUNT OF DEVICES RUNNING
     91 022040                          NXTU:   BREAK
                022040  104422                          TRAP    C$BRK
     92 022042  005237  002200                          INC     UNITN                 ;...AND SET NEXT UNIT NUMBER.
     93 022046  023737  002200  002012                  CMP     UNITN,L$UNIT
     94 022054  103423                                  BLO     SETU
     95 022056  012737  177777  003106                  MOV     # 1,DUFLG
     96 022064  000401                                  BR      11$
     97 022066                                          DOCLN                 ;ABORT, NO MORE UNITS.
                022066  104444                          TRAP    C$DCLN
     98 022070  000240                          11$:    NOP
     99 022072                          PASRPT:
    100 022072  023727  002012  000001                  CMP     L$UNIT,#1             ;HOW MANY UNITS SELECTED?
    101 022100  101752                                  BLOS    NEWPAS                ;BR IF ONLY 1
    102 022102  005737  002216                          TST     DEVCNT                ;ARE ANY STILL RUNNING?
    103 022106  001747                                  BEQ     NEWPAS                ;BR IF NO
    104 022110                                          RFLAGS  R0
                022110  104421                          TRAP    C$RFLA
    105 022112  032700  000100                          BIT     #ISR,R0               ;SHOULD WE PRINT STATISTICS
    106 022116  001343                                  BNE     NEWPAS                ;BR IF NO
    107
    108 022120                                          DORPT
                022120  104424                          TRAP    C$DRPT
    109 022122  000741                                  BR      NEWPAS
    110 022124                          10$:
    111
```

INITIALIZE SECTION

```
    112 022124                          SETU:   GPHARD  UNITN,R0               ;GET UNIT N P-TABLE POINTER.
        022124  013700  002200                  MOV     UNITN,R0
        022130  104442                          TRAP    C$GPHRD
    113 022132                                  BNCOMPLETE NXTU                 ;BR IF UNIT NOT AVAILABLE.
        022132  103342                          BCC     NXTU
    114 022134  005037  003106                  CLR     DUFLG                   ;CLEAR "DROPPED" FLAG.
    115 022140  005237  002216                  INC     DEVCNT
    116 022144  012001                          MOV     (R0)+,R1                ;GET 1ST REGISTER ADDRESS.
    117 022146  010137  002204                  MOV     R1,CSRADDR              ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
    118
    119 022152  012001                          MOV     (R0)+,R1                ;GET VECTOR ADDRESS.
    120                                         ;MOV    (R0),R2                 ;GET INTERRUPT PRIORITY
    121                                         ;MOV    R2,IPRI                 ;SET INTERRUPT PRIORITY.
    122 022154  010137  002206                  MOV     R1,IVEC                 ;SET INTERRUPT VECTOR POINTER...
    123 022160  012721  016206                  MOV     #INTR,(R1)+             ;...VECTOR...
    124 022164  013721  002210                  MOV     IPRI,(R1)+              ;...AND PRIORITY.
    125
    126 022170                          1$:
    127                                 ;       TST     QVP             ;1ST PASS ??
    128                                 ;       BEQ     5$              ;NO, SKIP THE PASS 1 STUFF.
    129
    130                                 ;
    131                                 ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
    132                                 ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
    133                                 ;
    134 022170  013701  002200                  MOV     UNITN,R1
    135 022174  006301                          ASL     R1
    136 022176  052761  100000  003172          BIS     #BIT15,ERTABL(R1)       ;SAY DEVICE RUNNING
    137 022204  005037  005766                  CLR     EXTA                    ;CLEAR ERROR EXTENSION FLAG.
    138 022210  023727  002012  000001          CMP     L$UNIT,#1               ;ARE WE TESTING MULTIPLE UNITS?
    139 022216  101416                          BLOS    10$                     ;BR IF NO.
    140 022220                                  RFLAGS  R0                      ;YES -- GET OPERATOR FLAGS.
        022220  104421                          TRAP    C$RFLA
    141 022222  032700  001000                  BIT     #PNT,R0                 ;SHOULD WE PRINT UNIT #?
    142 022226  001412                          BEQ     10$                     ;BR IF NOT.
    143 022230                                  PRINTF  #PUNIT,UNITN            ;PRINT THE UNIT #
        022230  013746  002200                  MOV     UNITN,-(SP)
        022234  012746  022322                  MOV     #PUNIT,-(SP)
        022240  012746  000002                  MOV     #2,-(SP)
        022244  010600                          MOV     SP,R0
        022246  104417                          TRAP    C$PNTF
        022250  062706  000006                  ADD     #6,SP
    144 022254                          10$:
    145 022254  005037  003110                  CLR     NODEV
    146 022260  013701  002204                  MOV     CSRADDR,R1              ;ADDRESS OF FIRST REGISTER
    147 022264  010102                          MOV     R1,R2                   ;START OF REGISTERS
    148 022266  062702  000002                  ADD     #TSSR,R2                ;ADDRESS OF TSSR REGISTER
    149 022272  004737  016366                  JSR     PC,XNXM                 ;TEST BOTH CONTROLLER REGISTERS...
    150 022276  103005                          BCC     2$                      ;...AND BR IF ALL OK.
    151 022300  010137  003110                  MOV     R1,NODEV                ;FLAG DEVICE AS NON-EXISTENT
    152 022304  012737  177777  003106          MOV     #-1,DUFLG               ;DROP THIS UNIT.
    153 022312                          2$:
    154                                 ;
    155                                 ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
    156                                 ;
    157 022312                          5$:     SETPRI  #PRI00          ;ENABLE INTERRUPTS.
        022312  012700  000000                  MOV     #PRI00,R0
```

INITIALIZE SECTION

```
        022316  104441                          TRAP    C$SPRI
    158 022320                                  ENDINIT
        022320                          L10030:
        022320  104411                          TRAP    C$INIT
    159
    160 022322     045     116     045  PUNIT:  .ASCIZ  /%N%N%A***** TESTING UNIT %D2%A *****/
    161                                         .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.
                                ;--
022370                                  BGNAU
022370                          L$AU::
022370  010001                          MOV     R0,R1                   ; GET UNIT TO BE ADDED (R0)
022372  006301                          ASL     R1                      ; MAKE IT A WORD INDEX
022374  052761  100000  003172          BIS     #100000,ERTABL(R1)      ; SET THE "ACTIVE" BIT
022402  042761  040000  003172          BIC     #40000,ERTABL(R1)       ; CLEAR THE "DROPPED" BIT
022410                                  PRINTF  #1$,R0
022410  010046                          MOV     R0,-(SP)
022412  012746  022436                  MOV     #1$,-(SP)
022416  012746  000002                  MOV     #2,-(SP)
022422  010600                          MOV     SP,R0
022424  104417                          TRAP    C$PNTF
022426  062706  000006                  ADD     #6,SP
022432                                  EXIT    AU
022432  000167                          .WORD   J$JMP
022434  000026                          .WORD   L10031-2-.
022436     045     116     045  1$:     .ASCIZ  /%N%A  UNIT %D%A ADDED/
                                        .EVEN


022464                                  ENDAU           ; UNUSED.
022464                          L10031:
022464  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.


022466                                  BGNDU
022466                          L$DU::
022466  012737  177777  003106          MOV     #-1,DUFLG
022474  010001                          MOV     R0,R1
022476  006301                          ASL     R1
022500  052761  140000  003172          BIS     #140000,ERTABL(R1)      ; SAY DROPPED
022506  000240  000240  000240          240,240,240                     ; ?????????
022514                                  PRINTF  #1$,R0
022514  010046                          MOV     R0,-(SP)
022516  012746  022542                  MOV     #1$, (SP)
022522  012746  000002                  MOV     #2,-(SP)
022526  010600                          MOV     SP,R0
022530  104417                          TRAP    C$PNTF
022532  062706  000006                  ADD     #6,SP
022536                                  EXIT    DU
022536  000167                          .WORD   J$JMP
022540  000030                          .WORD   L10032 2-.
```

ADD AND DROP UNITS SECTIONS

```
    200 022542     045     116     045  1$:     .ASCIZ  /%N%A  UNIT %D%A DROPPED/
    201                                         .EVEN
    202 022572                                  ENDDU
        022572                          L10032:
        022572  104453                         TRAP    C$DU
    203                                 ;++
    204                                 ; AUTO-DROP CODE SECTION.
    205                                 ;--
    206 022574                                  BGNAUTO
        022574                          L$AUTO::
    207 022574  013705  002204                  MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
    208 022600  012703  000550                  MOV     #360.,R3                ;ENOUGH TIME FOR 2400' REEL TO REWIND
    209 022604  004737  016240          10$:    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
    210 022610  103420                          BCS     20$                     ;LEAVE WHEN SSR IS SET
    211 022612                                  DELAY   250.                    ;WAIT FOR .25 SECONDS
        022612  012727  000372                  MOV     #250.,(PC)+
        022616  000000                          .WORD   0
        022620  013727  002116                  MOV     L$DLY,(PC)+
        022624  000000                          .WORD   0
        022626  005367  177772                  DEC     -6(PC)
        022632  001375                          BNE     .-4
        022634  005367  177756                  DEC     -22(PC)
        022640  001367                          BNE     .-20
    212 022642  005303                          DEC     R3                      ;BUMP COUNTER DOWN
    213 022644  001357                          BNE     10$                     ;KEEP GOING
    214 022646  004737  017172                  JSR     PC,CKDROP               ;TRY AND DROP UNIT
    215 022652                          20$:
    216 022652                                  ENDAUTO                 ; UNUSED.
        022652                          L10033:
        022652  104461                         TRAP    C$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.
                                ;--
022654                                  BGNCLN
022654                          L$CLEAN::
022654  013705  002204                  MOV     CSRADDR,R5              ;POINT TO DEVICE REGISTER
022660  005737  003106                  TST     DUFLG           ;"DROPPED" FLAG IS SET ON...
022664  100405                          BMI     1$              ;...AND GROSS CONTROLLER FAULT...
                                                                ;...DON'T TRY TO XCT CLEANUP CODE.
022666  012765  000000  000002          MOV     #0,TSSR(R5)     ;DO SOFT INIT
022674  004737  016240                  JSR     PC,WAITF
022700                          1$:
022700                          2$:     ENDCLN
022700                          L10034:
022700  104412                          TRAP    C$CLEAN
                                ;**
                                ; THE REPORT CODING SECTION CONTAINS THE
                                ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;--
022702                                  BGNRPT
022702                          L$RPT::
022702                                  PRINTS  #DEVSUM
022702  012746  023144                  MOV     #DEVSUM, (SP)
022706  012746  000001                  MOV     #1,-(SP)
022712  010600                          MOV     SP,R0
022714  104416                          TRAP    C$PNTS
022716  062706  000004                  ADD     #4,SP
022722  010246                          MOV     R2,-(SP)
022724  010346                          MOV     R3,-(SP)
022726  010446                          MOV     R4,-(SP)
022730  012704  003172                  MOV     #ERTABL,R4      ; GET START OF ERROR TABLE.
022734  005003                          CLR     R3              ; CLEAR UNIT NUMBER
022736  011402                  1$:     MOV     (R4),R2         ; GET ERROR TABLE ENTRY & TEST IT.
022740  001467                          BEQ     4$              ; ZERO IF UNIT NOT RUN
022742  100066                          BPL     4$
022744  032702  040000                  BIT     #BIT14,R2       ; WAS UNIT DROPPED?
022750  001015                          BNE     2$              ; BR IF YES
022752  042702  170000                  BIC     #^C7777,R2      ; GET ERROR COUNT FIELD
022756                                  PRINTS  #DEVONL,R3,R2   ; PRINT
022756  010246                          MOV     R2,-(SP)
022760  010346                          MOV     R3,-(SP)
022762  012746  023201                  MOV     #DEVONL,-(SP)
022766  012746  000003                  MOV     #3,-(SP)
022772  010600                          MOV     SP,R0
022774  104416                          TRAP    C$PNTS
022776  062706  000010                  ADD     #10,SP
023002  000446                          BR      4$
023004  020227  160000          2$:     CMP     R2,#160000      ; WAS UNIT NON-EXISTENT?
023010  001012                          BNE     3$              ; BR IF NO
023012                                  PRINTS  #DEVNXR,R3
023012  010346                          MOV     R3,-(SP)
023014  012746  023251                  MOV     #DEVNXR,-(SP)
```

CLEAN UP AND REPORT CODING SECTIONS

```
        023020  012746  000002                  MOV     #2,-(SP)
        023024  010600                          MOV     SP,R0
        023026  104416                          TRAP    C$PNTS
        023030  062706  000006                  ADD     #6,SP
    257 023034  000431                          BR      4$
    258 023036  020227  160001          3$:     CMP     R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
    259 023042  001012                          BNE     30$             ; BR IF NO.
    260 023044                                  PRINTS  #DEVNRD,R3
        023044  010346                          MOV     R3,-(SP)
        023046  012746  023333                  MOV     #DEVNRD,-(SP)
        023052  012746  000002                  MOV     #2,-(SP)
        023056  010600                          MOV     SP,R0
        023060  104416                          TRAP    C$PNTS
        023062  062706  000006                  ADD     #6,SP
    261 023066  000414                          BR      4$
    262 023070  042702  170000          30$:    BIC     #^C7777,R2
    263 023074                                  PRINTS  #DEVDRO,R3,R2
        023074  010246                          MOV     R2,-(SP)
        023076  010346                          MOV     R3,-(SP)
        023100  012746  023414                  MOV     #DEVDRO,-(SP)
        023104  012746  000003                  MOV     #3,-(SP)
        023110  010600                          MOV     SP,R0
        023112  104416                          TRAP    C$PNTS
        023114  062706  000010                  ADD     #10,SP
    264 023120  062704  000002          4$:     ADD     #2,R4
    265 023124  005203                          INC     R3
    266 023126  020427  003372                  CMP     R4,#ERTABE
    267 023132  103701                          BLO     1$
    268 023134  012604                          MOV     (SP)+,R4
    269 023136  012603                          MOV     (SP)+,R3
    270 023140  012602                          MOV     (SP)+,R2
    271 023142                                  ENDRPT          ; UNUSED.
        023142                          L10035:
        023142  104425                          TRAP    C$RPT
    272
    273
    274 023144      045     116     045  DEVSUM: .ASCIZ  /%N%ADEVICE STATUS SUMMARY:%N/
    275 023201      045     101     040  DEVONL: .ASCIZ  /%A  UNIT %D3%A    ONLINE,  ERRORS = %D%N/
    276 023251      045     101     040  DEVNXR: .ASCIZ  /%A  UNIT %D3%A    DROPPED, NON-EXISTENT REGISTER%N/
    277 023333      045     101     040  DEVNRD: .ASCIZ  /%A  UNIT %D3%A    DROPPED, NOT READY AT STARTUP%N/
    278 023414      045     101     040  DEVDRO: .ASCIZ  /%A  UNIT %D3%A    DROPPED, ERRORS = %D%N/
    279                                          .EVEN
    280
    281 023464                                  ENDMOD
    282
    283
```

CLEAN UP AND REPORT CODING SECTIONS

```
                                        .TITLE  TSV5A - HARDWARE TESTS
                                        BGNMOD  TSV5
023464                          TSV5::
023464
```

```
                                .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.
                        ;
                        ;

023464                          BGNTST
023464                                                                  T1::
023464  012700  023662          MOV     #TST1ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
023470  004737  016500          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
023474  012737  000024  002214  MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
023502                  T1LOOP:
023502  005003                  CLR     R3                      ;USE R3 AS FATAL ERROR FLAG

023504                          BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
023504                                                                  T1.1:
023504  104402                                                          TRAP    C$BSUB

023506                          BRESET                          ;ISSUE A BUS RESET
023506  104433                                                          TRAP    C$RESET
023510  004737  016240          JSR     PC,WAITF                ;WAIT FOR READY
023514  016501  000002          MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
023520  010102                  MOV     R1,R2                   ;CONTENTS OF TSSR
023522  042702  176277          BIC     #^C<HIADDR!OFL>,R2      ;THESE BITS MAY BE SET
```

SEQ 0123

TEST 1: BUS RESET TEST

```
     83 023526  052702  002200                BIS     @SSR!NBA,R2             ;READY AND NEW DATA SHOULD BE SET
     84 023532  020102                        CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
     85 023534  001405                        BEQ     10$                     ;BRANCH IF COMPARE
     89 023536                                ERRDF   ERRNO,SFHERR,SFFMSG     ;REPORT A FATAL ERROR
        023536  104455                                                                  TRAP    C$ERDF
        023540  000145                                                                  .WORD   101
        023542  003701                                                                  .WORD   SFHERR
        023544  012072                                                                  .WORD   SFFMSG
     90 023546  005203                        INC     R3                      ;SET THE FATAL ERROR FLAG
     91 023550                          10$:
     92 023550                                ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
                                                                                      L10037:
        023550  104403                                                                  TRAP    C$ESUB
     93
```

TEST 1: BUS RESET TEST

```
     95 023552  005703                        TST     R3                      ;DID WE HAVE FATAL ERROR ?
     96 023554  001402                        BEQ     20$                     ;BRANCH IF NOT
     97 023556  004737  017172                JSR     PC,CKDROP               ;GO DROP THIS UNIT, IF ALLOWED
     98 023562  005003                  20$:  CLR     R3                      ;RESET FATAL ERROR FLAG
     99
    100
    101 023564                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
        023564                                                                          T1.2:
        023564  104402                                                                 TRAP    C$BSUB
    102
    103 023566  005065  000002                CLR     TSSR(R5)                ;WRITE TO ISSUE A SOFT RESET
    104 023572  004737  016240                JSR     PC,WAITF                ;WAIT FOR READY TO SET
    105 023576  016501  000002                MOV     TSSR(R5),R1             ;GET REGISTER TSSR DATA
    106 023602  010102                        MOV     R1,R2                   ;CONTENTS OF TSSR
    107 023604  042702  176277                BIC     #^C<HIADDR!OFL>,R2      ;THESE BITS MAY BE SET
    108 023610  052702  002200                BIS     #SSR!NBA,R2             ;READY AND NEW DATA SHOULD BE SET
    109 023614  020102                        CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
    110 023616  001405                        BEQ     10$                     ;BRANCH IF COMPARE
    114 023620                                ERRDF   ERRNO,SFIERR,SFFMSG     ;REPORT A FATAL ERROR
        023620  104455                                                                  TRAP    C$ERDF
        023622  000146                                                                 .WORD   102
        023624  003646                                                                 .WORD   SFIERR
        023626  012072                                                                 .WORD   SFFMSG
    115 023630  005203                        INC     R3                      ;SET THE ERROR FLAG
    116 023632                          10$:
    117 023632                                ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
        023632                                                                          L10040:
        023632  104403                                                                 TRAP    C$ESUB
    118
    119
    120 023634  005703                        TST     R3                      ;FATAL ERROR DETECTED ?
    121 023636  001402                        BEQ     20$                     ;BRANCH IF NOT
    122 023640  004737  017172                JSR     PC,CKDROP               ;SEE IF TIME TO DROP UNIT
    123 023644  004737  016446          20$:  JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
    124 023650  103002                        BCC     40$                     ;BRANCH IF NOT
    125 023652  000137  023502                JMP     T1LOOP                  ;LOOP UNTIL COUNT EXPIRED
    126 023656                          40$:  EXIT    TST                     ;ALL DONE THIS TEST
        023656  104432                                                                  TRAP    C$EXIT
        023660  000022                                                                 .WORD   L10036-.
    127
    128                                 ;+
    129                                 ;LOCAL TEXT MESSAGES FOR TEST
    130                                 ;-
    131
    132 023662     111     156     151  TST1ID: .ASCIZ  'Initialization'
    133                                         .EVEN
    134 023702                                  ENDTST
        023702                                                                  L10036:
        023702  104401                                                                 TRAP    C$ETST
    135
    136
```

TEST 2: WRAP DATA - HIGH BYTE

```
                                        .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).
                                ;
173 023704                      ;       BGNTST
    023704                                                                              T2::
178 023704  012700  024352          MOV     #TST2ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
179 023710  004737  016500          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
180 023714  012737  000024  002214  MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
181 023722  005004              T2LOOP: CLR     R4                  ;STARTING DATA PATTERN
182 023724  012703  177777          MOV     #-1,R3                  ;DO INIT ON FIRST TIME THROUGH
183 023730  005703              5$:     TST     R3                  ;DO WE NEED SOFT INIT
184 023732  001412                      BEQ     10$                 ;BRANCH IF NOT
185 023734  005003                      CLR     R3                  ;DON'T NEED INIT NEXT TIME THRU
186 023736  004737  015764              JSR     PC,SOFINIT          ;DO SOFT INIT OF CONTROLLER
187 023742  103406                      BCS     10$                 ;BR IF SOFT INIT = OK
191 023744  010001                      MOV     R0,R1               ;SAVE CONTENTS OF TSSR
192 023746                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
    023746  104455                                                                      TRAP    C$ERDF
    023750  000311                                                                     .WORD   201
    023752  003646                                                                     .WORD   SFIERR
    023754  012024                                                                     .WORD   SFIMSG
193 023756  005203                      INC     R3                  ;FORCE SOFT INIT ON NEXT PASS
194 023760  005037  002220          10$:    CLR     FATFLG              ;CLEAR FATAL ERROR FLAG
195
196 023764                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
```

TEST 2: WRAP DATA - HIGH BYTE

```
        023764  104404                                                          TRAP    C$BSEG
    197
    198 023766  110465  000001          MOVB    R4,TSDBH(R5)        ;SET MAINT MODE + WRITE DATA
    199 023772  004737  016240          JSR     PC,WAITF            ;WAIT FOR SSR TO SET
    200 023776  103411                  BCS     15$                 ;BR IF CARRY SET (GOOD RETURN)
    201 024000  010001                  MOV     R0,R1               ;SAVE CONTENTS OF TSSR
    202 024002  010402                  MOV     R4,R2               ;DATA THAT WAS WRITTEN
    206 024004                          ERRDF   ERRNO,T2SSR,EXPREC  ;DEVICE FATAL SSR FAILED TO SET
        024004  104455                                                          TRAP    C$ERDF
        024006  000312                                                          .WORD   202
        024010  024300                                                          .WORD   T2SSR
        024012  015464                                                          .WORD   EXPREC
    207 024014  005203                  INC     R3                  ;FORCE SOFT INIT ON NEXT PASS
    208 024016  005237  002220          INC     FATFLG              ;SET FATAL ERROR FLAG
    209 024022                  15$:    CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
        024022  104406                                                          TRAP    C$CLP1
    210 024024  005737  002220          TST     FATFLG              ;WAS FATAL ERROR RECEIVED ?
    211 024030  001402                  BEQ     20$                 ;BRANCH IF NOT
    212 024032  004737  017172          JSR     PC,CKDROP           ;SEE IF TIME TO DROP UNIT
    213 024036  010402          20$:    MOV     R4,R2               ;DATA PATTERN WRITTEN
    214 024040  042702  177774          BIC     #^C<BIT0!BIT1>,R2   ;CLEAR ALL BUT LOW 2 BITS
    215 024044  000302                  SWAB    R2                  ;BITS 8 AND 9 HAVE LOW DATA BITS
    216 024046  052702  002200          BIS     #SSR!NBA,R2         ;THESE BITS MUST BE SET ALSO
    217 024052  016501  000002          MOV     TSSR(R5),R1         ;GET THE CONTENTS OF TSSR
    218 024056  032701  000100          BIT     #OFL,R1             ;IS OFF-LINE BIT SET ?
    219 024062  001402                  BEQ     25$                 ;BRANCH IF NOT OFF-LINE
    220 024064  052702  000100          BIS     #OFL,R2             ;SET OFF-LINE IN EXPECTED DATA
    221 024070  020201          25$:    CMP     R2,R1               ;DOES EXPECTED MATCH RECEIVED ?
    222 024072  001405                  BEQ     30$                 ;OKAY IF MATCH
    226 024074                          ERRHRD  ERRNO,T2TSSR,EXPREC ;TSSR WASN'T CORRECT
        024074  104456                                                          TRAP    C$ERHRD
        024076  000313                                                          .WORD   203
        024100  024233                                                          .WORD   T2TSSR
        024102  015464                                                          .WORD   EXPREC
    227 024104  005203                  INC     R3                  ;FORCE SOFT INIT ON NEXT PASS
    228 024106                  30$:    CKLOOP                      ;LOOP ON ERROR ?
        024106  104406                                                          TRAP    C$CLP1
    229 024110  016501  000000          MOV     TSBA(R5),R1         ;GET TSBA REGISTER CONTENTS
    230 024114  005002                  CLR     R2                  ;
    231 024116  150402                  BISB    R4,R2               ;DATA PATTERN WRITTEN
    232 024120  000302                  SWAB    R2                  ;MOVE INTO TOP BYTE
    233 024122  150402                  BISB    R4,R2               ;BOTH HALVES SHOULD BE SAME
    234 024124  020102                  CMP     R1,R2               ;COMPARE EXPECTED TO RECEIVED
    235 024126  001405                  BEQ     35$                 ;ERROR IF NOT EQUAL
    239 024130                          ERRHRD  ERRNO,T2TSBA,EXPREC ;PRINT THE ERROR & EXPD/RECV
        024130  104456                                                          TRAP    C$ERHRD
        024132  000314                                                          .WORD   204
        024134  024166                                                          .WORD   T2TSBA
        024136  015464                                                          .WORD   EXPREC
    240 024140  005203                  INC     R3                  ;FORCE SOFT INIT ON NEXT PASS
    241
    242 024142                  35$:    ENDSEG                      ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        024142                                                                  10000$:
        024142  104405                                                          TRAP    C$ESEG
    243
    244 024144  105204                  INCB    R4                  ;NEXT DATA PATTERN TO TEST
    245 024146  001270                  BNE     5$                  ;BRANCH TILL BACK TO ZERO
```

TEST 2: WRAP DATA - HIGH BYTE

```
    246 024150  004737  016446                  JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
    247 024154  103002                          BCC     40$                     ;BRANCH IF NOT
    248 024156  000137  023722                  JMP     T2LOOP                  ;LOOP UNTIL COUNT EXPIRED
    249 024162                          40$:    EXIT    TST                     ;ALL DONE THIS TEST
        024162  104432                                                                      TRAP    C$EXIT
        024164  000214                                                                      .WORD   L10041-.
    250
    251                                 ;+
    252                                 ;LOCAL TEXT MESSAGES FOR TEST
    253                                 ;-
    254
    255 024166     124     123     102  T2TSBA: .ASCIZ  'TSBA Incorrect After TSDB High Write'
    256 024233     124     123     123  T2TSSR: .ASCIZ  'TSSR Incorrect After TSDB High Write'
    257 024300     116     157     040  T2SSR:  .ASCIZ  'No Sub-System Ready After TSDB High Write'
    258 024352     127     162     141  TST2ID: .ASCIZ  'Wrap Data - High Byte'
    259                                         .EVEN
    260 024400                                  ENDTST
        024400                                                                  L10041:
        024400  104401                                                                TRAP    C$ETST
    261
    262
```

TEST 3: WRAP DATA - LOW BYTE

```
                                        .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).
                                        ;
                                        BGNTST
024402                                                                  T3::
024402  012700  025045                    MOV     #TST3ID,R0          ;ASCII MESSAGE TO IDENTIFY TEST
024406  004737  016500                    JSR     PC,TSTSETUP         ;DO INITIAL TEST SETUP
024412  012737  000024  002214            MOV     #20.,LOOPCNT        ;PERFORM 20 ITERATIONS
024420  005004              T3LOOP:       CLR     R4                  ;STARTING DATA PATTERN
024422  012703  000001                    MOV     #1,R3               ;FORCE SOFT INIT ON FIRST PASS
024426  005703              5$:           TST     R3                  ;SET IF INIT IS REQUIRED
024430  001412                            BEQ     10$                 ;BRANCH IF NO INIT NEEDED
024432  005003                            CLR     R3                  ;ASSUME NO INIT NEEDED ON NEXT LOOP
024434  004737  015764                    JSR     PC,SOFINIT          ;DO SOFT INIT OF CONTROLLER
024440  103406                            BCS     10$                 ;BR IF SOFT INIT = OK
024442  010001                            MOV     R0,R1               ;SAVE CONTENTS OF TSSR
024444                                    ERRDF   ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
024444  104455                                                                  TRAP    C$ERDF
024446  000455                                                                  .WORD   301
024450  003646                                                                  .WORD   SFIERR
024452  012024                                                                  .WORD   SFIMSG
024454  005203                            INC     R3                  ;FORCE INIT ON NEXT PASS
024456  005037  002220          10$:      CLR     FATFLG              ;CLEAR FATAL ERROR FLAG

024462                                    BGNSEG                      ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
024462  104404                                                                  TRAP    C$BSEG

024464  110465  000000                    MOVB    R4,TSDB(R5)         ;SET MAINT MODE + WRITE DATA
024470  004737  016240                    JSR     PC,WAITF            ;WAIT FOR SSR TO SET
024474  103411                            BCS     15$                 ;BR IF CARRY SET (GOOD RETURN)
024476  010001                            MOV     R0,R1               ;SAVE CONTENTS OF TSSR
024500  010402                            MOV     R4,R2               ;DATA THAT WAS WRITTEN
024502                                    ERRDF   ERRNO,T3SSR,EXPREC  ;DEVICE FATAL SSR FAILED TO SET
```

TEST 3: WRAP DATA - LOW BYTE

```
                                                                                    TRAP    C$ERDF
                                                                                    .WORD   302
                                                                                    .WORD   T3SSR
                                                                                    .WORD   EXPREC
325 024512  005203                      INC     R3                  ;FORCE INIT ON NEXT PASS
326 024514  005237  002220              INC     FATFLG              ;SET THE FATAL ERROR FLAG
327 024520                      15$:    CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
    024520  104406                                                  TRAP    C$CLP1
328 024522  005737  002220              TST     FATFLG              ;DID WE GET FATAL ERROR ?
329 024526  001402                      BEQ     20$                 ;BRANCH IF NOT
330 024530  004737  017172              JSR     PC,CKDROP           ;DROP UNIT, IF DROP ALLOWED
331 024534  010402                  20$:MOV     R4,R2               ;DATA PATTERN WRITTEN
332 024536  042702  177774              BIC     #^C<BIT0!BIT1>,R2   ;CLEAR ALL BUT LOW 2 BITS
333 024542  000302                      SWAB    R2                  ;BITS 8 AND 9 HAVE LOW DATA BITS
334 024544  052702  002200              BIS     #SSR!NBA,R2         ;THESE BITS MUST BE SET ALSO
335 024550  016501  000002              MOV     TSSR(R5),R1         ;GET THE CONTENTS OF TSSR
336 024554  032701  000100              BIT     #OFL,R1             ;IS DRIVE OFF-LINE
337 024560  001402                      BEQ     25$                 ;BRANCH IF NOT OFFLINE
338 024562  052702  000100              BIS     #OFL,R2             ;SET OFF-LINE IN EXPECTED
339 024566  020201                  25$:CMP     R2,R1               ;DOES EXPECTED MATCH RECEIVED ?
340 024570  001405                      BEQ     30$                 ;OKAY IF MATCH
344 024572                              ERRHRD  ERRNO,T3TSSR,EXPREC ;TSSR WASN'T CORRECT
    024572  104456                                                  TRAP    C$ERHRD
    024574  000457                                                  .WORD   303
    024576  024730                                                  .WORD   T3TSSR
    024600  015464                                                  .WORD   EXPREC
345 024602  005203                      INC     R3                  ;FORCE INIT ON NEXT PASS
346 024604                      30$:    CKLOOP                      ;LOOP ON ERROR ?
    024604  104406                                                  TRAP    C$CLP1
347 024606  016501  000000              MOV     TSBA(R5),R1         ;GET TSBA REGISTER CONTENTS
348 024612  005002                      CLR     R2                  ;
349 024614  150402                      BISB    R4,R2               ;DATA PATTERN WRITTEN
350 024616  000302                      SWAB    R2                  ;MOVE INTO TOP BYTE
351 024620  150402                      BISB    R4,R2               ;BOTH HALVES SHOULD BE SAME
352 024622  020102                      CMP     R1,R2               ;COMPARE EXPECTED TO RECEIVED
353 024624  001405                      BEQ     35$                 ;BR IF ERROR NOT EQUAL
357 024626                              ERRHRD  ERRNO,T3TSBA,EXPREC ;PRINT THE ERROR & EXPD/RECV
    024626  104456                                                  TRAP    C$ERHRD
    024630  000460                                                  .WORD   304
    024632  024664                                                  .WORD   T3TSBA
    024634  015464                                                  .WORD   EXPREC
358 024636  005203                      INC     R3                  ;FORCE INIT ON NEXT PASS
359
360 024640                      35$:    ENDSEG                      ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
    024640                                                                  10000$:
    024640  104405                                                  TRAP    C$ESEG
361
362 024642  105204                      INCB    R4                  ;NEXT DATA PATTERN TO TEST
363 024644  001270                      BNE     5$                  ;ALL DONE WHEN BACK TO ZERO
364 024646  004737  016446              JSR     PC,TSTLOOP          ;SHOULD WE DO ITERATIONS ?
365 024652  103002                      BCC     40$                 ;BRANCH IF NOT
366 024654  000137  024420              JMP     T3LOOP              ;LOOP UNTIL COUNT EXPIRED
367 024660                      40$:    EXIT    TST                 ;ALL DONE THIS TEST
    024660  104432                                                  TRAP    C$EXIT
    024662  000210                                                  .WORD   L10042-.
368
369                             ;*
```

```
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;
024664      124     123     102  T3TSBA: .ASCIZ  'TSBA Incorrect After TSDB Low Write'
024730      124     123     123  T3TSSR: .ASCIZ  'TSSR Incorrect After TSDB Low Write'
024774      116     157     040  T3SSR:  .ASCIZ  'No Sub-System Ready After TSDB Low Write'
025045      127     162     141  TST3ID: .ASCIZ  'Wrap Data - Low Byte'
                                        .EVEN
025072                                  ENDTST
025072
025072                                                          L10042:
025072  104401                                                          TRAP    C$ETST
```

TEST 4: RAM TEST

```
                                .SBTTL  TEST  4: RAM TEST

                        ;       THIS TEST VERIFIES THAT ALL LOCATIONS OF THE RAM ON THE M7196
                        ;       CAN PROPERLY STORE AND READ BACK ALL DATA PATTERNS, AND THAT
                        ;       EACH RAM LOCATION IS UNIQUELY ADDRESSED (I.E., THAT ONE AND ONLY
                        ;       ONE LOCATION IS ACCESSED BY ANY PARTICULAR ADDRESS).  THESE
                        ;       TESTS ARE PERFORMED BY THREE SUBTESTS, DESCRIBED BELOW.  A
                        ;       BYPRODUCT OF THESE TESTS IS A VERIFICATION OF TWO REGISTERS IN
                        ;       THE 2901 AND THE CAPABLITY OF THE 2901 TO CORRECTLY PERFORM AN
                        ;       ADD.
                        ;
                        ;
                        ;       TEST 4 , SUBTEST 1: -
                        ;
                        ;       THIS SUBTEST VERIFIES EACH RAM LOCATION BY FIRST PLACING THE
                        ;       M7196 INTO MAINTENANCE MODE BY WRITING INTO THE LOW BYTE OF TSDB
                        ;       AND THEN PERFORMING THE FOLLOWING SEQUENCE FOR EACH ADDRESS
                        ;       0-7777 (OCTAL):
                        ;
                        ;
                        ;            1.  THE ADDRESS TO BE TESTED IS LOADED INTO THE TSDB (VIA A
                        ;                WORD WRITE).
                        ;
                        ;            2.  THE ADDRESSED RAM LOCATION IS WRITTEN, THEN READ INTO
                        ;                THE LOW BYTE OF TSBA, BY WRITING A DATA BYTE INTO THE
                        ;                LOW BYTE OF TSDB.
                        ;
                        ;            3.  THE LOW BYTE OF TSBA IS CHECKED TO SEE IF IT CONTAINS
                        ;                THE DATA PATTERN ORIGINALLY WRITTEN;  A DISCREPANCY IS
                        ;                REPORTED AS AN ERROR.
                        ;
                        ;            4.  THE ADDRESS OF THE LOCATION BEING TESTED IS AGAIN
                        ;                WRITTEN INTO TSDB (WORD WRITE), TO CAUSE THE LOCATION
                        ;                UNDER TEST TO AGAIN BE READ INTO THE LOW BYTE OF TSBA.
                        ;                THE LOW BYTE OF TSBA IS AGAIN CHECKED AND DISCREPANCIES
                        ;                REPORTED.
                        ;
                        ;            5.  THE HIGH BYTE OF TSBA IS CHECKED;  IT SHOULD CONTAIN
                        ;                THE SUM OF THE HIGH AND LOW BYTES LAST WRITTEN INTO
                        ;                TSDB AS A WORD.  A DISCREPANCY IS REPORTED AS A 2901
                        ;                PROBLEM.
                        ;
                        ;            6.  THE CONTENT OF TSSR IS CHECKED;  SETTING OF THE SC BIT
                        ;                IS IGNORED.  OTHER DISCREPANCIES IN TSSR ARE REPORTED.
                        ;
                        ;

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

                        MOV     #TST4ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
                        JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
                        MOV     #5,LOOPCNT              ;PERFORM 5 ITERATIONS
```

TEST 4: RAM TEST

```
                                T4LOOP:
                                        JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
                                        BCS     20$                     ;BR IF INIT WAS OK
                                        MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
                                        ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
                                                                                TRAP    C$ERDF
                                                                                .WORD   401
                                                                                .WORD   SFIERR
                                                                                .WORD   SFIMSG
                                20$:    CLR     R4                      ;SET RAM ADDRESS AT ZERO
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        CLRB    TSDB(R5)                ;SET INTO MAINTENANCE MODE
                                25$:
                                        BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
                                                                                TRAP    C$BSEG
                                        MOVB    R4,R2                   ;EXPECTED DATA FROM WRAP-AROUND
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        MOV     R4,TSDB(R5)             ;LOAD ADDRESS INTO TSDB
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        MOVB    R2,TSDB(R5)             ;LOADS DATA INTO RAM LOCATION
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        MOV     TSBA(R5),R1             ;READS WRAP DATA
                                        CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ
                                        BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
                                        ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
                                                                                TRAP    C$ERHRD
                                                                                .WORD   402
                                                                                .WORD   TSBAM2
                                                                                .WORD   EXPREC
                                30$:
                                        ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                                                                          10000$:
                                                                                TRAP    C$ESEG
                                        INC     R4                      ;NEXT ADDRESS
                                        CMP     R4,#10000               ;END OF RAM MEMORY CHECK
                                        BNE     25$                     ;LOOP TILL ALL RAM WRITTEN
                                        ESCAPE  SUB                     ;DON'T CONTINUE IF ERROR ON WRITE
                                                                                TRAP    C$ESCAPE
                                                                                .WORD   L10044 .
                                        CLR     R2                      ;CLEAR OUT R2 HIGH BITS
                                        DEC     R4                      ;SET BACK TO 7777
                                40$:    MOVB    R4,R2                   ;GET DATA PATTERN BACK IN SHAPE
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        MOV     R4,TSDB(R5)             ;LOAD UP RAM ADDRESS POINTER
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        MOV     TSBA(R5),R1             ;READ RAM CONTENTS BACK
                                        CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
                                        BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
                                        ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
                                                                                TRAP    C$ERHRD
                                                                                .WORD   403
                                                                                .WORD   TSBAM2
                                                                                .WORD   EXPREC
                                45$:    CKLOOP                          ;SCOPE LOOP
```

TEST 4: RAM TEST SEQ 0133

```
     025276  104406
489  025300  116501  000001          MOVB    TSBAH(R5),R1        ;HIGH BYTE READ OF TSBA                 TRAP    C$CLP1
490  025304  010402                  MOV     R4,R2               ;DATA PATTERN WRITTEN
491  025306  000302                  SWAB    R2                  ;HIGH TO LOW
492  025310  060402                  ADD     R4,R2               ;TOTAL OF BYTES IN LOW BYTE
493  025312  120102                  CMPB    R1,R2               ;SUM OF BYTES WRITTEN TO TSDB = TSBAH
494  025314  001404                  BEQ     50$                 ;BR IF OK, THEY SHOULD BE
498  025316                          ERRHRD  ERRNO,M2901,EXPREC  ;2901 PROBLEM ADDER
     025316  104456                                                          TRAP    C$ERHRD
     025320  000624                                                          .WORD   404
     025322  026152                                                          .WORD   M2901
     025324  015464                                                          .WORD   EXPREC
499  025326                  50$:    CKLOOP                      ;SCOPE LOOP
     025326  104406                                                          TRAP    C$CLP1
500  025330  005304                  DEC     R4                  ;DROP DATA COUNTER (PATTERN)
501  025332  002342                  BGE     40$                 ;NOT AT LOC. ZERO YET
502
503  025334                          ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     025334                                                  L10044:
     025334  104403                                                          TRAP    C$ESUB
504
```

```
    025336                                  BGNSUB                              ;///////////// BEGIN SUBTEST ////////////
    025336                                                                                     T4.2:
    025336  104402                                                                                             TRAP    C$BSUB
                                    ;       TEST 4, SUBTEST 2

                                    ;
                                    ;       THIS SUBTEST WRITES RAM WITH ALL ZEROS
                                    ;       THEN WALKS AN ALL ONES WORD DOWN THROUGH MEMORY
                                    ;
    025340  004737  015764                  JSR     PC,SOFINIT                  ;DO INITIALIZE ON CONTROLLER
    025344  103405                          BCS     20$                         ;BR IF INIT WAS OK
    025346  010001                          MOV     R0,R1                       ;CONTENTS OF TSSR REGISTER
    025350                                  ERRDF   ERRNO,SFIERR,SFIMSG         ;FATAL ERROR TSSR WAS NOT OK
    025350  104455                                                                                         TRAP    C$ERDF
    025352  000625                                                                                         .WORD   405
    025354  003646                                                                                         .WORD   SFIERR
    025356  012024                                                                                         .WORD   SFIMSG
    025360  005002                  20$:    CLR     R2                          ;TEST DATA = 0
    025362  005004                          CLR     R4                          ;STARTING RAM ADDRESS = 0
    025364  004737  016326                  JSR     PC,CHKTSSR                  ;WAIT FOR READY, NON AMBIGUOUS
    025370  105065  000000                  CLRB    TSDB(R5)                    ;SET INTO MAINTENANCE MODE
    025374                          25$:
    025374                                  BGNSEG                              ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
    025374  104404                                                                                         TRAP    C$BSEG

    025376  004737  016326                  JSR     PC,CHKTSSR                  ;WAIT FOR READY, NON-AMBIGUOUS
    025402  010465  000000                  MOV     R4,TSDB(R5)                 ;LOAD ADDRESS INTO TSDB
    025406  004737  016326                  JSR     PC,CHKTSSR                  ;WAIT FOR READY, NON-AMBIGUOUS
    025412  110265  000000                  MOVB    R2,TSDB(R5)                 ;LOADS DATA INTO RAM LOCATION
    025416  004737  016326                  JSR     PC,CHKTSSR                  ;WAIT FOR READY, NON-AMBIGUOUS
    025422  016501  000000                  MOV     TSBA(R5),R1                 ;READS WRAP DATA
    025426  120102                          CMPB    R1,R2                       ;DOES WRITTEN(WRAP) = READ ?
    025430  001404                          BEQ     30$                         ;BR IF OK, THEY ARE EQUAL
    025432                                  ERRHRD  ERRNO,TSBAM2,EXPREC         ;DATA NOT WRAPPED CORRECTLY
    025432  104456                                                                                         TRAP    C$ERHRD
    025434  000626                                                                                         .WORD   406
    025436  026242                                                                                         .WORD   TSBAM2
    025440  015464                                                                                         .WORD   EXPREC
    025442                          30$:
    025442                                  ENDSEG                              ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
    025442                                                                                     10000$:
    025442  104405                                                                                         TRAP    C$ESEG

    025444  005204                          INC     R4                          ;NEXT ADDRESS
    025446  020427  010000                  CMP     R4,#10000                   ;END OF RAM MEMORY CHECK
    025452  001350                          BNE     25$                         ;BR, MORE RAM TO GO

    025454  005304                  35$:    DEC     R4                          ;SET BACK TO 7777
    025456  005002                          CLR     R2                          ;SET TO ALL ZEROS
    025460  004737  016326          40$:    JSR     PC,CHKTSSR                  ;WAIT FOR READY (SSR) TO SET
    025464  010465  000000                  MOV     R4,TSDB(R5)                 ;LOAD UP THE ADDRESS FOR RAM
    025470  004737  016326                  JSR     PC,CHKTSSR                  ;WAIT FOR READY (SSR) TO SET
    025474  016501  000000                  MOV     TSBA(R5),R1                 ;READ THE RAM CONTENTS BACK
    025500  005002                          CLR     R2                          ;LOOKING FOR 000000 (EXPECTED)
    025502  120102                          CMPB    R1,R2                       ;BOTH SHOULD BE 00000000 BINARY
    025504  001404                          BEQ     43$                         ;BR, IF DATA IS GOOD
```

TEST 4: RAM TEST

```
559 025506                              ERRHRD  ERRNO,TSBAM3,EXPREC     ;CHARACTERISTICS DATA NOT CORRECT
    025506  104456                                                              TRAP    C$ERHRD
    025510  000627                                                              .WORD   407
    025512  026324                                                              .WORD   TSBAM3
    025514  015464                                                              .WORD   EXPREC
560 025516  012702  000377      43$:    MOV     #000377,R2              ;SET ALL ONES WORD
561 025522  010465  000000              MOV     R4,TSDB(R5)             ;LOAD UP RAM ADDRESS POINTER
562 025526  004737  016326              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
563 025532  110265  000000              MOVB    R2,TSDB(R5)             ;WRITE DATA INTO RAM
564 025536  004737  016326              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
565 025542  016501  000000              MOV     TSBA(R5),R1             ;READ RAM CONTENTS BACK
566 025546  120102                      CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
567 025550  001404                      BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
571 025552                              ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
    025552  104456                                                              TRAP    C$ERHRD
    025554  000630                                                              .WORD   408
    025556  026242                                                              .WORD   TSBAM2
    025560  015464                                                              .WORD   EXPREC
572 025562                      45$.    CKLOOP                          ;SCOPE LOOP
    025562  104406                                                              TRAP    C$CLP1
573 025564  004737  016326              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
574 025570  010465  000000              MOV     R4,TSDB(R5)             ;WORD WRITE TO SET UP ADDRESS
575 025574  004737  016326              JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
576 025600  116501  000001              MOVB    TSBAH(R5),R1            ;HIGH BYTE READ OF TSBA
577 025604  010403                      MOV     R4,R3                   ;DATA PATTERN WRITTEN
578 025606  000303                      SWAB    R3                      ;HIGH TO LOW
579 025610  060403                      ADD     R4,R3                   ;TOTAL OF BYTES IN LOW BYTE
580 025612  120103                      CMPB    R1,R3                   ;SUM OF BYTES WRITTEN TO TSDB = TSBAH
581 025614  001404                      BEQ     50$                     ;BR IF OK, THEY SHOULD BE
585 025616                              ERRHRD  ERRNO,M2901,EXPREC      ;2901 PROBLEM ADDER
    025616  104456                                                              TRAP    C$ERHRD
    025620  000631                                                              .WORD   409
    025622  026152                                                              .WORD   M2901
    025624  015464                                                              .WORD   EXPREC
586 025626                      50$:    CKLOOP                          ;SCOPE LOOP
    025626  104406                                                              TRAP    C$CLP1
587 025630  005304                      DEC     R4                      ;DROP RAM ADDRESS POINTER
588 025632  002312                      BGE     40$                     ;NOT AT LOC. ZERO YET
589
590 025634                              ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
    025634                                                                      L10045:
    025634  104403                                                              TRAP    C$ESUB
591
```

TEST 4: RAM TEST

```
                                        BGNSUB                          ;////,////////// BEGIN SUBTEST ////////////
                                                                                          T4.3:
                                                                                                  TRAP    C$BSUB
                                ;       TEST 4, SUBTEST 3
                                ;
                                ;       THIS SUBTEST WRITES RAM WITH ALL ONES
                                ;       THEN WALKS A ZERO WORD DOWN THROUGH MEMORY
                                ;
025640  004737  015764                  JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
025644  103405                          BCS     20$                     ;BR IF INIT WAS OK
025646  010001                          MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
025650                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
025650  104455                                                                  TRAP    C$ERDF
025652  000632                                                                  .WORD   410
025654  003646                                                                  .WORD   SFIERR
025656  012024                                                                  .WORD   SFIMSG
025660  012702  177777          20$:    MOV     #177777,R2              ;SET DATA AT ALL ONES
025664  005004                          CLR     R4                      ;SET RAM ADDRESS AT ZERO
025666  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025672  105065  000000                  CLRB    TSDB(R5)                ;SET INTO MAINTENANCE MODE
025676                          25$:
025676                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
025676  104404                                                                  TRAP    C$BSEG
025700  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025704  010465  000000                  MOV     R4,TSDB(R5)             ;LOAD ADDRESS INTO TSDB
025710  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025714  110265  000000                  MOVB    R2,TSDB(R5)             ;LOADS DATA INTO RAM LOCATION
025720  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
025724  016501  000000                  MOV     TSBA(R5),R1             ;READS WRAP DATA
025730  120102                          CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ ?
025732  001404                          BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
025734                                  ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
025734  104456                                                                  TRAP    C$ERHRD
025736  000633                                                                  .WORD   411
025740  026242                                                                  .WORD   TSBAM2
025742  015464                                                                  .WORD   EXPREC
025744                          30$:
025744                                  ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                                                                                        10000$:
025744  104405                                                                  TRAP    C$ESEG
025746  005204                          INC     R4                      ;NEXT ADDRESS
025750  020427  010000                  CMP     R4,#10000               ;END OF RAM MEMORY CHECK
025754  001350                          BNE     25$                     ;BR, MORE RAM TO GO
025756                                  ESCAPE  SUB                     ;NO CHECK IF WRITTEN INCORRECTLY
025756  104410                                                                  TRAP    C$ESCAPE
025760  000152                                                                   WORD   L10046 .
025762  005304          35$:    DEC     R4                              ;SET BACK TO 7777
025764  004737  016326  40$:    JSR     PC,CHKTSSR                      ;WAIT FOR READY, NON-AMBIGUOUS
025770  012702  000377          MOV     #000377,R2                      ;SET UP EXPECTED DATA REGISTER
025774  005001                  CLR     R1                              ;CLEAN OUT REGISTER
025776  010465  000000          MOV     R4,TSDB(R5)                     ;SELECT ADDRESS IN RAM
026002  004737  016326          JSR     PC,CHKTSSR                      ;WAIT FOR READY (SSR)
```

TEST 4: RAM TEST

```
641 026006  016501  000000                  MOV     TSBA(R5),R1             ;PICK UP RAM CONTENTS
642 026012  120102                          CMPB    R1,R2                   ;IS MEMORY STILL ALL ONES
643 026014  001404                          BEQ     43$                     ;BR, IF OK (ALL ONES)
647 026016                                  ERRHRD  ERRNO,TSBAM3,EXPREC     ;MEMORY CHANGED AFTER ALL ONES WRITE
    026016  104456                                                              TRAP    C$ERHRD
    026020  000634                                                              .WORD   412
    026022  026324                                                              .WORD   TSBAM3
    026024  015464                                                              .WORD   EXPREC
648 026026  005002                  43$:    CLR     R2                      ;SET UP NEW EXPECTED
649 026030  010465  000000                  MOV     R4,TSDB(R5)             ;LOAD UP RAM ADDRESS POINTER
650 026034  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
651 026040  110265  000000                  MOVB    R2,TSDB(R5)             ;WRITE DATA INTO RAM
652 026044  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
653 026050  016501  000000                  MOV     TSBA(R5),R1             ;READ RAM CONTENTS BACK
654 026054  120102                          CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
655 026056  001404                          BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
659 026060                                  ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
    026060  104456                                                              TRAP    C$ERHRD
    026062  000635                                                              .WORD   413
    026064  026242                                                              .WORD   TSBAM2
    026066  015464                                                              .WORD   EXPREC
660 026070                          45$:    CKLOOP                          ;SCOPE LOOP
    026070  104406                                                              TRAP    C$CLP1
661 026072  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
662 026076  116501  000001                  MOVB    TSBAH(R5),R1            ;HIGH BYTE READ OF TSBA
663 026102  010203                          MOV     R2,R3                   ;DATA PATTERN WRITTEN
664 026104  000303                          SWAB    R3                      ;HIGH TO LOW
665 026106  060203                          ADD     R2,R3                   ;TOTAL OF BYTES IN LOW BYTE
666 026110  120103                          CMPB    R1,R3                   ;SUM OF BYTES WRITTEN TO TSDB = TSBAH
667 026112  001404                          BEQ     50$                     ;BR IF OK, THEY SHOULD BE
671 026114                                  ERRHRD  ERRNO,M2901,EXPREC      ;2901 PROBLEM ADDER
    026114  104456                                                              TRAP    C$ERHRD
    026116  000636                                                              .WORD   414
    026120  026152                                                              .WORD   M2901
    026122  015464                                                              .WORD   EXPREC
672 026124                          50$:    CKLOOP                          ;SCOPE LOOP
    026124  104406                                                              TRAP    C$CLP1
673 026126  005304                          DEC     R4                      ;DROP RAM ADDRESS POINTER
674 026130  001315                          BNE     40$                     ;NOT AT LOC. ZERO YET
675
676 026132                                  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
    026132                                                                      L10046:
    026132  104403                                                              TRAP    C$ESUB
677
678 026134  004737  016446                  JSR     PC,TSTLOOP              ;DO WE NEED TO ITERATE TEST ?
679 026140  103002                          BCC     63$                     ;BRANCH IF NOT
680 026142  000137  025114                  JMP     T4LOOP                  ;EXECUTE AGAIN
681 026146                          63$:    EXIT    TST                     ;ALL DONE THIS TEST
    026146  104432                                                              TRAP    C$EXIT
    026150  000256                                                              .WORD   L10043 .
682                                 ;+
683                                 ;LOCAL TEXT MESSAGES FOR TEST
684                                 ;-
685
686 026152      040     124     123 M2901:  .ASCIZ  ' TSBA High Byte Not Sum of Last TSDB Write (2901 Error)'
687 026242      040     127     162 TSBAM2: .ASCIZ  ' Write to TSDB Not Equal to Read of TSBA Low Byte'
688 026324      127     162     151 TSBAM3: .ASCIZ  'Write To RAM Location Modified Another Location'
```

TEST 4: RAM TEST

```
    689 026404     122     101     115  TST4ID: .ASCIZ  'RAM Verification'
    690                                         .EVEN
    691 026426                                  ENDTST
        026426                                                  L10043:
        026426  104401                                                 TRAP    C$ETST
    692
```

```
                                .SBTTL  TEST  5: SECOND INITIALIZATION TEST

                        ;       THIS TEST VERIFIES THE SAME ELEMENTS AS DID INITIALIZATION TEST
                        ;       #1 AND ALSO CHECKS THAT CERTAIN PARTS OF RAM IS CLEARED TO ZERO
                        ;       AND THAT 2901 REGISTERS 10 AND 11 ARE ALSO CLEARED TO ZERO.
                        ;       THIS IS A CONFIDENCE CHECK OF A PART OF THE SELF-TEST SEQUENCE
                        ;       (I.E., THAT IT IS REALLY BEING EXECUTED).  FOR EACH OF TWO
                        ;       SUBTESTS (ONE FOR INITIALIZING VIA A BUS INIT, THE OTHER FOR
                        ;       INITIALIZING BY WRITING INTO THE TSSR), THE FOLLOWING SEQUENCE
                        ;       IS PERFORMED:
                        ;
                        ;
                        ;            1.  EACH RAM LOCATION AND 2901 REGISTERS 10 AND 11 ARE SET
                        ;                TO ALL 1'S BY USING WRITES INTO THE TSDB REGISTER (LOW
                        ;                BYTE AND MAINTENANCE MODE WORD WRITES).
                        ;
                        ;            2.  THE CONTROLLER IS INITIALIZED AND THE VARIOUS CHECKS ON
                        ;                THE TSSR DESCRIBED IN INITIALIZATION TEST #1 ARE
                        ;                PERFORMED.
                        ;
                        ;            3.  #1'S (377 OCTAL) ARE WRITTEN INTO THE LOW BYTE OF TSDB,
                        ;                WHICH SHOULD CAUSE RAM LOCATION 0 TO BE WRITTEN TO ALL
                        ;                1'S SINCE 2901 REGISTERS 10 AND 11, SPECIFYING THE RAM
                        ;                ADDRESS, SHOULD BE 0.  RAM LOCATION 0 IS VERIFIED BY
                        ;                WRITING A WORD OF ZEROS INTO THE TSDB.  THE RESULTING
                        ;                LOW BYTE OF TSBA SHOULD CONTAIN ALL 1'S.
                        ;
                        ;            4.  THE ENTIRE RAM IS SCANNED.  LOCATION 0 SHOULD CONTAIN
                        ;                ALL 1'S AND THE REMAINING LOCATIONS, EXCEPT FOR THE
                        ;                MESSAGE BUFFER IMAGE AREA, SHOULD CONTAIN 0.
                        ;                DISCREPANCIES ARE REPORTED.  AN ERROR AT THIS POINT IS
                        ;                MOST LIKELY DUE TO A ROM, PIPELINE OR SEQUENCER PROBLEM
                        ;                OR A TIMING PROBLEM.
                        ;
                        ;
                        ;
                        ;
                        ;
026430                          BGNTST
026430                                                                          T5::
026430  012700  027402          MOV     #TST5ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
026434  004737  016500          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
026440  012737  000024  002214  MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
026446                  T5LOOP:
026446  005037  002220          CLR     FATFLG                  ;CLEAR THE FATAL ERROR FLAG
                        ;
026452                          BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
026452                                                                          T5.1:
026452  104402                                                                  TRAP    C$BSUB
                        ;
026454  004737  015764          JSR     PC,SOFINIT              ;DO A SOFT TO START
026460  103404                  BCS     10$                     ;BRANCH IF O.K.
026462                          ERRDF   ERRNO,SFIERR,SFIMSG     ;REPORT ERROR AND DROP DRIVE
026462  104455                                                                  TRAP    C$ERDF
026464  000765                                                                  .WORD   501
026466  003646                                                                  .WORD   SFIERR
026470  012024                                                                  .WORD   SFIMSG
026472  012702  177777  10$:    MOV     #-1,R2                  ;ALL ONE DATA PATTERN
```

TEST 5: SECOND INITIALIZATION TEST

```
751 026476  005004                      CLR     R4                      ;STARTING RAM ADDRESS
752 026500  004737  016326               JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
753 026504  105065  000000       15$:    CLRB    TSDB(R5)                ;SET MAINTENANCE MODE
754 026510  004737  016326               JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
755 026514  010465  000000               MOV     R4,TSDB(R5)             ;SET THE NEXT RAM ADDRESS
756 026520  004737  016326               JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
757 026524  110265  000000               MOVB    R2,TSDB(R5)             ;LOAD TEST DATA
758 026530  005204                       INC     R4                      ;NEXT ADDRESS TO TEST
759 026532  020427  007777               CMP     R4,#7777                ;COMPARE TO LAST ADDRESS
760 026536  003762                       BLE     15$                     ;BRANCH TILL ALL DATA WRITTEN
761 026540                               BRESET                          ;ISSUE A BUS RESET
    026540  104433                                                               TRAP    C$RESET
762 026542  004737  016326               JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
763 026546  016501  000002               MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
764 026552  010102                       MOV     R1,R2                   ;CONTENTS OF TSSR
765 026554  042702  176277               BIC     #^C<HIADDR!OFL>,R2      ;THESE BITS MAY BE SET
766 026560  052702  002200               BIS     #SSR!NBA,R2             ;READY AND NEW DATA SHOULD BE SET
767 026564  020102                       CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
768 026566  001406                       BEQ     20$                     ;BRANCH IF COMPARE
772 026570                               ERRDF   ERRNO,SFHERR,SFFMSG     ;REPORT A FATAL ERROR
    026570  104455                                                               TRAP    C$ERDF
    026572  000766                                                               .WORD   502
    026574  003701                                                               .WORD   SFHERR
    026576  012072                                                               .WORD   SFFMSG
773 026600  005237  002220               INC     FATFLG                  ;SET FATAL ERROR FLAG
774 026604                       20$:    CKLOOP                          ;LOOP ON ERROR IF FLAG SET
    026604  104406                                                               TRAP    C$CLP1
775 026606                               ESCAPE  SUB                     ;EXIT IF FATAL ERROR DETECTED
    026606  104410                                                               TRAP    C$ESCAPE
    026610  000170                                                               .WORD   L10050-.
776 026612  004737  016326               JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
777 026616  105065  000000               CLRB    TSDB(R5)                ;PUT BACK INTO MAINTENANCE MODE
778 026622  004737  016326               JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
779 026626  005065  000000               CLR     TSDB(R5)                ;SET ADDRESS BACK TO 0000
780 026632  012702  000377               MOV     #377,R2                 ;
781 026636  004737  016326               JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
782 026642  110265  000000               MOVB    R2,TSDB(R5)             ;SHOULD POINT TO RAM 0
783 026646  004737  016326               JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
784 026652  005065  000000               CLR     TSDB(R5)                ;SELECT LOCATION 0
785 026656  004737  016326               JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
786 026662  116501  000000               MOVB    TSBA(R5),R1             ;READ RAM LOCATION SPECIFIED
787 026666  120102                       CMPB    R1,R2                   ;LOCATION SHOULD BE 377 OCTAL
788 026670  001406                       BEQ     25$                     ;BR IF OK
789 026672                               ERRDF   ERRNO,TSADDR,EXPREC     ;WASN'T POINTING TO CORRECT LOC.
    026672  104455                                                               TRAP    C$ERDF
    026674  000766                                                               .WORD   502
    026676  027470                                                               .WORD   TSADDR
    026700  015464                                                               .WORD   EXPREC
790 026702  005237  002220               INC     FATFLG                  ;SET THE FATAL ERROR FLAG
791 026706                       25$:    CKLOOP                          ;SCOPE LOOP
    026706  104406                                                               TRAP    C$CLP1
792 026710                               ESCAPE  SUB                     ;NO MORE CHECKS IF FATAL ERROR
    026710  104410                                                               TRAP    C$ESCAPE
    026712  000066                                                               .WORD   L10050-.
793 026714  012704  000310               MOV     #310,R4                 ;START WITH LOC 310
794 026720  005002                       CLR     R2                      ;MEMORY EXPECTED SHOULD BE 000000
795 026722  004737  016326               JSR     PC,CHKTSSR              ;WAIT FOR READY, NON AMBIGUOUS
```

TEST 5: SECOND INITIALIZATION TEST

```
    796 026726  010465  000000          30$:    MOV     R4,TSDB(R5)             ;SELECT LOCATION SPECIFIED
    797 026732  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
    798 026736  116501  000000                  MOVB    TSBA(R5),R1             ;READ LOC CONTENTS
    799 026742  120102                          CMPB    R1,R2                   ;CHECK MEMORY FOR 000000
    800 026744  001406                          BEQ     40$                     ;BRANCH IF DATA OKAY
    801 026746                                  ERRDF   ERRNO,T5MEM,SFFMSG      ;MEMORY NOT ZERO AFTER INIT.
        026746  104455                                                              TRAP    C$ERDF
        026750  000766                                                              .WORD   502
        026752  027432                                                              .WORD   T5MEM
        026754  012072                                                              .WORD   SFFMSG
    802 026756  005237  002220                  INC     FATFLG                  ;SET THE FATAL ERROR FLAG
    803 026762                          40$:    CKLOOP
        026762  104406                                                              TRAP    C$CLP1
    804 026764                                  ESCAPE  SUB                     ;EXIT ON FATAL ERROR
        026764  104410                                                              TRAP    C$ESCAPE
        026766  000012                                                              .WORD   L10050-.
    805 026770  005204                          INC     R4                      ;LOOK AT NEXT RAM LOC.
    806 026772  020427  000400                  CMP     R4,#400                 ;AT TOP OF RAM ADDRESS SPACE
    807 026776  001353                          BNE     30$                     ;BRANCH TILL ALL MEMORY TESTED
    808
    809 027000                                  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        027000                                                                      L10050:
        027000  104403                                                              TRAP    C$ESUB
    810
    811 027002  005737  002220                  TST     FATFLG                  ;IS FATAL ERROR FLAG SET ?
    812 027006  001404                          BEQ     50$                     ;BRANCH IF NOT
    813 027010  004737  017172                  JSR     PC,CKDROP               ;NO LOOP, TRY TO DROP DEVICE
    814 027014  005037  002220                  CLR     FATFLG                  ;CLEAR THE FATAL ERROR FLAG
    815 027020                          50$:
```

```
817 027020                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
    027020                                                                          T5.2:
    027020  104402                                                                      TRAP    C$BSUB
818
819 027022  004737  015764          JSR     PC,SOFINIT              ;DO A SOFT TO START
820 027026  103404                  BCS     10$                     ;BRANCH IF O.K.
824 027030                          ERRDF   ERRNO,SFIERR,SFIMSG     ;REPORT ERROR AND DROP DRIVE
    027030  104455                                                                      TRAP    C$ERDF
    027032  000767                                                                     .WORD   503
    027034  003646                                                                     .WORD   SFIERR
    027036  012024                                                                     .WORD   SFIMSG
825 027040  012702  177777      10$:    MOV     #-1,R2                  ;ALL ONE DATA PATTERN
826 027044  005004                  CLR     R4                      ;STARTING RAM ADDRESS
827 027046  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
828 027052  105065  000000      15$:    CLRB    TSDB(R5)                ;SET MAINTENANCE MODE
829 027056  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
830 027062  010465  000000          MOV     R4,TSDB(R5)             ;SET THE NEXT RAM ADDRESS
831 027066  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
832 027072  110265  000000          MOVB    R2,TSDB(R5)             ;LOAD TEST DATA
833 027076  005204                  INC     R4                      ;NEXT ADDRESS TO TEST
834 027100  020427  007777          CMP     R4,#7777                ;COMPARE TO LAST ADDRESS
835 027104  003762                  BLE     15$                     ;BRANCH TILL ALL DATA WRITTEN
836 027106  005065  000002          CLR     TSSR(R5)                ;ISSUE A SOFT RESET
837 027112  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
838 027116  016501  000002          MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
839 027122  010102                  MOV     R1,R2                   ;CONTENTS OF TSSR
840 027124  042702  176277          BIC     #^C<HIADDR!OFL>,R2      ;THESE BITS MAY BE SET
841 027130  052702  002200          BIS     #SSR!NBA,R2             ;READY AND NEW DATA SHOULD BE SET
842 027134  020102                  CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
843 027136  001406                  BEQ     20$                     ;BRANCH IF COMPARE
847 027140                          ERRDF   ERRNO,SFHERR,SFFMSG     ;REPORT A FATAL ERROR
    027140  104455                                                                      TRAP    C$ERDF
    027142  000770                                                                     .WORD   504
    027144  003701                                                                     .WORD   SFHERR
    027146  012072                                                                     .WORD   SFFMSG
848 027150  005237  002220          INC     FATFLG                  ;SET FATAL ERROR FLAG
849 027154                  20$:    CKLOOP                          ;LOOP ON ERROR IF FLAG SET
    027154  104406                                                                      TRAP    C$CLP1
850 027156                          ESCAPE  SUB                     ;EXIT IF FATAL ERROR DETECTED
    027156  104410                                                                      TRAP    C$ESCAPE
    027160  000170                                                                     .WORD   L10051 .
851 027162  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
852 027166  105065  000000          CLRB    TSDB(R5)                ;PUT BACK INTO MAINTENANCE MODE
853 027172  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
854 027176  005065  000000          CLR     TSDB(R5)                ;SET ADDRESS BACK TO 0000
855 027202  012702  000377          MOV     #377,R2                 ;
856 027206  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
857 027212  110265  000000          MOVB    R2,TSDB(R5)             ;SHOULD POINT TO RAM 0
858 027216  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
859 027222  005065  000000          CLR     TSDB(R5)                ;SELECT LOCATION 0
860 027226  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
861 027232  116501  000000          MOVB    TSBA(R5),R1             ;READ RAM LOCATION SPECIFIED
862 027236  120102                  CMPB    R1,R2                   ;LOCATION SHOULD BE 377 OCTAL
863 027240  001406                  BEQ     25$                     ;BR IF OK
864 027242                          ERRDF   ERRNO,T5ADDR,EXPREC     ;WASN'T POINTING TO CORRECT LOC.
    027242  104455                                                                      TRAP    C$ERDF
    027244  000770                                                                     .WORD   504
```

```
                                                                                .WORD   T5ADDR
                                                                                .WORD   EXPREC
865 027252  005237  002220          INC     FATFLG          ;SET THE FATAL ERROR FLAG
866 027256                  25$:    CKLOOP                  ;SCOPE LOOP
    027256  104406                                                              TRAP    C$CLP1
867 027260                          ESCAPE  SUB             ;NO MORE CHECKS IF FATAL ERROR
    027260  104410                                                              TRAP    C$ESCAPE
    027262  000066                                                              .WORD   L10051-.
868 027264  012704  000310          MOV     #310,R4         ;START WITH LOC 310
869 027270  005002                  CLR     R2              ;MEMORY EXPECTED SHOULD BE 000000
870 027272  004737  016326          JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
871 027276  010465  000000  30$:    MOV     R4,TSDB(R5)     ;SELECT LOCATION SPECIFIED
872 027302  004737  016326          JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
873 027306  116501  000000          MOVB    TSBA(R5),R1     ;READ LOC CONTENTS
874 027312  120102                  CMPB    R1,R2           ;CHECK MEMORY FOR 000000
875 027314  001406                  BEQ     40$             ;BRANCH IF DATA OKAY
876 027316                          ERRDF   ERRNO,T5MEM,SFFMSG      ;MEMORY NOT ZERO AFTER INIT.
    027316  104455                                                              TRAP    C$ERDF
    027320  000770                                                              .WORD   504
    027322  027432                                                              .WORD   T5MEM
    027324  012072                                                              .WORD   SFFMSG
877 027326  005237  002220          INC     FATFLG          ;SET THE FATAL ERROR FLAG
878 027332                  40$:    CKLOOP
    027332  104406                                                              TRAP    C$CLP1
879 027334                          ESCAPE  SUB             ;EXIT ON FATAL ERROR
    027334  104410                                                              TRAP    C$ESCAPE
    027336  000012                                                              .WORD   L10051-.
880 027340  005204                  INC     R4              ;LOOK AT NEXT RAM LOC.
881 027342  020427  000400          CMP     R4,#400         ;AT TOP OF RAM ADDRESS SPACE
882 027346  001353                  BNE     30$             ;BRANCH TILL ALL MEMORY TESTED
883
884 027350                          ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
    027350                                                              L10051:
    027350  104403                                                              TRAP    C$ESUB
885
886 027352  005737  002220          TST     FATFLG          ;IS FATAL ERROR FLAG SET ?
887 027356  001402                  BEQ     50$             ;BRANCH IF NOT
888 027360  004737  017172          JSR     PC,CKDROP       ;NO LOOP, TRY TO DROP DEVICE
889 027364  004737  016446  50$:    JSR     PC,TSTLOOP      ;SHOULD WE DO ITERATIONS ?
890 027370  103002                  BCC     60$             ;BRANCH IF NOT
891 027372  000137  026446          JMP     T5LOOP          ;LOOP UNTIL COUNT EXPIRED
892 027376                  60$:    EXIT    TST             ;ALL DONE THIS TEST
    027376  104432                                                              TRAP    C$EXIT
    027400  000132                                                              .WORD   L10047-.
893
894                                 ;+
895                                 ;LOCAL TEXT MESSAGES FOR TEST
896                                 ;-
897
898 027402     105     170     164  T5T5ID: .ASCIZ  'Extended Initialization'
899 027432     111     156     143  T5MEM:  .ASCIZ  'Incorrect RAM Data After Init'
900 027470     111     156     143  T5ADDR: .ASCIZ  'Incorrect RAM Address After Init'
901                                 .EVEN
902 027532                          ENDTST
    027532                                                              L10047:
    027532  104401                                                              TRAP    C$ETST
903
```

```
                        .SBTTL  TEST  6: COMMAND REJECT


                ;       THIS TEST VERIFIES THAT ALL COMMANDS OTHER THAN WRITE
                ;       CHARACTERISTICS ARE REJECTED DUE TO THE NEED BUFFER ADDRESS
                ;       (NBA) BIT BEING SET IN TSSR, AND THAT THE TSBA AND TSSR
                ;       REGISTERS ARE LEFT IN THE PROPER STATE AFTER EACH COMMAND IS
                ;       REJECTED.  THIS TEST CHECKS MICROPROCESSOR SEQUENCING, BASIC
                ;       COMMAND DECODING AND DATI DMA HANDLING.  THIS TEST CONTAINS TWO
                ;       SUBTESTS:  SUBTEST 1 SEQUENCES THROUGH ALL COMMAND WORDS (OTHER
                ;       THAN WRITE CHARACTERISTICS) WITH THE INTERRUPT ENABLE (IE) BIT
                ;       CLEAR AND VERIFIES THAT AN INTERRUPT IS NOT GENERATED BY THE
                ;       REJECTED COMMAND;  SUBTEST 2 PERFORMS SIMILARLY TO SUBTEST 1 BUT
                ;       SETS THE IE BIT IN EACH COMMAND WORD AND VERIFIES THAT AN
                ;       INTERRUPT IS GENERATED WHEN THE COMMAND IS REJECTED.  SUBTEST 1
                ;       SETS UP THE INTERRUPT SERVICE ROUTINE TO FLAG UNEXPECTED
                ;       INTERRUPTS.  THE COMMAND WORD IN THE COMMAND BUFFER IS
                ;       INITIALIZED TO 100000 (OCTAL) AND THE REMAINING THREE WORDS IN
                ;       THE COMMAND BUFFER ARE SET TO KNOWN UNIQUE PATTERNS.  THEN THE
                ;       FOLLOWING SEQUENCE IS PERFORMED:
                ;
                ;            1.  INITIALIZE THE CONTROLLER BY WRITING INTO THE TSSR;
                ;                PROPER INITIAL CONDITIONS ARE VERIFIED.
                ;
                ;            2.  TSDB IS WRITTEN WITH ADDRESS OF THE COMMAND BUFFER TO
                ;                START PROCESSING.
                ;
                ;            3.  THE PROGRAM WAITS FOR SSR TO SET;  IF SSR DOES NOT SET,
                ;                AN ERROR REPORT IS ISSUED AND THE TEST IS ABORTED.
                ;
                ;            4.  THE CONTENTS OF TSSR ARE CHECKED.  TSSR IS CORRECT IF
                ;                IT CONTAINS EITHER OCTAL 102206 OR 102306 (BIT 6
                ;                DEPENDS UPON THE STATE OF THE TAPE TRANSPORT).
                ;
                ;            5.  THE CONTENTS OF TSBA ARE CHECKED.  TSBA SHOULD CONTAIN
                ;                THE INITIAL COMMAND BUFFER ADDRESS (LOADED IN STEP 2)
                ;                PLUS 10 (OCTAL);  I.E., TSBA SHOULD POINT TO THE WORD
                ;                JUST AFTER THE COMMAND PACKET (NOTE THAT 4 COMMAND
                ;                PACKET WORDS ARE ALWAYS FETCHED).
                ;
                ;            6.  USING THE MAINTENANCE MODE WRAPAROUND FUNCTIONS, THE
                ;                COMMAND IMAGE BLOCK IN THE M7196'S RAM (LOCATIONS
                ;                201-210 (OCTAL)) ARE CHECKED;  THE IMAGE SHOULD CONTAIN
                ;                A COPY OF THE FOUR COMMAND PACKET WORDS AS SET UP IN
                ;                CPU MEMORY.
                ;
                ;            7.  THE COMMAND WORD IN THE COMMAND BUFFER IS INCREMENTED
                ;                TO THE NEXT PATTERN NOT CONTAINING WRITE
                ;                CHARACTERISTICS OR IE.  THE REMAINING THREE WORD OF THE
                ;                COMMAND BUFFER ARE SEQUENCED WITH PSEUDO-RANDOM DATA.
                ;                IF THE COMMAND WORD HAS NOT REACHED ITS MAXIMUM VALUE
                ;                (177777+1), THE TEST SEQUENCE IS REPEATED.
                ;
                ;
                ;       SUBTEST 2 IS IDENTICAL TO SUBTEST 1, EXCEPT THAT THE PROGRAM
```

TEST 6: COMMAND REJECT

```
                                        ;       CAUSES THE IE BIT TO BE SET IN EACH COMMAND WORD AND THEN
                                        ;       VERIFIES THAT AN INTERRUPT OCCURS.
                                        ;
                                        BGNTST
                                                                                        T6::
 027534  012700  030763                 MOV     #TST6ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
 027540  004737  016500                 JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
 027544  012737  000024  002214         MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
 027552                                 T6LOOP:
 027552                                 BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
                                                                                        T6.1:
 027552  104402                                                                          TRAP    C$BSUB
                                        SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
 027554  012700  000000                                                                 MOV     #PRI00,R0
 027560  104441                                                                        TRAP    C$SPRI
 027562  012704  030440                 MOV     #T6PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
 027566  012703  002750                 MOV     #TSTBLK,R3              ;BLOCK OF TEST DATA
 027572  012314                 5$:     MOV     (R3)+,(R4)              ;INSERT THE NEXT TEST DATA WORD
 027574                                 BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
 027574  104404                                                                          TRAP    C$BSEG
 027576  004737  015764                 JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 027602  103405                         BCS     10$                     ;BR IF SOFT INIT = OK
 027604  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 027606                                 ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
 027606  104455                                                                          TRAP    C$ERDF
 027610  001131                                                                        .WORD   601
 027612  003646                                                                        .WORD   SFIERR
 027614  012024                                                                        .WORD   SFIMSG
 027616  005037  002220         10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 027622  005037  002222                 CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 027626  004737  016326                 JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
 027632  042714  000200                 BIC     #BIT7,(R4)              ;DISABLE INTERRUPTS
 027636  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 027642  004737  016240                 JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 027646  103407                         BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 027650  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 027652                                 ERRDF   ERRNO,T6SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
 027652  104455                                                                          TRAP    C$ERDF
 027654  001132                                                                        .WORD   602
 027656  030475                                                                        .WORD   T6SSR
 027660  012036                                                                        .WORD   PKTSSR
 027662  005237  002220                 INC     FATFLG                  ;SET FATAL ERROR FLAG
 027666                         15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
 027666  104406                                                                        TRAP    C$CLP1
 027670                                 ESCAPE  SUB                     ;BY PASS SUBTEST IF FATAL ERROR
 027670  104410                                                                        TRAP    C$ESCAPE
 027672  000170                                                                        .WORD   L10053 .
 027674  005737  002222                 TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 027700  001404                         BEQ     22$                     ;BRANCH IF NOT
 027702                                 ERRHRD  ERRNO,T6INT,PKTSSR
 027702  104456                                                                        TRAP    C$ERHRD
 027704  001133                                                                        .WORD   603
 027706  030553                                                                        .WORD   T6INT
 027710  012036                                                                        .WORD   PKTSSR
 027712  012702  102206         22$:    MOV     #SC!NBA!SSR!TSREJ,R2    ;EXPECTED CONTENTS OF TSSR
 027716  004737  016326                 JSR     PC,CHKTSSR              ;WAIT FOR READY, NON AMBIGUOUS
```

TEST 6: COMMAND REJECT

```
1011 027722  016501  000002          MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1012 027726  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1013 027732  001402                  BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1014 027734  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1015 027740  020201            25$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1016 027742  001404                  BEQ     30$                     ;OKAY IF MATCH
1020 027744                          ERRHRD  ERRNO,T6NBA,PKTSSR      ;NBA NOT SET TO REJECT
     027744  104456                                                          TRAP    C$ERHRD
     027746  001134                                                          .WORD   604
     027750  030450                                                          .WORD   T6NBA
     027752  012036                                                          .WORD   PKTSSR
1021 027754                    30$:  CKLOOP                          ;LOOP ON ERROR ?
     027754  104406                                                          TRAP    C$CLP1
1022 027756  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1023 027762  016501  000000          MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
1024 027766  010402                  MOV     R4,R2                   ;START OF THE PACKET
1025 027770  062702  000010          ADD     #10,R2                  ;EXPECT TSDA TO PACKET + 10
1026 027774  020102                  CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1027 027776  001404                  BEQ     35$                     ;ERROR IF NOT EQUAL
1031 030000                          ERRHRD  ERRNO,T6TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     030000  104456                                                          TRAP    C$ERHRD
     030002  001135                                                          .WORD   605
     030004  030711                                                          .WORD   T6TSBA
     030006  015464                                                          .WORD   EXPREC
1032
1033
1034 030010  004737  011104    35$:  JSR     PC,CKRAM                ;SEE IF DATA IN RAM IS CORRECT
1035 030014  103404                  BCS     40$                     ;BRANCH IF PACKET IN RAM IS CORRECT
1039 030016                          ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
     030016  104456                                                          TRAP    C$ERHRD
     030020  001136                                                          .WORD   606
     030022  004741                                                          .WORD   PKTRAM
     030024  015500                                                          .WORD   RAMERR
1040 030026                    40$:  ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     030026                                                                 10000$:
     030026  104405                                                                 TRAP    C$ESEG
1041 030030  011300                  MOV     (R3),R0                 ;PACKET COMMAND WORD
1042 030032  042700  177740          BIC     #177740,R0              ;GET BITS 0-4
1043 030036  020027  000004          CMP     R0,#4                   ;DON'T TEST WRITE CHARACTERISTICS
1044 030042  001002                  BNE     45$                     ;BRANCH IF OK
1045 030044  062703  000002          ADD     #2,R3                   ;GET NEXT WORD FROM DATA TABLE
1046 030050  020327  003056    45$:  CMP     R3,#TBLEND              ;REACHED END OF TABLE ?
1047 030054  103002                  BHIS    50$                     ;BRANCH IF END OF TAB
1048 030056  000137  027572          JMP     5$                      ;CONTINUE TEST WITH NEW DATA
1049
1050 030062                    50$:  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     030062                                                                 L10053:
     030062  104403                                                                 TRAP    C$ESUB
1051
1052 030064  005737  002220          TST     FATFLG                  ;ANY FATAL ERRORS ?
1053 030070  001402                  BEQ     60$                     ;BRANCH IF NOT
1054 030072  004737  017172          JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
```

TEST 6: COMMAND REJECT

```
1056 030076                          60$:   BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     030076                                                                                    T6.2:
     030076  104402                                                                                  TRAP    C$BSUB
1057
1058 030100                                 SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     030100  012700  000000                                                                          MOV     #PRI00,R0
     030104  104441                                                                                  TRAP    C$SPRI
1059 030106  012704  030440                 MOV     #T6PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1060 030112  012703  002750                 MOV     #TSTBLK,R3              ;START OF TEST DATA
1061 030116  012314                  5$:    MOV     (R3)+,(R4)              ;PLACE NEXT DATA WORD IN PACKET
1062 030120                                 BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     030120  104404                                                                                  TRAP    C$BSEG
1063 030122  004737  015764                 JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1064 030126  103405                         BCS     10$                     ;BR IF SOFT INIT = OK
1068 030130  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1069 030132                                 ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     030132  104455                                                                                  TRAP    C$ERDF
     030134  001137                                                                                  .WORD   607
     030136  003646                                                                                  .WORD   SFIERR
     030140  012024                                                                                  .WORD   SFIMSG
1070 030142  005037  002220          10$:   CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1071 030146  005037  002222                 CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1072 030152  004737  016326                 JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1073 030156  052714  000200                 BIS     #BIT7,(R4)              ;ENABLE INTERRUPTS
1074 030162  010465  000000                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
1075 030166  004737  016240                 JSR     PC,WAITF                ;WAIT FOR SSR TO SET
1076 030172  103407                         BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1077 030174  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1081 030176                                 ERRDF   ERRNO,T6SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     030176  104455                                                                                  TRAP    C$ERDF
     030200  001140                                                                                  .WORD   608
     030202  030475                                                                                  .WORD   T6SSR
     030204  012036                                                                                  .WORD   PKTSSR
1082 030206  005237  002220                 INC     FATFLG                  ;SET FATAL ERROR FLAG
1083 030212                          15$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     030212  104406                                                                                  TRAP    C$CLP1
1084 030214                                 ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
     030214  104410                                                                                  TRAP    C$ESCAPE
     030216  000170                                                                                  .WORD   L10054 .
1085 030220  005737  002222                 TST     INTRFCV                 ;DID AN INTERRUPT OCCUR ?
1086 030224  001004                         BNE     22$                     ;BRANCH IF YES
1090 030226                                 ERRHRD  ERRNO,T6NINT,PKTSSR     ;REPORT ERROR IF NO INTERRUPT
     030226  104456                                                                                  TRAP    C$ERHRD
     030230  001141                                                                                  .WORD   609
     030232  030631                                                                                  .WORD   T6NINT
     030234  012036                                                                                  .WORD   PKTSSR
1091 030236  012702  102206          22$:   MOV     #SC!NBA!SSR!TSREJ,R2    ;EXPECTED CONTENTS OF TSSR
1092 030242  004737  016326                 JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1093 030246  016501  000002                 MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1094 030252  032701  000100                 BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1095 030256  001402                         BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1096 030260  052702  000100                 BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1097 030264  020201                  25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1098 030266  001404                         BEQ     30$                     ;OKAY IF MATCH
1102 030270                                 ERRHRD  ERRNO,T6NBA,PKTSSR      ;NBA NOT SET TO REJECT
     030270  104456                                                                                  TRAP    C$ERHRD
     030272  001142                                                                                  .WORD   610
```

```
TEST  6: COMMAND REJECT

        030274  030450                                                          .WORD   T6NBA
        030276  012036                                                          .WORD   PKTSGR
   1103 030300                          30$:    CKLOOP                      ;LOOP ON ERROR ?
        030300  104406                                                          TRAP    C$CLP1
   1104 030302  004737  016326                   JSR     PC,CHKTSSR          ;WAIT FOR READY, NON-AMBIGUOUS
   1105 030306  016501  000000                   MOV     TSBA(R5),R1         ;GET TSBA REGISTER CONTENTS
   1106 030312  010402                           MOV     R4,R2               ;START OF THE PACKET
   1107 030314  062702  000010                   ADD     #10,R2              ;EXPECT TSDA TO PACKET + 10
   1108 030320  020102                           CMP     R1,R2               ;COMPARE EXPECTED TO RECEIVED
   1109 030322  001404                           BEQ     35$                 ;ERROR IF NOT EQUAL
   1113 030324                                   ERRHRD  ERRNO,T6TSBA,EXPREC ;PRINT THE ERROR & EXPD/RECV
        030324  104456                                                          TRAP    C$ERHRD
        030326  001143                                                          .WORD   611
        030330  030711                                                          .WORD   T6TSBA
        030332  015464                                                          .WORD   EXPREC
   1114
   1115
   1116 030334  004737  011104          35$:    JSR     PC,CKRAM            ;SEE IF DATA IN RAM IS CORRECT
   1117 030340  103404                           BCS     40$                 ;BRANCH IF PACKET IN RAM IS CORRECT
   1121 030342                                   ERRHRD  ERRNO,PKTRAM,RAMERR ;REPORT THE RAM ERROR(S)
        030342  104456                                                          TRAP    C$ERHRD
        030344  001144                                                          .WORD   612
        030346  004741                                                          .WORD   PKTRAM
        030350  015500                                                          .WORD   RAMERR
   1122 030352                          40$:    ENDSEG                      ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        030352                                                                  10000$:
        030352  104405                                                          TRAP    C$ESEG
   1123 030354  011300                           MOV     (R3),R0             ;NEXT PACKET COMMAND WORD
   1124 030356  042700  177740                   BIC     #177740,R0          ;GET BITS 0-4
   1125 030362  020027  000004                   CMP     R0,#4               ;DON'T TEST WRITE CHARACTERISTICS
   1126 030366  001002                           BNE     45$                 ;BRANCH IF NOT WRITE CHARACTERISTICS
   1127 030370  062703  000002                   ADD     #2,R3               ;BY-PASS WRITE CHARACTERISTCS
   1128 030374  020327  003056          45$:    CMP     R3,#TBLEND          ;HAVE WE COMPLETED DATA TABLE ?
   1129 030400  103002                           BHIS    50$                 ;BRANCH IF ALL TESTED
   1130 030402  000137  030116                   JMP     5$                  ;TEST WITH NEXT DATA
   1131
   1132 030406                          50$:    ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        030406                                                                  L10054:
        030406  104403                                                          TRAP    C$ESUB
   1133 030410  005737  002220                   TST     FATFLG              ;ANY FATAL ERRORS ?
   1134 030414  001402                           BEQ     60$                 ;BRANCH IF NOT
   1135 030416  004737  017172                   JSR     PC,CKDROP           ;TRY TO DROP THE UNIT
   1136 030422  004737  016446          60$:    JSR     PC,TSTLOOP          ;SHOULD WE DO ITERATIONS ?
   1137 030426  103002                           BCC     62$                 ;BRANCH IF NOT
   1138 030430  000137  027552                   JMP     T6LOOP              ;LOOP UNTIL COUNT EXPIRED
   1139 030434                          62$:    EXIT    TST                 ;ALL DONE THIS TEST
        030434  104432                                                          TRAP    C$EXIT
        030436  000344                                                          .WORD   L10052 .
   1140
   1141                                 ;+
   1142                                 ;LOCAL STORAGE FOR THIS TEST
   1143                                 ;-
   1144
   1148 030440                          T6PACKET:                           ;COMMAND PACKET FOR TEST
   1149 030440  000000                           .WORD   0                   ;WILL CONTAIN VARIABLE COMMANDS
   1150 030442  052525                           .WORD   052525
   1151 030444  125252                           .WORD   125252
```

```
   1152 030446  052525                          .WORD   052525
   1153
   1154
   1155                                 ;*
   1156                                 ;LOCAL TEXT MESSAGES FOR TEST
   1157                                 ;
   1158
   1159 030450      103     157     155  T6NBA:  .ASCIZ  'Command Not Rejected'
   1160 030475      103     157     156  T6SSR:  .ASCIZ  'Contents of TSSR Incorrect After Write Packet'
   1161 030553      125     156     145  T6INT:  .ASCIZ  'Unexpected Interrupt Received On Write Packet'
   1162 030631      105     170     160  T6NINT: .ASCIZ  'Expected Interrupt Not Received On Write Packet'
   1163 030711      111     156     143  T6TSBA: .ASCIZ  'Incorrect TSBA Address After Packet Write'
   1164 030763      103     157     155  TST6ID: .ASCIZ  'Command Reject'
   1165                                         .EVEN
   1166 031002                                  ENDTST
        031002                                                              L10052:
        031002  104401                                                            TRAP    C$ETST
```

```
1168
1169                                            .SBTTL  TEST  7: WRITE CHARACTERISTICS
1170
1171
1172                                    ;       THIS TEST VERIFIES BASIC OPERATION OF THE WRITE CHARACTERISTICS
1173                                    ;       COMMAND.  IT VERIFIES THAT THE COMMAND BLOCK AND CHARACTERISTICS
1174                                    ;       DATA BLOCK ARE FETCHED PROPERLY FROM CPU MEMORY, THE NEED BUFFER
1175                                    ;       ADDRESS (NBA) BIT IN TSSR IS HANDLED PROPERLY, AND THAT A PROPER
1176                                    ;       MESSAGE PACKET IS STORED, WHERE APPROPRIATE.  THIS TEST DOES NOT
1177                                    ;       CHECK THAT THE VARIOUS FUNCTIONS ENABLED BY CHARACTERISTIC MODE
1178                                    ;       DATA BITS OPERATE PROPERLY;  THE FUNCTIONING OF THESE BITS IS
1179                                    ;       VERIFIED IN SUBSEQUENT TESTS.  ALL COMMANDS EXECUTED IN THIS
1180                                    ;       TEST HAVE THE INTERRUPT ENABLE (IE) BIT CLEARED TO ZERO, SO NO
1181                                    ;       INTERRUPTS SHOULD BE GENERATED.  HOWEVER, THE PROGRAM RUNS AT
1182                                    ;       PROCESSOR PRIORITY 0, WITH THE INTERRUPT SERVICE ROUTINE SET UP
1183                                    ;       TO FLAG UNEXPECTED INTERRUPTS.  IF AN INTERRUPT OCCURS, A
1184                                    ;       PROBLEM EXISTS IN EITHER THE LSI-11 BUS INTERFACE SECTION OR IN
1185                                    ;       THE ROM OR PIPELINE.
1186                                    ;
1187                                    ;       THIS TEST CHECKS VARIOUS MICROPROGRAM SEQUENCES, COMMAND
1188                                    ;       DECODING, DMA LOGIC, AND BASIC PACKET PROTOCOL HANDLING.  THIS
1189                                    ;       IS THE FIRST TEST IN WHICH DATO DMA CYCLES (FOR STORING THE
1190                                    ;       MESSAGE PACKET) ARE PERFORMED.  ANY ERRORS IN THE BODY OF THE
1191                                    ;       TEST (I.E, ERRORS OTHER THAN INITIALIZATION ERRORS RELATED TO
1192                                    ;       THE TRANSPORT BUS) DEFINITELY INDICATE A BAD M7196 MODULE.
1193                                    ;
1194                                    ;
1195                                    ;
1196 031004                                     BGNTST
     031004                                                                     T7::
1201 031004  012700  034273                     MOV     @TST7ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
1202 031010  004737  016500                     JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
1203 031014  012737  000024  002214             MOV     @20.,LOOPCNT            ;PERFORM 20 ITERATIONS
1204 031022                          T7LOOP:
1205 031022                                     BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
     031022                                                                             T7.1:
     031022  104402                                                                     TRAP    C$BSUB
1206 031024  004737  034322                     JSR     PC,T7REST               ;SET PACKET TO START-UP VALUES
1207
1208 031030                                     SETPRI  @PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     031030  012700  000000                                                             MOV     @PRI00,R0
     031034  104441                                                                     TRAP    C$SPRI
1209 031036  012703  002762                     MOV     @TSTBLK+10.,R3          ;START OF TEST DATA
1210 031042  012704  033140                     MOV     @T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1211 031046  012764  000010  000006             MOV     @8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
1212 031054                          5$:
1213 031054                                     BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     031054  104404                                                                     TRAP    C$BSEG
1214
1215 031056  004737  015764                     JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1216 031062  103405                             BCS     10$                     ;BR IF SOFT INIT = OK
1220 031064  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1221 031066                                     ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     031066  104455                                                                     TRAP    C$ERDF
     031070  001275                                                                     .WORD   701
     031072  003646                                                                     .WORD   SFIERR
     031074  012024                                                                     .WORD   SFIMSG
```

TEST 7: WRITE CHARACTERISTICS

```
1222 031076  005037  002220          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1223 031102  005037  002222                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1224 031106  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
1225 031112  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1226 031116  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1227 031120  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1231 031122                                  ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     031122  104455                                                                  TRAP    C$ERDF
     031124  001276                                                                  .WORD   702
     031126  033741                                                                  .WORD   T7SSR
     031130  012036                                                                  .WORD   PKTSSR
1232 031132  005237  002220                  INC     FATFLG                  ;SET FATAL ERROR FLAG
1233 031136                          15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     031136  104406                                                                  TRAP    C$CLP1
1234 031140                                  ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     031140  104410                                                                  TRAP    C$ESCAPE
     031142  000152                                                                  .WORD   10000$ .
1235 031144  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1236 031150  001404                          BEQ     22$                     ;BRANCH IF NOT
1240 031152                                  ERRHRD  ERRNO,T7INT,PKTSSR
     031152  104456                                                                  TRAP    C$ERHRD
     031154  001277                                                                  .WORD   703
     031156  034121                                                                  .WORD   T7INT
     031160  012036                                                                  .WORD   PKTSSR
1241 031162  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1242 031166  012702  000200                  MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
1243 031172  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1244 031176  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1245 031200  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1246 031204  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1247 031206  001404                          BEQ     30$                     ;OKAY IF MATCH
1251 031210                                  ERRHRD  ERRNO,T7NBA,PKTSSR      ;NBA NOT ZERO
     031210  104456                                                                  TRAP    C$ERHRD
     031212  001300                                                                  .WORD   704
     031214  033300                                                                  .WORD   T7NBA
     031216  012036                                                                  .WORD   PKTSSR
1252 031220                          30$:    CKLOOP                          ;LOOP ON ERROR ?
     031220  104406                                                                  TRAP    C$CLP1
1253 031222  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1254 031226  016501  000000                  MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
1255 031232  012702  033166                  MOV     #T7BFR,R2               ;START OF THE DATA BUFFER
1256 031236  032762  000200  000012          BIT     #BIT7,XST2(R2)          ;IS EXTENDED FEATURES BIT SET ?
1257 031244  001404                          BEQ     32$                     ;BRANCH IF EXTENDED FEATURES OFF
1258 031246  005237  002224                  INC     EXTFEA                  ;SET EXTENDED FEATURES FOR SUBTEST 6
1259 031252  062702  000002                  ADD     #2,R2                   ;EXTRA WORD IF SPECIAL FEATURES
1260 031256  062702  000016          32$:    ADD     #14.,R2                 ;EXPECTED CONTENTS OF TSDA
1261 031262  020102                          CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1262 031264  001404                          BEQ     35$                     ;ERROR IF NOT EQUAL
1266 031266                                  ERRHRD  ERRNO,T7TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
     031266  104456                                                                  TRAP    C$ERHRD
     031270  001301                                                                  .WORD   705
     031272  034210                                                                  .WORD   T7TSBA
     031274  015464                                                                  .WORD   EXPREC
1267
1268
1269 031276  004737  011104          35$:    JSR     PC,CKRAM                ;SEE IF DATA IN RAM IS CORRECT
1270 031302  103404                          BCS     40$                     ;BRANCH IF PACKET IN RAM IS CORRECT
```

J12

TEST 7: WRITE CHARACTERISTICS

SEQ 0152

```
1274 031304
     031304  104456                ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
     031306  001302                                                        TRAP    C$ERHRD
     031310  004741                                                        .WORD   706
     031312  015500                                                        .WORD   PKTRAM
1275                                                                       .WORD   RAMERR
1276 031314
     031314                  40$:    ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     031314  104405                                                      10000$:
1277                                                                       TRAP    C$ESEG
1278 031316  012364  000006          MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
1279 031322  020327  003056          CMP     R3,#TBLEND              ;HAS ALL DATA BEEN TESTED ?
1280 031326  103002                  BHIS    55$                     ;BRANCH IF ALL DATA DONE
1281 031330  000137  031054          JMP     5$                      ;BRANCH TILL BACK TO ZERO
1282
1283 031334                  55$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     031334                                                          L10056:
     031334  104403                                                        TRAP    C$ESUB
1284
1285 031336  005737  002220          TST     FATFLG                  ;ANY FATAL ERRORS ?
1286 031342  001402                  BEQ     60$                     ;BRANCH IF NOT
1287 031344  004737  017172          JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
1288 031350                  60$:
```

TEST 7: WRITE CHARACTERISTICS

```
1290                                    ;+
1291                                    ;
1292                                    ;TEST 7, SUBTEST 2
1293                                    ;
1294                                    ;CHECK THAT UNUSED BITS BEING SET CAUSES
1295                                    ;WRITE CHARACTERISTICS COMMAND TO BE REJECTED
1296                                    ;
1297                                    ;-
1298
1299 031350                                    BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     031350                                                                                    T7.2:
     031350  104402                                                                                   TRAP    C$BSUB
1300
1301 031352                                    SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     031352  012700  000000                                                                           MOV     #PRI00,R0
     031356  104441                                                                                   TRAP    C$SPRI
1302 031360  012703  033206                    MOV     #T72DATA,R3             ;START OF TEST DATA FOR SUBTEST
1303 031364  012704  033140              5$:   MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1304 031370  004737  034322                    JSR     PC,T7REST               ;RESTORE PACKET TO STARTING VALUES
1305
1306 031374                                    BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     031374  104404                                                                                   TRAP    C$BSEG
1307
1308 031376  004737  015764                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1309 031402  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
1313 031404  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1314 031406                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     031406  104455                                                                                   TRAP    C$ERDF
     031410  001303                                                                                   .WORD   707
     031412  003646                                                                                   .WORD   SFIERR
     031414  012024                                                                                   .WORD   SFIMSG
1315 031416  005037  002222              10$:  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1316 031422  010400                          MOV     R4,R0                   ;START OF THE COMMAND PACKET
1317 031424  061300                          ADD     (R3),R0                 ;OFFSET TO THE DATA WORD TO TEST
1318 031426  056310  000002                  BIS     2(R3),(R0)              ;SET THE DATA BITS TO BE TESTED
1319 031432  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
1320 031436  004737  016240                  JSR     PC,WAITF                ;WAIT FOR SSR TO SET
1321 031442  103405                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1322 031444  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1326 031446                                    ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     031446  104455                                                                                   TRAP    C$ERDF
     031450  001304                                                                                   .WORD   708
     031452  033741                                                                                   .WORD   T7SSR
     031454  012036                                                                                   .WORD   PKTSSR
1327 031456                              15$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     031456  104406                                                                                   TRAP    C$CLP1
1328 031460                                    ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
     031460  104410                                                                                   TRAP    C$ESCAPE
     031462  000116                                                                                   .WORD   10000$-.
1329 031464  005737  002222                    TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1330 031470  001404                          BEQ     22$                     ;BRANCH IF NOT
1334 031472                                    ERRHRD  ERRNO,T7INT,PKTSSR
     031472  104456                                                                                   TRAP    C$ERHRD
     031474  001305                                                                                   .WORD   709
     031476  034121                                                                                   .WORD   T7INT
     031500  012036                                                                                   .WORD   PKTSSR
1335 031502  016501  000002              22$:  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
```

TEST 7: WRITE CHARACTERISTICS

```
1336 031506  012702  102206                MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
1337 031512  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
1338 031516  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1339 031520  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1340 031524  020201                25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1341 031526  001414                        BEQ     30$                     ;OKAY IF MATCH
1342 031530  010100                        MOV     R1,R0                   ;DATA FROM TSSR
1343 031532                                XOR     R2,R0                   ;FIND BITS IN ERROR
1344 031542  020027  002000                CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
1345 031546  001404                        BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
1349 031550                                ERRHRD  ERRNO,T72REJ,PKTSSR     ;COMMAND NOT REJECTED
     031550  104456                                                                TRAP    C$ERHRD
     031552  001306                                                                .WORD   710
     031554  033353                                                                .WORD   T72REJ
     031556  012036                                                                .WORD   PKTSSR
1350 031560                        30$:    CKLOOP                          ;LOOP ON ERROR ?
     031560  104406                                                                TRAP    C$CLP1
1351 031562  032701  002000                BIT     #NBA,R1                 ;IS NBA BIT SET ?
1352 031566  001004                        BNE     35$                     ;OKAY IF NBA SET
1356 031570                                ERRHRD  ERRNO,T72NBA,PKTSSR     ;NBA NOT SET
     031570  104456                                                                TRAP    C$ERHRD
     031572  001307                                                                .WORD   711
     031574  033222                                                                .WORD   T72NBA
     031576  012036                                                                .WORD   PKTSSR
1357 031600                        35$:
1358 031600                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     031600                                                                  10000$:
     031600  104405                                                                TRAP    C$ESEG
1359
1360 031602  062703  000004                ADD     #4,R3                   ;POINT TO NEXT DATA PAIR
1361 031606  020327  033222                CMP     R3,#T72DONE             ;COMPARE TO END OF TEST DATA
1362 031612  103002                        BHIS    57$                     ;BRANCH IF ALL DATA TESTED
1363 031614  000137  031364                JMP     5$                      ;BRANCH TILL BACK TO ZERO
1364
1365 031620                        57$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     031620                                                                  L10057:
     031620  104403                                                                TRAP    C$ESUB
```

TEST 7: WRITE CHARACTERISTICS

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

                                                BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
031622                                                                                                  T7.3:
031622  104402                                                                                                  TRAP    C$BSUB

                                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
031624  012700  000000                                                                                          MOV     #PRI00,R0
031630  104441                                                                                                  TRAP    C$SPRI
031632  012703  000001                        MOV     #1,R3                   ;STARTING BYTE COUNT
031636  012704  033140                5$:     MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
031642  004737  034322                        JSR     PC,T7REST               ;RESTORE PACKET TO STARTING VALUES

031646                                          BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT <<<<<<<<<<<<<
031646  104404                                                                                                  TRAP    C$BSEG

031650  004737  015764                        JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
031654  103405                                BCS     10$                     ;BR IF SOFT INIT = OK
031656  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
031660                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
031660  104455                                                                                                  TRAP    C$ERDF
031662  001310                                                                                                  .WORD   712
031664  003646                                                                                                  .WORD   SFIERR
031666  012024                                                                                                  .WORD   SFIMSG

031670  005037  002222                10$:     CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
031674  010364  000006                        MOV     R3,PKBCNT(R4)           ;INSERT THE BYTE COUNT FOR TEST
031700  010465  000000                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
031704  004737  016240                        JSR     PC,WAITF                ;WAIT FOR SSR TO SET
031710  103405                                BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
031712  010001                                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
031714                                          ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
031714  104455                                                                                                  TRAP    C$ERDF
031716  001311                                                                                                  .WORD   713
031720  033741                                                                                                  .WORD   T7SSR
031722  012036                                                                                                  .WORD   PKTSSR
031724                                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
031724  104406                                                                                                  TRAP    C$CLP1
C31726                                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
031726  104410                                                                                                  TRAP    C$ESCAPE
031730  000116                                                                                                  .WORD   10000$-.
031732  005737  002222                        TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
031736  001404                                BEQ     22$                     ;BRANCH IF NOT
031740                                          ERRHRD  ERRNO,T7INT,PKTSSR
031740  104456                                                                                                  TRAP    C$ERHRD
031742  001312                                                                                                  .WORD   714
031744  034121                                                                                                  .WORD   T7INT
031746  012036                                                                                                  .WORD   PKTSSR
031750  016501  000002                22$:     MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
031754  012702  102206                        MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
031760  032701  000100                        BIT     #OFL,R1                 ;IS OFF LINE BIT SET ?
```

```
1413 031764  001402                    BEQ     25$                 ;BRANCH IF NOT OFF-LINE
1414 031766  052702  000100            BIS     #OFL,R2             ;SET OFF-LINE IN EXPECTED DATA
1415 031772  020201              25$:  CMP     R2,R1               ;DOES EXPECTED MATCH RECEIVED ?
1416 031774  001414                    BEQ     30$                 ;OKAY IF MATCH
1417 031776  010100                    MOV     R1,R0               ;DATA FROM TSSR
1418 032000                            XOR     R2,R0               ;FIND BITS IN ERROR
1419 032010  020027  002000            CMP     R0,#NBA             ;IS NBA ONLY BIT IN ERROR ?
1420 032014  001404                    BEQ     30$                 ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
1424 032016                            ERRHRD  ERRNO,T73REJ,PKTSSR ;COMMAND NOT REJECTED
     032016  104456                                                        TRAP    C$ERHRD
     032020  001313                                                        .WORD   715
     032022  033452                                                        .WORD   T73REJ
     032024  012036                                                        .WORD   PKTSSR
1425 032026              30$:  CKLOOP                              ;LOOP ON ERROR ?
     032026  104406                                                        TRAP    C$CLP1
1426 032030  032701  002000            BIT     #NBA,R1             ;IS NBA BIT SET ?
1427 032034  001004                    BNE     35$                 ;OKAY IF NBA SET
1431 032036                            ERRHRD  ERRNO,T72NBA,PKTSSR ;NBA NOT SET
     032036  104456                                                        TRAP    C$ERHRD
     032040  001314                                                        .WORD   716
     032042  033222                                                        .WORD   T72NBA
     032044  012036                                                        .WORD   PKTSSR
1432 032046              35$:
1433 032046                            ENDSEG                      ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     032046                                                    10000$:
     032046  104405                                                        TRAP    C$ESEG
1434
1435 032050  005203                    INC     R3                  ;NEXT BYTE COUNT
1436 032052  020327  000006            CMP     R3,#6               ;TESTED ALL INVALID ?
1437 032056  002002                    BGE     57$                 ;BRANCH IF TEST DONE
1438 032060  000137  031636            JMP     5$                  ;BRANCH TILL BACK TO ZERO
1439
1440 032064              57$:  ENDSUB                              ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     032064                                                    L10060:
     032064  104403                                                        TRAP    C$ESUB
```

TEST 7: WRITE CHARACTERISTICS

```
1442                                    ;*
1443                                    ;
1444                                    ;TEST 7, SUBTEST 4
1445                                    ;
1446                                    ;SUBTEST TO VERIFY THAT A WRITE CHARACTERISTICS COMMAND IS
1447                                    ;REJECTED IF AN ILLEGAL DATA BLOCK ADDRESS IS ISSUED.
1448                                    ;
1449                                    ;-
1450
1451 032066                             BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     032066                                                                     T7.4:
     032066  104402                                                             TRAP    C$BSUB
1452
1453 032070                             SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     032070  012700  000000                                                     MOV     #PRI00,R0
     032074  104441                                                             TRAP    C$SPRI
1454 032076  012703  033206            MOV     #T72DATA,R3             ;START OF TEST DATA FOR SUBTEST
1455 032102  012704  033140    5$:     MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1456 032106  004737  034322            JSR     PC,T7REST               ;RESTORE PACKET TO STARTING VALUES
1457
1458
1459 032112  004737  015764            JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1460 032116  103405                    BCS     10$                     ;BR IF SOFT INIT = OK
1464 032120  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1465 032122                            ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     032122  104455                                                             TRAP    C$ERDF
     032124  001315                                                             .WORD   717
     032126  003646                                                             .WORD   SFIERR
     032130  012024                                                             .WORD   SFIMSG
1466 032132  005037  002222    10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1467 032136  052737  000001  033150    BIS     #1,T7DATA               ;MAKE ADDRESS ODD
1468 032144  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
1469 032150  004737  016240            JSR     PC,WAITF                ;WAIT FOR SSR TO SET
1470 032154  103405                    BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1471 032156  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1475 032160                            ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     032160  104455                                                             TRAP    C$ERDF
     032162  001316                                                             .WORD   718
     032164  033741                                                             .WORD   T7SSR
     032166  012036                                                             .WORD   PKTSSR
1476 032170                    15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     032170  104406                                                             TRAP    C$CLP1
1477 032172                            ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
     032172  104410                                                             TRAP    C$ESCAPE
     032174  000116                                                             .WORD   L10061-.
1478 032176  005737  002222            TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1479 032202  001404                    BEQ     22$                     ;BRANCH IF NOT
1483 032204                            ERRHRD  ERRNO,T7INT,PKTSSR
     032204  104456                                                             TRAP    C$ERHRD
     032206  001317                                                             .WORD   719
     032210  034121                                                             .WORD   T7INT
     032212  012036                                                             .WORD   PKTSSR
1484 032214  016501  000002    22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1485 032220  012702  102206            MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
1486 032224  032701  000100            BIT     #OFL,R1                 ;IS OFF LINE BIT SET ?
1487 032230  001402                    BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1488 032232  052702  000100            BIS     #OFL,R2                 ;SET OFF LINE IN EXPECTED DATA
```

```
 1489 032236  020201                25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 1490 032240  001414                        BEQ     30$                     ;OKAY IF MATCH
 1491 032242  010100                        MOV     R1,R0                   ;DATA FROM TSSR
 1492 032244                                XOR     R2,R0                   ;FIND BITS IN ERROR
 1493 032254  020027  002000                CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
 1494 032260  001404                        BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
 1498 032262                                ERRHRD  ERRNO,T74REJ,PKTSSR     ;COMMAND NOT REJECTED
      032262  104456                                                                TRAP    C$ERHRD
      032264  001320                                                                .WORD   720
      032266  033545                                                                .WORD   T74REJ
      032270  012036                                                                .WORD   PKTSSR
 1499 032272                        30$:    CKLOOP                          ;LOOP ON ERROR ?
      032272  104406                                                                TRAP    C$CLP1
 1500 032274  032701  002000                BIT     #NBA,R1                 ;IS NBA BIT SET ?
 1501 032300  001004                        BNE     35$                     ;OKAY IF NBA SET
 1505 032302                                ERRHRD  ERRNO,T72NBA,PKTSSR     ;NBA NOT SET
      032302  104456                                                                TRAP    C$ERHRD
      032304  001321                                                                .WORD   721
      032306  033222                                                                .WORD   T72NBA
      032310  012036                                                                .WORD   PKTSSR
 1506
 1507 032312                        35$:    ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
      032312                                                                L10061:
      032312  104403                                                                TRAP    C$ESUB
 1508
```

TEST 7: WRITE CHARACTERISTICS

```
1510                                    ;*
1511
1512                                    ;TEST 7, SUBTEST 5
1513                                    ;
1514                                    ;CHECK THAT WRITE CHARACTERISTICS COMMAND IS REJECTED IF THE
1515                                    ;MESSAGE BUFFER LENGTH SPECIFIES AN INVALID COUNT (LESS THAN 14)
1516                                    ;
1517                                    ;-
1518
1519 032314                                     BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
      032314                                                                                    T7.5:
      032314  104402                                                                            TRAP    C$BSUB
1520
1521 032316                                     SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      032316  012700  000000                                                                    MOV     #PRI00,R0
      032322  104441                                                                            TRAP    C$SPRI
1522 032324  012703  000001                     MOV     #1,R3                   ;STARTING BUFFER LENGTH
1523 032330  012704  033140             5$:     MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1524 032334  004737  034322                     JSR     PC,T7REST               ;RESTORE PACKET TO STARTING VALUES
1525
1526 032340                                     BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      032340  104404                                                                            TRAP    C$BSEG
1527
1528 032342  004737  015764                     JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1529 032346  103405                             BCS     10$                     ;BR IF SOFT INIT = OK
1533 032350  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1534 032352                                     ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      032352  104455                                                                            TRAP    C$ERDF
      032354  001322                                                                            .WORD   722
      032356  003646                                                                            .WORD   SFIERR
      032360  012024                                                                            .WORD   SFIMSG
1535 032362  005037  002222             10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1536 032366  010337  033154                     MOV     R3,T7DATA+4             ;INSERT THE BAD MESSAGE LENGTH
1537 032372  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
1538 032376  004737  016240                     JSR     PC,WAITF                ;WAIT FOR SSR TO SET
1539 032402  103405                             BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1540 032404  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1544 032406                                     ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      032406  104455                                                                            TRAP    C$ERDF
      032410  001323                                                                            .WORD   723
      032412  033741                                                                            .WORD   T7SSR
      032414  012036                                                                            .WORD   PKTSSR
1545 032416                             15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      032416  104406                                                                            TRAP    C$CLP1
1546 032420                                     ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
      032420  104410                                                                            TRAP    C$ESCAPE
      032422  000116                                                                            .WORD   10000$ .
1547 032424  005737  002222                     TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1548 032430  001404                             BEQ     22$                     ;BRANCH IF NOT
1552 032432                                     ERRHRD  ERRNO,T7INT,PKTSSR
      032432  104456                                                                            TRAP    C$ERHRD
      032434  001324                                                                            .WORD   724
      032436  034121                                                                            .WORD   T7INT
      032440  012036                                                                            .WORD   PKTSSR
1553 032442  016501  000002             22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1554 032446  012702  102206                     MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
1555 032452  032701  000100                     BIT     #OFL,R1                 ;IS OFF LINE BIT SET ?
```

TEST 7: WRITE CHARACTERISTICS

```
1556 032456  001402                        BEQ     25$                     ;BRANCH IF NOT OFF LINE
1557 032460  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1558 032464  020201                 25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1559 032466  001414                        BEQ     30$                     ;OKAY IF MATCH
1560 032470  010100                        MOV     R1,R0                   ;DATA FROM TSSR
1561 032472                                XOR     R2,R0                   ;FIND BITS IN ERROR
1562 032502  020027  002000                CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
1563 032506  001404                        BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
1567 032510                                ERRHRD  ERRNO,T75REJ,PKTSSR     ;COMMAND NOT REJECTED
     032510  104456                                                                TRAP    C$ERHRD
     032512  001325                                                                .WORD   725
     032514  033643                                                                .WORD   T75REJ
     032516  012036                                                                .WORD   PKTSSR
1568 032520                         30$:    CKLOOP                          ;LOOP ON ERROR ?
     032520  104406                                                                TRAP    C$CLP1
1569 032522  032701  002000                BIT     #NBA,R1                 ;IS NBA BIT SET ?
1570 032526  001004                        BNE     35$                     ;OKAY IF NBA SET
1574 032530                                ERRHRD  ERRNO,T72NBA,PKTSSR     ;NBA NOT SET
     032530  104456                                                                TRAP    C$ERHRD
     032532  001326                                                                .WORD   726
     032534  033222                                                                .WORD   T72NBA
     032536  012036                                                                .WORD   PKTSSR
1575 032540                         35$:
1576 032540                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     032540                                                                 10000$:
     032540  104405                                                                TRAP    C$ESEG
1577
1578 032542  005203                        INC     R3                      ;NEXT BUFFER LENGTH
1579 032544  020327  000016                CMP     R3,#14.                 ;HAVE ALL BAD VALUES BEEN TESTED ?
1580 032550  002002                        BGE     57$                     ;BRANCH IF ALL TESTED
1581 032552  000137  032330                JMP     5$                      ;BRANCH TILL BACK TO ZERO
1582
1583 032556                         57$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     032556                                                                 L10062:
     032556  104403                                                                TRAP    C$ESUB
```

TEST 7: WRITE CHARACTERISTICS

```
1585                                    ;*
1586                                    ;
1587                                    ;TEST 7, SUBTEST 6
1588                                    ;
1589                                    ;THIS SUBTEST IS EXECUTED ONLY IF THE EXTENDED
1590                                    ;FEATURES MODE IS ENABLED (AS DETERMINED BY EXAMINING
1591                                    ;XST2 AFTER A PREVIOUS EXECUTION OF WRITE CHARACTERISTICS).
1592                                    ;IT VERIFIES THAT A FIFTH CHARACTERISTICS DATA WORD IS FETCHED
1593                                    ;IF THE BYTE COUNT PARAMETER IN THE COMMAND PACKET IS 10 DECIMAL
1594                                    ;OR GREATER.
1595                                    ;
1596                                    ;-
1597 032560                                    BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
      032560                                                                                      T7.6:
      032560  104402                                                                                TRAP    C$BSUB
1598 032562  005737  002224                    TST     EXTFEA                  ;IS EXTENDED FEATURES SOFT. SW SET?
1599 032566  001002                            BNE     4$                      ;BR. IF SOFTWARE SWITCH IS SET (ON)
1600 032570                                    EXIT    TST                     ;NO EXTENDED FEATURES EXIT THIS TEST
      032570  104432                                                                                TRAP    C$EXIT
      032572  001576                                                                                .WORD   L10055 .
1601 032574  004737  034322            4$:     JSR     PC,T7REST               ;SET PACKET TO START-UP VALUES
1602
1603 032600                                    SETPRI  @PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      032600  012700  000000                                                                          MOV     @PRI00,R0
      032604  104441                                                                                 TRAP    C$SPRI
1604 032606  012703  002764                    MOV     @T$TBLK+12.,R3          ;START OF TEST DATA
1605 032612  012704  033140                    MOV     @T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1606 032616  012737  000200  033160            MOV     @200,T7SP               ;SPECIAL BIT SET FOR EXTFEA RAM RD
1607 032624  012764  000012  000006            MOV     @10.,PKBCNT(R4)         ;START WITH EXTENDED FEATURES VALUE
1608 032632                            5$:
1609 032632                                    BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      032632  104404                                                                                TRAP    C$BSEG
1610
1611 032634  004737  015764                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1612 032640  103405                            BCS     10$                     ;BR IF SOFT INIT = OK
1616 032642  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1617 032644                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      032644  104455                                                                                TRAP    C$ERDF
      032646  001327                                                                                .WORD   727
      032650  003646                                                                                .WORD   SFIERR
      032652  012024                                                                                 WORD   SFIMSG
1618 032654  005037  002220            10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1619 032660  005037  002222                    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1620 032664  010465  000000                    MOV     R4,T$DB(R5)             ;SET THE PACKET ADDRESS
1621 032670  004737  016326                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1622 032674  103407                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1623 032676  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1627 032700                                    ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      032700  104455                                                                                TRAP    C$ERDF
      032702  001330                                                                                .WORD   728
      032704  033741                                                                                .WORD   T7SSR
      032706  012036                                                                                .WORD   PKTSSR
1628 032710  005237  002220                    INC     FATFLG                  ;SET FATAL ERROR FLAG
1629 032714                            15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      032714  104406                                                                                TRAP    C$CLP1
1630 032716                                    ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
      032716  104410                                                                                TRAP    C$ESCAPE
```

```
                032720  000156
1631            032722  005737  002222          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
1632            032726  001404                  BEQ     22$                     ;BRANCH IF NOT
1636            032730                          ERRHRD  ERRNO,T7INT,PKTSSR
                032730  104456                                                  TRAP    C$ERHRD
                032732  001331                                                  .WORD   729
                032734  034121                                                  .WORD   T7INT
                032736  012036                                                  .WORD   PKTSSR
1637            032740  016501  000002  22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
1638            032744  012702  000200          MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
1639            032750  032701  000100          BIT     #OFL,R1                 ;IS OFF LINE BIT SET ?
1640            032754  001402                  BEQ     25$                     ;BRANCH IF NOT OFF-LINE
1641            032756  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
1642            032762  020201          25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
1643            032764  001404                  BEQ     30$                     ;OKAY IF MATCH
1647            032766                          ERRHRD  ERRNO,T7NBA,PKTSSR      ;NBA NOT ZERO
                032766  104456                                                  TRAP    C$ERHRD
                032770  001332                                                  .WORD   730
                032772  033300                                                  .WORD   T7NBA
                032774  012036                                                  .WORD   PKTSSR
1648            032776          30$:    CKLOOP                                  ;LOOP ON ERROR ?
                032776  104406                                                  TRAP    C$CLP1
1649            033000  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
1650            033004  016501  000000          MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
1651            033010  012702  033166          MOV     #T7BFR,R2               ;START OF THE DATA BUFFER
1652            033014  032762  000200  000012  BIT     #BIT7,XST2(R2)          ;IS EXTENDED FEATURES BIT SET ?
1653            033022  001402                  BEQ     32$                     ;BRANCH IF EXTENDED FEATURES OFF
1654            033024  062702  000002          ADD     #2,R2                   ;EXTRA WORD IF SPECIAL FEATURES
1655            033030  062702  000016  32$:    ADD     #14.,R2                 ;EXPECTED CONTENTS OF TSDA
1656            033034  020102                  CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
1657            033036  001404                  BEQ     35$                     ;ERROR IF NOT EQUAL
1661            033040                          ERRHRD  ERRNO,T7TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
                033040  104456                                                  TRAP    C$ERHRD
                033042  001333                                                  .WORD   731
                033044  034210                                                  .WORD   T7TSBA
                033046  015464                                                  .WORD   EXPREC
1662
1663
1664            033050  012704  033150  35$:    MOV     #T7DATA,R4              ;SET POINTER FOR CHECKER
1665            033054  004737  011214          JSR     PC,CKRAM2               ;SEE IF DATA IN RAM IS CORRECT
1666            033060  103404                  BCS     40$                     ;BRANCH IF PACKET IN RAM IS CORRECT
1670            033062                          ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
                033062  104456                                                  TRAP    C$ERHRD
                033064  001334                                                  .WORD   732
                033066  004741                                                  .WORD   PKTRAM
                033070  015500                                                  .WORD   RAMERR
1671
1672            033072  012704  033140  40$:    MOV     #T7PACKET,R4            ;RESET PACKET POINTER
1673            033076                          ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                033076                                                  10000$:
                033076  104405                                                  TRAP    C$ESEG
1674
1675            033100  012364  000006          MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
1676            033104  020327  003056          CMP     R3,#TBLEND              ;HAS ALL DATA BEEN TESTED ?
1677            033110  103002                  BHIS    55$                     ;BRANCH IF ALL DATA DONE
1678            033112  000137  032632          JMP     5$                      ;BRANCH TILL BACK TO ZERO
1679
```

TEST 7: WRITE CHARACTERISTICS

```
1680 033116                          55$:    ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     033116                                                                                L10063:
     033116  104403                                                                               TRAP    C$ESUB
1681
1682 033120  005737  002220                  TST     FATFLG                  ;ANY FATAL ERRORS ?
1683 033124  001402                          BEQ     60$                     ;BRANCH IF NOT
1684 033126  004737  017172                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
1685 033132                          60$:
1686 033132                                  EXIT    TST                     ;ALL DONE THIS TEST
     033132  104432                                                                               TRAP    C$EXIT
     033134  001234                                                                               .WORD   L10055-.
1687
1688                                 ;*
1689                                 ;LOCAL STORAGE FOR THIS TEST
1690                                 ;
1691
1693         033140                          .=<.+10>&177770
1695 033140                          T7PACKET:
1696 033140  100004                          .WORD   100004                  ;COMMAND PACKET FOR TEST
1697 033142  033150                          .WORD   T7DATA                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
1698 033144  000000                          .WORD   0                       ;ADDRESS OF CHARACTERISTICS BLOCK
1699 033146  000010                          .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
1700
1701 033150                          T7DATA:                                 ;CHARACTERISTICS DATA BLOCK
1702 033150  033166                          .WORD   T7BFR                   ;ADDRESS OF MESSAGE BUFFER
1703 033152  000000                          .WORD   0
1704 033154  000016                          .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
1705 033156  000000                          .WORD   0
1706 033160  000000                  T7SP:   .WORD   0                       ;EXTFEA EXTRA WORD
1707
1708 033162  000000  000000                  .WORD   0,0                             ;SPACE
1709 033166                          T7BFR:  .BLKW   8.                      ;MESSAGE BUFFER
1710
1711                                 ;*
1712                                 ;
1713                                 ;TEST DATA FOR SUBTEST TWO
1714                                 ;
1715                                 ;DATA HAS FORMAT:
1716                                 ;
1717                                 ;       1ST WORD        OFFSET TO TEST WORD IN PACKET
1718                                 ;       2ND WORD        BITS TO SET FOR TEST
1719                                 ;
1720                                 ;-
1721
1722 033206                          T72DATA:
1723 033206  000000  037140                  .WORD   0,BIT5!BIT6!BIT9!BIT10!BIT11!BIT12!BIT13
1724 033212  000002  000001                  .WORD   2,BIT0
1725 033216  000004  100100                  .WORD   4,BIT6!BIT15
1726         033222                  T72DONE=.
1727
1728
1729                                 ;*
1730                                 ;LOCAL TEXT MESSAGES FOR TEST
1731                                 ;
1732
1733 033222     116     102     101  T72NBA: .ASCIZ  'NBA Not Set On Rejected WRITE CHARACTERISTICS'
1734 033300     127     122     111  T7NBA:  .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
```

TEST 7: WRITE CHARACTERISTICS

```
1735 033353     127     122     111  T72REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Non-Zero Unused Fields'
1736 033452     127     122     111  T73REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Data Count'
1737 033545     127     122     111  T74REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Block Address'
1738 033643     127     122     111  T75REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Buffer Length'
1739 033741     103     157     156  T7SSR:  .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
1740 034030     105     170     160  T7NINT: .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
1741 034121     125     156     145  T7INT:  .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
1742 034210     111     156     143  T7TSBA: .ASCIZ  'Incorrect TSBA Address After WRITE CHARACTERISTICS'
1743 034273     127     162     151  TST7ID: .ASCIZ  'Write Characteristics'
1744                                          .EVEN
1745
```

TEST 7: WRITE CHARACTERISTICS

```
1747
1748                                    ;+
1749                                    ;
1750                                    ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
1751                                    ;
1752                                    ;-
1753
1754 034322                             T7REST:
1755 034322                                     SAVREG                          ;SAVE THE REGISTERS
1756 034326  012701  033140                       MOV     #T7PACKET,R1        ;START OF THE PACKET
1757 034332  012721  100004                       MOV     #100004,(R1)+       ;WRITE CHARACTERISTICS WITH ACK
1758 034336  012721  033150                       MOV     #T7DATA,(R1)+       ;ADDRESS OF CHAR DATA BLOCK
1759 034342  005021                               CLR     (R1)+               ;EXTENDED ADDRESS
1760 034344  012721  000010                       MOV     #8.,(R1)+           ;SIZE OF DATA BLOCK IN BYTES
1761 034350  012721  033166                       MOV     #T7BFR,(R1)+        ;ADDRESS OF MESSAGE BUFFER
1762 034354  005021                               CLR     (R1)+
1763 034356  012721  000020                       MOV     #16.,(R1)+          ;LENGTH OF MESSAGE BUFFER
1764 034362  005021                               CLR     (R1)+
1765 034364  005011                               CLR     (R1)
1766 034366  000207                               RTS     PC                  ;RETURN
1767 034370                                     ENDTST
     034370                                                                    L10055:
     034370  104401                                                                    TRAP    C$ETST
1768
```

```
1770                                    .SBTTL  TEST  8: VOLUME CHECK
1771
1772
1773                            ;       THIS TEST VERIFIES THAT THE VOLUME CHECK (VCK) BIT, A FLAG HELD
1774                            ;       WITHIN THE M7196 AND APPEARING IN XSTO, IS SET BY INITIALIZE AND
1775                            ;       CLEARED BY EXECUTING A WRITE CHARACTERISTICS COMMAND WITH THE
1776                            ;       CVC BIT SET.  IT IS ALSO VERIFIED THAT A WRITE CHARACTERISTICS
1777                            ;       COMMAND WITH THE CVC BIT CLEAR DOES NOT AFFECT THE STATE OF THE
1778                            ;       VOLUME CHECK BIT.  THE ACTUAL FUNCTION OF VOLUME CHECK, THAT OF
1779                            ;       PREVENTING OR ALLOWING A TAPE MOTION COMMAND DEPENDING UPON
1780                            ;       WHETHER VOLUME CHECK IS SET OR CLEAR, IS NOT CHECKED BY THIS
1781                            ;       TEST;  THIS FUNCTIONALITY IS CHECKED IN THE INDIVIDUAL TESTS OF
1782                            ;       TAPE MOTION COMMANDS.
1783                            ;
1784                            ;       THE TEST PROCEEDS AS FOLLOWS:
1785                            ;
1786                            ;            1.  THE CONTROLLER IS INITIALIZED BY WRITING INTO THE TSSR.
1787                            ;
1788                            ;            2.  A WRITE CHARACTERISTICS COMMAND IS ISSUED (WITH CVC=0)
1789                            ;                AND XSTO IN THE RETURNED MESSAGE BUFFER IS EXAMINED;
1790                            ;                THE VCK BIT SHOULD BE CLEAR (0).
1791                            ;
1792                            ;            3.  THE PREVIOUS STEP IS REPEATED TO VERIFY THAT VCK DOES
1793                            ;                NOT CHANGE (REMAINS AT 0).
1794                            ;
1795                            ;            4.  A WRITE CHARACTERISTICS COMMAND IS ISSUED WITH CVC=1
1796                            ;                AND THE VCK BIT IN XSTO IN THE MESSAGE BUFFER IS
1797                            ;                EXAMINED;  THE VCK BIT SHOULD BE CLEAR (0).
1798                            ;
1799                            ;            5.  A WRITE CHARACTERISTICS COMMAND IS ISSUED WITH CVC=0
1800                            ;                AND THE VCK BIT IN XSTO IN THE MESSAGE BUFFER IS
1801                            ;                EXAMINED;  THE VCK BIT SHOULD REMAIN CLEAR (0).
1802                            ;
1803                            ;
1804 034372                            BGNTST
      034372                                                                          T8::
1809 034372  012700  035257            MOV     #TST8ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
1810 034376  004737  016500            JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
1811 034402  012737  000024  002214    MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
1812 034410                    T8LOOP:
1813
1814 034410  012704  035000            MOV     #T8PACKET,R4            ;PACKET FOR WRITE CHARACTERISTICS
1815 034414  004737  015764    5$:     JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1816 034420  103405                    BCS     10$                     ;BR IF SOFT INIT = OK
1820 034422  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1821 034424                            ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      034424  104455                                                                  TRAP    C$ERDF
      034426  001441                                                                  .WORD   801
      034430  003646                                                                  .WORD   SFIERR
      034432  012024                                                                  .WORD   SFIMSG
1822 034434  042714  040000    10$:    BIC     #BIT14,(R4)             ;CLEAR THE CVC BIT
1823 034440  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
1824 034444  004737  016326            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1825 034450  103405                    BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1826 034452  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1830 034454                            ERRDF   ERRNO,T8SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      034454  104455                                                                  TRAP    C$ERDF
```

TEST 8: VOLUME CHECK

```
        034456  001442                                                          .WORD   802
        034460  035170                                                          .WORD   T8SSR
        034462  012036                                                          .WORD   PKTSSR
   1831 034464                          15$:    CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
        034464  104406                                                          TRAP    C$CLP1
   1832 034466                                  ESCAPE  TST                 ;EXIT IF FATAL ERROR
        034466  104410                                                          TRAP    C$ESCAPE
        034470  000604                                                          .WORD   L10064 .
   1833 034472  012702  035022                  MOV     #T8BFR,R2           ;ADDRESS OF THE MESSAGE BUFFER
   1834 034476  032762  000020  000006         BIT     #XSOVCK,XST0(R2)    ;IS VOLUME CHECK SET IN XST0 ?
   1835 034504  001406                          BEQ     20$                 ;OKAY IF VOLUME CHECK IS CLEAR
   1839 034506  016501  000002                  MOV     TSSR(R5),R1         ;CONTENTS OF TSSR FOR ERROR REPORT
   1840 034512                                  ERRHRD  ERRNO,T8NVCK,PKTMES ;VOLUME CHECK NOT CLEAR
        034512  104456                                                          TRAP    C$ERHRD
        034514  001443                                                          .WORD   803
        034516  035077                                                          .WORD   T8NVCK
        034520  012100                                                          .WORD   PKTMES
   1841 034522                          20$:    CKLOOP                      ;LOOP ON ERROR ?
        034522  104406                                                          TRAP    C$CLP1
   1842 034524  010465  000000                  MOV     R4,TSDB(R5)         ;SET THE PACKET ADDRESS FOR WRITE CHAR
   1843 034530  004737  016326                  JSR     PC,CHKTSSR          ;WAIT FOR SSR TO SET
   1844 034534  103405                          BCS     25$                 ;BR IF CARRY SET (GOOD RETURN)
   1845 034536  010001                          MOV     R0,R1               ;SAVE CONTENTS OF TSSR
   1849 034540                                  ERRDF   ERRNO,T8SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
        034540  104455                                                          TRAP    C$ERDF
        034542  001444                                                          .WORD   804
        034544  035170                                                          .WORD   T8SSR
        034546  012036                                                          .WORD   PKTSSR
   1850 034550                          25$:    CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
        034550  104406                                                          TRAP    C$CLP1
   1851 034552                                  ESCAPE  TST                 ;EXIT IF FATAL ERROR
        034552  104410                                                          TRAP    C$ESCAPE
        034554  000520                                                          .WORD   L10064-.
   1852 034556  032762  000020  000006         BIT     #XSOVCK,XST0(R2)    ;IS VOLUME CHECK SET IN XST0 ?
   1853 034564  001406                          BEQ     30$                 ;OKAY IF VOLUME CHECK IS SET
   1857 034566  016501  000002                  MOV     TSSR(R5),R1         ;CONTENTS OF TSSR FOR ERROR REPORT
   1858 034572                                  ERRHRD  ERRNO,T8NVCK,PKTMES ;VOLUME CHECK NOT SET
        034572  104456                                                          TRAP    C$ERHRD
        034574  001445                                                          .WORD   805
        034576  035077                                                          .WORD   T8NVCK
        034600  012100                                                          .WORD   PKTMES
   1859 034602                          30$:    CKLOOP                      ;LOOP ON ERROR ?
        034602  104406                                                          TRAP    C$CLP1
   1860 034604  052714  040000                  BIS     #BIT14,(R4)         ;SET THE CVC BIT
   1861 034610  010465  000000                  MOV     R4,TSDB(R5)         ;SET THE PACKET ADDRESS FOR WRITE CHAR
   1862 034614  004737  016326                  JSR     PC,CHKTSSR          ;WAIT FOR SSR TO SET
   1863 034620  103405                          BCS     35$                 ;BR IF CARRY SET (GOOD RETURN)
   1864 034622  010001                          MOV     R0,R1               ;SAVE CONTENTS OF TSSR
   1868 034624                                  ERRDF   ERRNO,T8SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
        034624  104455                                                          TRAP    C$ERDF
        034626  001446                                                          .WORD   806
        034630  035170                                                          .WORD   T8SSR
        034632  012036                                                          .WORD   PKTSSR
   1869 034634                          35$:    CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
        034634  104406                                                          TRAP    C$CLP1
   1870 034636                                  ESCAPE  TST                 ;EXIT IF FATAL ERROR
        034636  104410                                                          TRAP    C$ESCAPE
```

TEST 8: VOLUME CHECK

```
                034640  000434                                                          .WORD   L10064-.
1871            034642  032762  000020  000006          BIT     #XSOVCK,XSTO(R2)        ;IS VOLUME CHECK CLEAR IN XSTO ?
1872            034650  001406                          BEQ     40$                     ;OKAY IF VOLUME CHECK IS CLEARED
1876            034652  016501  000002                  MOV     TSSR(R5),R1             ;CONTENTS OF TSSR FOR ERROR REPORT
1877            034656                                  ERRHRD  ERRNO,T8VCK,PKTMES      ;VOLUME CHECK NOT CLEARED
                034656  104456                                                          TRAP    C$ERHRD
                034660  001447                                                          .WORD   807
                034662  035042                                                          .WORD   T8VCK
                034664  012100                                                          .WORD   PKTMES
1878            034666                          40$:    CKLOOP                          ;LOOP ON ERROR ?
                034666  104406                                                          TRAP    C$CLP1
1879            034670  042714  040000                  BIC     #BIT14,(R4)             ;CLEAR THE CVC BIT
1880            034674  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
1881            034700  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1882            034704  103405                          BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
1883            034706  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1887            034710                                  ERRDF   ERRNO,T8SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                034710  104455                                                          TRAP    C$ERDF
                034712  001450                                                          .WORD   808
                034714  035170                                                          .WORD   T8SSR
                034716  012036                                                          .WORD   PKTSSR
1888            034720                          45$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                034720  104406                                                          TRAP    C$CLP1
1889            034722                                  ESCAPE  TST                     ;EXIT IF FATAL ERROR
                034722  104410                                                          TRAP    C$ESCAPE
                034724  000350                                                          .WORD   L10064-.
1890            034726  032762  000020  000006          BIT     #XSOVCK,XSTO(R2)        ;IS VOLUME CHECK CLEAR IN XSTO ?
1891            034734  001406                          BEQ     50$                     ;OKAY IF VOLUME CHECK IS CLEARED
1895            034736  016501  000002                  MOV     TSSR(R5),R1             ;CONTENTS OF TSSR FOR ERROR REPORT
1896            034742                                  ERRHRD  ERRNO,T8VCK,PKTMES      ;VOLUME CHECK NOT CLEARED
                034742  104456                                                          TRAP    C$ERHRD
                034744  001451                                                          .WORD   809
                034746  035042                                                          .WORD   T8VCK
                034750  012100                                                          .WORD   PKTMES
1897            034752                          50$:    CKLOOP                          ;LOOP ON ERROR ?
                034752  104406                                                          TRAP    C$CLP1
1898            034754  004737  016446          60$:    JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
1899            034760  103002                          BCC     62$                     ;BRANCH IF NOT
1900            034762  000137  034410                  JMP     T8LOOP                  ;LOOP UNTIL COUNT EXPIRED
1901            034766                          62$:    EXIT    TST                     ;ALL DONE THIS TEST
                034766  104432                                                          TRAP    C$EXIT
                034770  000304                                                          .WORD   L10064 .
1902
1903                                            ;+
1904                                            ;LOCAL STORAGE FOR THIS TEST
1905                                            ;-
1906
1908                    035000                          .=<.+10>&177770
1910            035000                          T8PACKET:                               ;COMMAND PACKET FOR TEST
1911            035000  100004                          .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND
1912            035002  035010                          .WORD   T8DATA                  ;ADDRESS OF CHARACTERISTICS BLOCK
1913            035004  000000                          .WORD   0
1914            035006  000010                          .WORD   10                      ;STARTING VALUE OF COUNTER
1915
1916            035010                          T8DATA:                                 ;CHARACTERISTICS DATA BLOCK
1917            035010  035022                          .WORD   T8BFR                   ;ADDRESS OF MESSAGE BUFFER
1918            035012  000000                          .WORD   0
```

SEQ 0169

TEST 8: VOLUME CHECK

```
1919 035014  000020                          .WORD   16.                     ;LENGTH OF MESSAGE BUFFER
1920 035016  000000  000000                  .WORD   0,0
1921
1922 035022                          T8BFR:  .BLKW   8.                      ;MESSAGE BUFFER
1923
1924
1925                                 ;+
1926                                 ;LOCAL TEXT MESSAGES FOR TEST
1927                                 ;-
1928
1929 035042      126      157      154  T8VCK:  .ASCIZ  'Volume Check Bit Not Cleared'
1930 035077      126      157      154  T8NVCK: .ASCIZ  'Volume Check Bit (VCK) Not Clear After Initialize (XSTO)'
1931 035170      103      157      156  T8SSR:  .ASCIZ  'Contents of TSSR Incorrect After Write Characteristics'
1932 035257      126      157      154  TST8ID: .ASCIZ  'Volume Check'
1933                                         .EVEN
1934 035274                              ENDTST
     035274                                                          L10064:
     035274  104401                                                          TRAP    C$ETST
```

```
                                        .SBTTL  TEST  9: COMPLETION INTERRUPT


                                ;       THIS TEST VERIFIES THAT AN INTERRUPT IS GENERATED AT THE
                                ;       COMPLETION OF THE WRITE CHARACTERISTICS COMMAND IF THE INTERRUPT
                                ;       ENABLE (IE) BIT IN THE COMMAND HEADER WORD IS SET.  THIS TEST
                                ;       CHECKS THE FUNCTIONING OF THE INTERRUPT LOGIC AND BASIC
                                ;       PROCESSING OF THE IE BIT.
                                ;
                                ;       THE SEQUENCES OF TEST 7 ARE REPEATED, EXCEPT THAT THE INTERRUPT
                                ;       SERVICE ROUTINE IS SET UP TO EXPECT INTERRUPTS AND EACH WRITE
                                ;       CHARACTERISTICS COMMAND IS ISSUED WITH THE IE BIT SET (1).  IT
                                ;       IS VERIFIED, WHERE APPROPRIATE, THAT THE IE STATUS BIT IN XST0
                                ;       OF ANY MESSAGE PACKET IS SET AND THAT A COMPLETION INTERRUPT IS
                                ;       GENERATED.  FINALLY, A SEQUENCE OF TWO COMMANDS ARE ISSUED, THE
                                ;       FIRST WITH IE=1 AND THE SECOND WITH IE=0.  IT IS VERIFIED THAT
                                ;       NO INTERRUPT IS GENERATED AFTER THE SECOND COMMAND AND THAT THE
                                ;       IE BIT IN XST0 IS 0.
                                ;
1955 035276                             BGNTST
     035276                                                             T9::
1960 035276  005037  002224            CLR     EXTFEA                  ;CLEAR EXTENDED FEATURES SWITCH
1961 035302  012700  040311            MOV     #TST9ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
1962 035306  004737  016500            JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
1963 035312  012737  000024  002214    MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
1964 035320                     T9LOOP:
1965 035320                             BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     035320                                                                T9.1:
     035320  104402                                                                TRAP    C$BSUB
1966
1967 035322  004737  040336            JSR     PC,T9REST               ;SET PACKET TO INITIAL VALUES
1968 035326                             SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     035326  012700  000000                                                         MOV     #PRI00,R0
     035332  104441                                                                TRAP    C$SPRI
1969 035334  012703  002762            MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
1970 035340  012704  037240            MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
1971 035344  012764  000010  000006    MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
1972 035352                     5$:
1973 035352                             BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     035352  104404                                                                TRAP    C$BSEG
1974
1975 035354  004737  015764            JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
1976 035360  103405                    BCS     10$                     ;BR IF SOFT INIT = OK
1980 035362  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1981 035364                             ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     035364  104455                                                                TRAP    C$ERDF
     035366  001605                                                                .WORD   901
     035370  003646                                                                .WORD   SFIERR
     035372  012024                                                                .WORD   SFIMSG
1982 035374  005037  002220    10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
1983 035400  005037  002222            CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
1984 035404  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
1985 035410  004737  016326            JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
1986 035414  103407                    BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
1987 035416  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
1991 035420                             ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     035420  104455                                                                TRAP    C$ERDF
```

TEST 9: COMPLETION INTERRUPT SEQ 0171

```
        035422  001606                                                          .WORD   902
        035424  037757                                                          .WORD   T9SSR
        035426  012036                                                          .WORD   PKTSSR
  1992  035430  005237  002220            INC     FATFLG          ;SET FATAL ERROR FLAG
  1993  035434                    15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
        035434  104406                                                          TRAP    C$CLP1
  1994  035436                            ESCAPE  SEG             ;BY PASS SUBTEST IF FATAL ERROR
        035436  104410                                                          TRAP    C$ESCAPE
        035440  000056                                                          .WORD   10000$ .
  1995  035442  005737  002222            TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
  1996  035446  001004                    BNE     22$             ;BRANCH IF YES
  2000  035450                            ERRHRD  ERRNO,T9NINT,PKTSSR
        035450  104456                                                          TRAP    C$ERHRD
        035452  001607                                                          .WORD   903
        035454  040046                                                          .WORD   T9NINT
        035456  012036                                                          .WORD   PKTSSR
  2001  035460  016501  000002    22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
  2002  035464  012702  000200            MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
  2003  035470  032701  000100            BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
  2004  035474  001402                    BEQ     25$             ;BRANCH IF NOT OFF LINE
  2005  035476  052702  000100            BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
  2006  035502  020201    25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
  2007  035504  001404                    BEQ     30$             ;OKAY IF MATCH
  2011  035506                            ERRHRD  ERRNO,T9NBA,PKTSSR      ;NBA NOT ZERO
        035506  104456                                                          TRAP    C$ERHRD
        035510  001610                                                          .WORD   904
        035512  037316                                                          .WORD   T9NBA
        035514  012036                                                          .WORD   PKTSSR
  2012  035516                    30$:
  2013  035516                            ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        035516                                                          10000$:
        035516  104405                                                          TRAP    C$ESEG
  2014
  2015  035520  012364  000006            MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
  2016  035524  020327  003056            CMP     R3,#TBLEND      ;HAS ALL DATA BEEN TESTED ?
  2017  035530  103002                    BHIS    55$             ;BRANCH IF ALL DATA DONE
  2018  035532  000137  035352            JMP     5$              ;BRANCH TILL BACK TO ZERO
  2019
  2020  035536                    55$:    ENDSUB                  ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
        035536                                                          L10066:
        035536  104403                                                          TRAP    C$ESUB
  2021
  2022  035540  005737  002220            TST     FATFLG          ;ANY FATAL ERRORS ?
  2023  035544  001402                    BEQ     60$             ;BRANCH IF NOT
  2024  035546  004737  017172            JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
  2025  035552  033727  037274  000200    BIT     T9BFR+12,#BIT7  ;EXTENDED FEATURES SET?
  2026  035560  001402                    BEQ     70$             ;BR IF NO
  2027  035562  005237  002224            INC     EXTFEA          ;SET EXT FEATURE FLAG
  2028  035566                    70$:
```

TEST 9: COMPLETION INTERRUPT

```
2030                                    ;*
2031                                    ;
2032                                    ;TEST 9, SUBTEST 2
2033                                    ;
2034                                    ;CHECK THAT UNUSED BITS BEING SET CAUSES
2035                                    ;WRITE CHARACTERISTICS COMMAND TO BE REJECTED
2036                                    ;
2037                                    ;-
2038
2039 035566                                     BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     035566                                                                                     T9.2:
     035566  104402                                                                                 TRAP    C$BSUB
2040
2041 035570                                     SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     035570  012700  000000                                                                         MOV     #PRI00,R0
     035574  104441                                                                                 TRAP    C$SPRI
2042 035576  012703  037302                     MOV     #T92DATA,R3             ;START OF TEST DATA FOR SUBTEST
2043 035602  012704  037240             5$:     MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
2044 035606  004737  040336                     JSR     PC,T9REST               ;RESTORE PACKET TO STARTING VALUES
2045
2046 035612                                     BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     035612  104404                                                                               TRAP    C$BSEG
2047
2048 035614  004737  015764                     JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2049 035620  103405                           BCS     10$                     ;BR IF SOFT INIT = OK
2053 035622  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2054 035624                                     ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     035624  104455                                                                               TRAP    C$ERDF
     035626  001611                                                                               .WORD   905
     035630  003646                                                                               .WORD   SFIERR
     035632  012024                                                                               .WORD   SFIMSG
2055 035634  005037  002222             10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2056 035640  010400                           MOV     R4,R0                   ;START OF THE COMMAND PACKET
2057 035642  061300                           ADD     (R3),R0                 ;OFFSET TO THE DATA WORD TO TEST
2058 035644  056310  000002                   BIS     2(R3),(R0)              ;SET THE DATA BITS TO BE TESTED
2059 035650  010465  000000                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2060 035654  004737  016240                   JSR     PC,WAITF                ;WAIT FOR SSR TO SET
2061 035660  103405                           BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2062 035662  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2066 035664                                     ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     035664  104455                                                                               TRAP    C$ERDF
     035666  001612                                                                               .WORD   906
     035670  037757                                                                               .WORD   T9SSR
     035672  012036                                                                               .WORD   PKTSSR
2067 035674                             15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     035674  104406                                                                               TRAP    C$CLP1
2068 035676                                     ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
     035676  104410                                                                               TRAP    C$ESCAPE
     035700  000056                                                                               .WORD   10000$-.
2069 035702  005737  002222                     TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2070 035706  001004                           BNE     22$                     ;BRANCH IF YES
2074 035710                                     ERRHRD  ERRNO,T9NINT,PKTSSR
     035710  104456                                                                               TRAP    C$ERHRD
     035712  001613                                                                               .WORD   907
     035714  040046                                                                               .WORD   T9NINT
     035716  012036                                                                               .WORD   PKTSSR
2075 035720  016501  000002             22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
```

SEQ 0173

TEST 9: COMPLETION INTERRUPT

```
2076 035724  012702  102206                MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
2077 035730  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
2078 035734  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
2079 035736  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2080 035742  020201              25$:      CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2081 035744  001404                        BEQ     30$                     ;OKAY IF MATCH
2085 035746                                ERRHRD  ERRNO,T92REJ,PKTSSR     ;COMMAND NOT REJECTED
     035746  104456                                                        TRAP    C$ERHRD
     035750  000614                                                        .WORD   908
     035752  037371                                                        .WORD   T92REJ
     035754  012036                                                        .WORD   PKTSSR
2086 035756                              30$:
2087 035756                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     035756                                                                10000$:
     035756  104405                                                        TRAP    C$ESEG
2088
2089 035760  062703  000004                ADD     #4,R3                   ;POINT TO NEXT DATA PAIR
2090 035764  020327  037316                CMP     R3,#T92DONE             ;COMPARE TO END OF TEST DATA
2091 035770  103002                        BHIS    59$                     ;BRANCH IF ALL DATA TESTED
2092 035772  000137  035602                JMP     5$                      ;BRANCH TILL BACK TO ZERO
2093
2094 035776                              59$:      ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     035776                                                                L10067:
     035776  104403                                                        TRAP    C$ESUB
```

TEST 9: COMPLETION INTERRUPT

```
2096                                    ;*
2097                                    ;
2098                                    ;TEST 9, SUBTEST 3
2099                                    ;
2100                                    ;CHECK THE WRITE CHARACTERISTICS COMMAND IS REJECTED
2101                                    ;IF ISSUED WITH AN INVALID DATA BLOCK BYTE COUNT
2102                                    ;
2103                                    ;-
2104
2105 036000                                     BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
      036000                                                                                      T9.3:
      036000  104402                                                                             TRAP    C$BSUB
2106
2107 036002                                     SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS.
      036002  012700  000000                                                                     MOV     #PRI00,R0
      036006  104441                                                                             TRAP    C$SPRI
2108 036010  012703  000001                     MOV     #1,R3                   ;STARTING BYTE COUNT
2109 036014  012704  037240           5$:       MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
2110 036020  004737  040336                     JSR     PC,T9REST               ;RESTORE PACKET TO STARTING VALUES
2111
2112 036024                                     BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      036024  104404                                                                             TRAP    C$BSEG
2113
2114 036026  004737  015764                     JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2115 036032  103405                             BCS     10$                     ;BR IF SOFT INIT = OK
2119 036034  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2120 036036                                     ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      036036  104455                                                                             TRAP    C$ERDF
      036040  001615                                                                             .WORD   909
      036042  003646                                                                             .WORD   SFIERR
      036044  012024                                                                             .WORD   SFIMSG
2121 036046  005037  002222           10$:       CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2122 036052  010364  000006                     MOV     R3,PKBCNT(R4)           ;INSERT THE BYTE COUNT FOR TEST
2123 036056  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2124 036062  004737  016240                     JSR     PC,WAITF                ;WAIT FOR SSR TO SET
2125 036066  103405                             BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2126 036070  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2130 036072                                     ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      036072  104455                                                                             .                       TRAP    C$ERDF
      036074  001616                                                                             .WORD   910
      036076  037757                                                                             .WORD   T9SSR
      036100  012036                                                                             .WORD   PKTSSR
2131 036102                             15$:       CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      036102  104406                                                                             TRAP    C$CLP1
2132 036104                                     FSCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
      036104  104410                                                                             TRAP    C$ESCAPE
      036106  000056                                                                             .WORD   10000$ .
2133 036110  005737  002222                     TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2134 036114  001004                             BNE     22$                     ;BRANCH IF YES
2138 036116                                     ERRHRD  ERRNO,T9NINT,PKTSSR
      036116  104456                                                                             TRAP    C$ERHRD
      036120  001617                                                                             .WORD   911
      036122  040046                                                                             .WORD   T9NINT
      036124  012036                                                                             .WORD   PKTSSR
2139 036126  016501  000002           22$:       MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
2140 036132  012702  102206                     MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
2141 036136  032701  000100                     BIT     #OFL,R1                 ;IS OFF LINE BIT SET ?
```

TEST 9: COMPLETION INTERRUPT

```
2142 036142  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
2143 036144  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2144 036150  020201                 25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2145 036152  001404                        BEQ     30$                     ;OKAY IF MATCH
2149 036154                                 ERRHRD  ERRNO,T93REJ,PKTSSR     ;COMMAND NOT REJECTED
     036154  104456                                                                TRAP    C$ERHRD
     036156  001620                                                                .WORD   912
     036160  037470                                                                .WORD   T93REJ
     036162  012036                                                                .WORD   PKTSSR
2150 036164                         30$:
2151 036164                                 ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     036164                                                                     10000$:
     036164  104405                                                                TRAP    C$ESEG
2152
2153 036166  005203                        INC     R3                      ;NEXT BYTE COUNT
2154 036170  020327  000006                CMP     R3,#6                   ;TESTED ALL INVALID ?
2155 036174  002002                        BGE     59$                     ;BRANCH IF TEST DONE
2156 036176  000137  036014                JMP     5$                      ;BRANCH TILL BACK TO ZERO
2157
2158 036202                         59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     036202                                                                     L10070:
     036202  104403                                                                TRAP    C$ESUB
```

TEST 9: COMPLETION INTERRUPT

```
2160                                    ;*
2161                                    ;
2162                                    ;TEST 9, SUBTEST 4
2163                                    ;
2164                                    ;SUBTEST TO VERIFY THAT A WRITE CHARACTERISTICS COMMAND IS
2165                                    ;REJECTED IF AN ILLEGAL DATA BLOCK ADDRESS IS ISSUED.
2166                                    ;
2167                                    ;-
2168
2169 036204                                     BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
      036204                                                                                      T9.4:
      036204  104402                                                                                    TRAP    C$BSUB
2170
2171 036206                                     SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      036206  012700  000000                                                                             MOV     #PRI00,R0
      036212  104441                                                                                     TRAP    C$SPRI
2172 036214  012703  037302                     MOV     #T92DATA,R3             ;START OF TEST DATA FOR SUBTEST
2173 036220  012704  037240               5$:   MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
2174 036224  004737  040336                     JSR     PC,T9REST               ;RESTORE PACKET TO STARTING VALUES
2175
2176
2177 036230  004737  015764                     JS      PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2178 036234  103405                             BCS     10$                     ;BR IF SOFT INIT = OK
2182 036236  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2183 036240                                     ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      036240  104455                                                                                     TRAP    C$ERDF
      036242  001621                                                                                     .WORD   913
      036244  003646                                                                                     .WORD   SFIERR
      036246  012024                                                                                     .WORD   SFIMSG
2184 036250  005037  002222               10$:   CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2185 036254  052737  000001  037250             BIS     #1,T9DATA               ;MAKE ADDRESS ODD
2186 036262  010465  000000                     MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2187 036266  004737  016240                     JSR     PC,WAITF                ;WAIT FOR SSR TO SET
2188 036272  103405                             BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2189 036274  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2193 036276                                     ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      036276  104455                                                                                     TRAP    C$ERDF
      036300  001622                                                                                     .WORD   914
      036302  037757                                                                                     .WORD   T9SSR
      036304  012036                                                                                     .WORD   PKTSSR
2194 036306                             15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      036306  104406                                                                                     TRAP    C$CLP1
2195 036310                                     ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
      036310  104410                                                                                     TRAP    C$ESCAPE
      036312  000056                                                                                     .WORD   L10071 .
2196 036314  005737  002222                     TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2197 036320  001004                             BNE     22$                     ;BRANCH IF YES
2201 036322                                     ERRHRD  ERRNO,T9NINT,PKTSSR
      036322  104456                                                                                     TRAP    C$ERHRD
      036324  001623                                                                                     .WORD   915
      036326  040046                                                                                     .WORD   T9NINT
      036330  012036                                                                                     .WORD   PKTSSR
2202 036332  016501  000002               22$:   MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
2203 036336  012702  102206                     MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
2204 036342  032701  000100                     BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
2205 036346  001402                             BEQ     25$                     ;BRANCH IF NOT OFF-LINE
2206 036350  052702  000100                     BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
```

TEST 9: COMPLETION INTERRUPT

SEQ 0177

```
 2207 036354  020201          25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
 2208 036356  001404                  BEQ     30$             ;OKAY IF MATCH
 2212 036360                          ERRHRD  ERRNO,T94REJ,PKTSSR     ;COMMAND NOT REJECTED
      036360  104456                                                  TRAP    C$ERHRD
      036362  001624                                                  .WORD   916
      036364  037563                                                  .WORD   T94REJ
      036366  012036                                                  .WORD   PKTSSR
 2213 036370                  30$:
 2214
 2215 036370                          ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      036370                                                  L10071:
      036370  104403                                                  TRAP    C$ESUB
 2216
```

TEST 9: COMPLETION INTERRUPT

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

2227 036372                             BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     036372                                                                         T9.5:
     036372  104402                                                                TRAP    C$BSUB
2228
2229 036374                             SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     036374  012700  000000                                                         MOV     #PRI00,R0
     036400  104441                                                                 TRAP    C$SPRI
2230 036402  012703  000001            MOV     #1,R3                   ;STARTING BUFFER LENGTH
2231 036406  012704  037240    5$:     MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
2232 036412  004737  040336            JSR     PC,T9REST               ;RESTORE PACKET TO STARTING VALUES
2233
2234 036416                             BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     036416  104404                                                                 TRAP    C$BSEG
2235
2236 036420  004737  015764            JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2237 036424  103405                    BCS     10$                     ;BR IF SOFT INIT = OK
2241 036426  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2242 036430                             ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     036430  104455                                                                 TRAP    C$ERDF
     036432  001625                                                                 .WORD   917
     036434  003646                                                                 .WORD   SFIERR
     036436  012024                                                                 .WORD   SFIMSG
2243 036440  005037  002222    10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2244 036444  010337  037254            MOV     R3,T9DATA+4             ;INSERT THE BAD MESSAGE LENGTH
2245 036450  010465  000000            MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2246 036454  004737  016240            JSR     PC,WAITF                ;WAIT FOR SSR TO SET
2247 036460  103405                    BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2248 036462  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2252 036464                             ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     036464  104455                                                                 TRAP    C$ERDF
     036466  001626                                                                 .WORD   918
     036470  037757                                                                 .WORD   T9SSR
     036472  012036                                                                 .WORD   PKTSSR
2253 036474                    15$:     CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     036474  104406                                                                 TRAP    C$CLP1
2254 036476                             ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     036476  104410                                                                 TRAP    C$ESCAPE
     036500  000056                                                                 .WORD   10000$ .
2255 036502  005737  002222            TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2256 036506  001004                    BNE     22$                     ;BRANCH IF YES
2260 036510                             ERRHRD  ERRNO,T9NINT,PKTSSR
     036510  104456                                                                 TRAP    C$ERHRD
     036512  001627                                                                 .WORD   919
     036514  040046                                                                 .WORD   T9NINT
     036516  012036                                                                 .WORD   PKTSSR
2261 036520  016501  000002    22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
2262 036524  012702  102206            MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
2263 036530  032701  000100            BIT     #OFL,R1                 ;IS OFF LINE BIT SET ?
```

TEST 9: COMPLETION INTERRUPT

```
2264 036534  001402                    BEQ     25$                     ;BRANCH IF NOT OFF-LINE
2265 036536  052702  000100            BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2266 036542  020201              25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2267 036544  001404                    BEQ     30$                     ;OKAY IF MATCH
2271 036546                            ERRHRD  ERRNO,T95REJ,PKTSSR     ;COMMAND NOT REJECTED
     036546  104456                                                            TRAP    C$ERHRD
     036550  001630                                                            .WORD   920
     036552  037661                                                            .WORD   T95REJ
     036554  012036                                                            .WORD   PKTSSR
2272 036556                      30$:
2273 036556                            ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     036556                                                                  10000$:
     036556  104405                                                            TRAP    C$ESEG
2274
2275 036560  005203                    INC     R3                      ;NEXT BUFFER LENGTH
2276 036562  020327  000014            CMP     R3,#14                  ;HAVE ALL BAD VALUES BEEN TESTED ?
2277 036566  002002                    BGE     59$                     ;BRANCH IF ALL TESTED
2278 036570  000137  036406            JMP     5$                      ;BRANCH TILL BACK TO ZERO
2279
2280 036574                      59$:  ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     036574                                                                  L10072:
     036574  104403                                                            TRAP    C$ESUB
```

TEST 9: COMPLETION INTERRUPT

```
2282                                    ;*
2283                                    ;
2284                                    ;TEST 9, SUBTEST 6
2285                                    ;
2286                                    ;THIS SUBTEST IS EXECUTED ONLY IF THE EXTENDED
2287                                    ;FEATURES MODE IS ENABLED (AS DETERMINED BY EXAMINING
2288                                    ;XST2 AFTER A PREVIOUS EXECUTION OF WRITE CHARACTERISTICS).
2289                                    ;IT VERIFIES THAT A FIFTH CHARACTERISTICS DATA WORD IS FETCHED
2290                                    ;IF THE BYTE COUNT PARAMETER IN THE COMMAND PACKET IS 10 DECIMAL
2291                                    ;OR GREATER.
2292                                    ;
2293                                    ;-
2294 036576                                BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
      036576                                                                        T9.6:
      036576  104402                                                                   TRAP    C$BSUB
2295 036600  005737  002224                TST     EXTFEA                  ;IS EXTENDED FEATURES SOFT. SW SET?
2296 036604  001002                        BNE     4$                      ;BR, IF SOFTWARE SWITCH IS SET (ON)
2297 036606  000137  037026                JMP     55$                     ;EXIT SUBTEST
2298 036612  004737  040336          4$:   JSR     PC,T9REST               ;SET PACKET TO START-UP VALUES
2299
2300 036616                                SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      036616  012700  000000                                                        MOV     #PRI00,R0
      036622  104441                                                               TRAP    C$SPRI
2301 036624  012703  002762                MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
2302 036630  012704  037240                MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
2303 036634  012764  000012  000006        MOV     #10.,PKBCNT(R4)         ;START WITH EXTENDED FEATURES VALUE
2304 036642                          5$:
2305 036642                                BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      036642  104404                                                               TRAP    C$BSEG
2306
2307 036644  004737  015764                JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2308 036650  103405                        BCS     10$                     ;BR IF SOFT INIT = OK
2312 036652  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2313 036654                                ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      036654  104455                                                               TRAP    C$ERDF
      036656  001631                                                               .WORD   921
      036660  003646                                                               .WORD   SFIERR
      036662  012024                                                               .WORD   SFIMSG
2314 036664  005037  002220          10$:  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
2315 036670  005037  002222                CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2316 036674  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2317 036700  004737  016326                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2318 036704  103407                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2319 036706  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2323 036710                                ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      036710  104455                                                               TRAP    C$ERDF
      036712  001632                                                               .WORD   922
      036714  037757                                                               .WORD   T9SSR
      036716  012036                                                               .WORD   PKTSSR
2324 036720  005237  002220                INC     FATFLG                  ;SET FATAL ERROR FLAG
2325 036724                          15$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      036724  104406                                                               TRAP    C$CLP1
2326 036726                                ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
      036726  104410                                                               TRAP    C$ESCAPE
      036730  000056                                                               .WORD   10000$-.
2327 036732  005737  002222                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2328 036736  001004                        BNE     22$                     ;BRANCH IF YES
```

TEST 9: COMPLETION INTERRUPT

```
2332 036740                                ERRHRD  ERRNO,T9NINT,PKTSSR
     036740  104456                                                    TRAP    C$ERHRD
     036742  001633                                                    .WORD   923
     036744  040046                                                    .WORD   T9NINT
     036746  012036                                                    .WORD   PKTSSR
2333 036750  016501  000002        22$:    MOV     TSSR(R5),R1         ;GET THE CONTENTS OF TSSR
2334 036754  012702  000200              MOV     #SSR,R2             ;EXPECTED CONTENTS OF TSSR
2335 036760  032701  000100              BIT     #OFL,R1             ;IS OFF-LINE BIT SET ?
2336 036764  001402                      BEQ     25$                 ;BRANCH IF NOT OFF-LINE
2337 036766  052702  000100              BIS     #OFL,R2             ;SET OFF-LINE IN EXPECTED DATA
2338 036772  020201              25$:    CMP     R2,R1               ;DOES EXPECTED MATCH RECEIVED ?
2339 036774  001404                      BEQ     30$                 ;OKAY IF MATCH
2343 036776                              ERRHRD  ERRNO,T9NBA,PKTSSR  ;NBA NOT ZERO
     036776  104456                                                    TRAP    C$ERHRD
     037000  001634                                                    .WORD   924
     037002  037316                                                    .WORD   T9NBA
     037004  012036                                                    .WORD   PKTSSR
2344 037006                        30$:
2345 037006                              ENDSEG                      ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     037006                                                                10000$:
     037006  104405                                                    TRAP    C$ESEG
2346
2347 037010  012364  000006              MOV     (R3)+,PKBCNT(R4)    ;SET THE TEST WORD
2348 037014  020327  003056              CMP     R3,#TBLEND          ;HAS ALL DATA BEEN TESTED ?
2349 037020  103002                      BHIS    55$                 ;BRANCH IF ALL DATA DONE
2350 037022  000137  036642              JMP     5$                  ;BRANCH TILL BACK TO ZERO
2351
2352 037026                        55$:  ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     037026                                                                L10073:
     037026  104403                                                    TRAP    C$ESUB
```

TEST 9: COMPLETION INTERRUPT

```
                                        ;+
                                        ;
                                        ;TEST 9, SUBTEST 7
                                        ;
                                        ;TEST WRITE CHARACTERISTICS WITH/WITHOUT INTERRUPTS ENABLED
                                        ;
                                        ;-

037030                                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
037030                                                                                          T9.7:
037030  104402                                                                                          TRAP    C$BSUB

037032                                          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
037032  012700  000000                                                                                  MOV     #PRI00,R0
037036  104441                                                                                          TRAP    C$SPRI
037040  012704  037240                         MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
037044  004737  040336                         JSR     PC,T9REST               ;SET UP A VALID PACKET
037050  004737  015764                         JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
037054  103405                                 BCS     10$                     ;BR IF SOFT INIT = OK
037056  010001                                 MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
037060                                         ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
037060  104455                                                                                          TRAP    C$ERDF
037062  001635                                                                                          .WORD   925
037064  003646                                                                                          .WORD   SFIERR
037066  012024                                                                                          .WORD   SFIMSG
037070  005037  002222                 10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
037074  052714  000200                         BIS     #BIT7,(R4)              ;ENABLE INTERRUPTS
037100  010465  000000                         MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
037104  004737  016326                         JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
037110  103405                                 BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
037112  010001                                 MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
037114                                         ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
037114  104455                                                                                          TRAP    C$ERDF
037116  001636                                                                                          .WORD   926
037120  037757                                                                                          .WORD   T9SSR
037122  012036                                                                                          .WORD   PKTSSR
037124                          15$:    CKLOOP                                  ;LOOP ON ERROR, IF FLAG SET
037124  104406                                                                                          TRAP    C$CLP1
037126                                         ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
037126  104410                                                                                          TRAP    C$ESCAPE
037130  000102                                                                                          .WORD   L10074-.
037132  005737  002222                         TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
037136  001004                                 BNE     22$                     ;BRANCH IF YES
037140                                         ERRHRD  ERRNO,T9NINT,PKTSSR
037140  104456                                                                                          TRAP    C$ERHRD
037142  001637                                                                                          .WORD   927
037144  040046                                                                                          .WORD   T9NINT
037146  012036                                                                                          .WORD   PKTSSR
037150                          22$:    CKLOOP                                  ;LOOP ON ERROR ?
037150  104406                                                                                          TRAP    C$CLP1

037152  005037  002222                         CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
037156  042714  000200                         BIC     #BIT7,(R4)              ;DISABLE INTERRUPTS
037162  010465  000000                         MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
037166  004737  016326                         JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
037172  103405                                 BCS     25$                     ;BR IF CARRY SET (GOOD RETURN)
```

TEST 9: COMPLETION INTERRUPT

```
2400 037174  010001                    MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2404 037176                            ERRDF   ERRNO,T9SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     037176  104455                                                            TRAP    C$ERDF
     037200  001640                                                            .WORD   928
     037202  037757                                                            .WORD   T9SSR
     037204  012036                                                            .WORD   PKTSSR
2405 037206                    25$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     037206  104406                                                            TRAP    C$CLP1
2406 037210                            ESCAPE  SUB                     ;BY PASS SUBTEST IF FATAL ERROR
     037210  104410                                                            TRAP    C$ESCAPE
     037212  000020                                                            .WORD   L10074 .
2407 037214  005737  002222            TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2408 037220  001404                    BEQ     30$                     ;BRANCH IF NOT
2412 037222                            ERRHRD  ERRNO,T9INT,PKTSSR
     037222  104456                                                            TRAP    C$ERHRD
     037224  001641                                                            .WORD   929
     037226  040137                                                            .WORD   T9INT
     037230  012036                                                            .WORD   PKTSSR
2413 037232                    30$:
2414 037232                            ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
     037232                                                                  L10074:
     037232  104403                                                            TRAP    C$ESUB
2415
2416 037234                            EXIT    TST                     ;ALL DONE THIS TEST
     037234  104432                                                            TRAP    C$EXIT
     037236  001152                                                            .WORD   L10065-.
2417
2418                                   ;*
2419                                   ;LOCAL STORAGE FOR THIS TEST
2420                                   ;
2421
2425 037240                    T9PACKET:                               ;COMMAND PACKET FOR TEST
2426 037240  100204                    .WORD   100204                  ;WRITE CHAR COMMAND, WITH IE, ACK
2427 037242  037250                    .WORD   T9DATA                  ;ADDRESS OF CHARACTERISTICS BLOCK
2428 037244  000000                    .WORD   0
2429 037246  000010                    .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
2430
2431 037250                    T9DATA:                                 ;CHARACTERISTICS DATA BLOCK
2432 037250  037262                    .WORD   T9BFR                   ;ADDRESS OF MESSAGE BUFFER
2433 037252  000000                    .WORD   0
2434 037254  000016                    .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
2435 037256  000000  000000            .WORD   0,0
2436
2437 037262                    T9BFR:  .BLKW   8.                      ;MESSAGE BUFFER
2438
2439                                   ;*
2440                                   ;
2441                                   ;TEST DATA FOR SUBTEST TWO
2442                                   ;
2443                                   ;DATA HAS FORMAT:
2444                                   ;
2445                                   ;       1ST WORD        OFFSET TO TEST WORD IN PACKET
2446                                   ;       2ND WORD        BITS TO SET FOR TEST
2447                                   ;
2448                                   ;
2449
2450 037302                    T92DATA:
```

TEST 9: COMPLETION INTERRUPT

```
2451 037302  000000  037140                  .WORD   0,BIT5!BIT6!BIT9!BIT10!BIT11!BIT12!BIT13
2452 037306  000002  000001                  .WORD   2,BIT0
2453 037312  000004  100100                  .WORD   4,BIT6!BIT15
2454         037316                  T92DONE=.
2455
2456
2457                                 ;+
2458                                 ;LOCAL TEXT MESSAGES FOR TEST
2459                                 ;-
2460
2461 037316     127     122     111  T9NBA:  .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
2462 037371     127     122     111  T92REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Non-Zero Unused Fields'
2463 037470     127     122     111  T93REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Data Count'
2464 037563     127     122     111  T94REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Block Address'
2465 037661     127     122     111  T95REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Buffer Length'
2466 037757     103     157     156  T9SSR:  .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
2467 040046     105     170     160  T9NINT: .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
2468 040137     125     156     145  T9INT:  .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
2469 040226     111     156     143  T9TSBA: .ASCIZ  'Incorrect TSBA Address After WRITE CHARACTERISTICS'
2470 040311     103     157     155  TST9ID: .ASCIZ  'Completion Interrupt'
2471                                 .EVEN
2472
```

TEST 9: COMPLETION INTERRUPT

```
2474
2475                                    ;+
2476                                    ;
2477                                    ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
2478                                    ;
2479                                    ;-
2480
2481  040336                            T9REST:
2482  040336                                    SAVREG                          ;SAVE THE REGISTERS
2483  040342  012701  037240                   MOV     #T9PACKET,R1            ;START OF THE PACKET
2484  040346  012721  100204                   MOV     #100204,(R1)+           ;WRITE CHARACTERISTICS WITH ACK, IE
2485  040352  012721  037250                   MOV     #T9DATA,(R1)+           ;ADDRESS OF CHAR DATA BLOCK
2486  040356  005021                            CLR     (R1)+                   ;EXTENDED ADDRESS
2487  040360  012721  000010                   MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
2488  040364  012721  037262                   MOV     #T98FR,(R1)+            ;ADDRESS OF MESSAGE BUFFER
2489  040370  005021                            CLR     (R1)+
2490  040372  012721  000016                   MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
2491  040376  005021                            CLR     (R1)+
2492  040400  005011                            CLR     (R1)
2493  040402  005037  037262                   CLR     T98FR                   ;CLEAR 1ST LOC IN MESSAGE BUFFER
2494  040406  000207                            RTS     PC                      ;RETURN
2495  040410                                    ENDTST
      040410                                                                            L10065:
      040410  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
                                ;
                                ;-
                                        BGNTST
                                                                                        T10::
2517 040412  012700  043337             MOV     #TST10ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
2518 040416  004737  016500             JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
2519 040422  012737  000024  002214     MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
2520 040430                     T10LOOP:
2521
2522 040430                             BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
     040430                                                                     T10.1:
     040430  104402                                                                TRAP    C$BSUB
2523
2524 040432  004737  043366             JSR     PC,T10RST               ;SET PACKET TO INITIAL VALUES
2525 040436                             SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     040436  012700  000000                                                         MOV     #PRI00,R0
     040442  104441                                                                 TRAP    C$SPRI
2526 040444  012704  042530             MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
2527 040450  012764  000010  000006     MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
2528 040456                     5$:
2529 040456                             BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     040456  104404                                                                 TRAP    C$BSEG
2530
2531 040460  004737  015764             JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2532 040464  103405                     BCS     10$                     ;BR IF SOFT INIT = OK
2536 040466  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2537 040470                             ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     040470  104455                                                                 TRAP    C$ERDF
     040472  001751                                                                 .WORD   1001
     040474  003646                                                                 .WORD   SFIERR
     040476  012024                                                                 .WORD   SFIMSG
2538 040500  005037  002220     10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
2539 040504  005037  002222             CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2540 040510  010465  000000             MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2541 040514  004737  016326             JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2542 040520  103407                     BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2543 040522  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2547 040524                             ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     040524  104455                                                                 TRAP    C$ERDF
     040526  001752                                                                 .WORD   1002
     040530  043070                                                                 .WORD   T10SSR
     040532  012036                                                                 .WORD   PKTSSR
2548 040534  005237  002220             INC     FATFLG                  ;SET FATAL ERROR FLAG
2549 040540                     15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
```

TEST 10: BASIC PACKET PROTOCOL

```
                040540  104406                                                          TRAP    C$CLP1
          2550  040542                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
                040542  104410                                                          TRAP    C$ESCAPE
                040544  000056                                                          .WORD   10000$ .
          2551  040546  005737  002222          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
          2552  040552  001004                  BNE     22$                     ;BRANCH IF YES
          2556  040554                          ERRHRD  ERRNO,T10NINT,PKTSSR
                040554  104456                                                          TRAP    C$ERHRD
                040556  001753                                                          .WORD   1003
                040560  043157                                                          .WORD   T10NINT
                040562  012036                                                          .WORD   PKTSSR
          2557  040564  016501  000002    22$:  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
          2558  040570  012702  000200          MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
          2559  040574  032701  000100          BIT     #OFL,R1                 ;IS OFF LINE BIT SET ?
          2560  040600  001402                  BEQ     25$                     ;BRANCH IF NOT OFF-LINE
          2561  040602  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
          2562  040606  020201          25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
          2563  040610  001404                  BEQ     30$                     ;OKAY IF MATCH
          2567  040612                          ERRHRD  ERRNO,T10NBA,PKTSSR     ;NBA NOT ZERO
                040612  104456                                                          TRAP    C$ERHRD
                040614  001754                                                          .WORD   1004
                040616  042731                                                          .WORD   T10NBA
                040620  012036                                                          .WORD   PKTSSR
          2568  040622                    30$:
          2569  040622                          ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                040622                                                                10000$:
                040622  104405                                                          TRAP    C$ESEG
          2570  040624                          BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
                040624  104404                                                          TRAP    C$BSEG
          2571
          2572  040626  005037  002222          CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
          2573  040632  012737  025252  042552  MOV     #025252,T10BFR          ;WIPE OUT MESSAGE BUFFER AREA
          2574  040640  012714  100212          MOV     #100212,(R4)            ;SET COMMAND PACKET TO MESS BUF REL
          2575  040644  010465  000000          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
          2576  040650  004737  016326          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
          2577  040654  103407                  BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
          2578  040656  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
          2582  040660                          ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                040660  104455                                                          TRAP    C$ERDF
                040662  001755                                                          .WORD   1005
                040664  043070                                                          .WORD   T10SSR
                040666  012036                                                          .WORD   PKTSSR
          2583  040670  005237  002220          INC     FATFLG                  ;SET FATAL ERROR FLAG
          2584  040674                    45$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                040674  104406                                                          TRAP    C$CLP1
          2585  040676  005737  002222          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
          2586  040702  001404                  BEQ     52$                     ;BRANCH IF NO
          2590  040704                          ERRHRD  ERRNO,T10INT,PKTSSR
                040704  104456                                                          TRAP    C$ERHRD
                040706  001756                                                          .WORD   1006
                040710  043250                                                          .WORD   T10INT
                040712  012036                                                          .WORD   PKTSSR
          2591  040714  016501  000002    52$:  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
          2592  040720  012702  000200          MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
          2593  040724  032701  000100          BIT     #OFL,R1                 ;IS OFF LINE BIT SET ?
          2594  040730  001402                  BEQ     55$                     ;BRANCH IF NOT OFF LINE
          2595  040732  052702  000100          BIS     #OFL,R2                 ;SET OFF LINE IN EXPECTED DATA
```

TEST 10: BASIC PACKET PROTOCOL

```
 2596 040736  020201                55$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 2597 040740  001404                        BEQ     60$                     ;OKAY IF MATCH
 2601 040742                                ERRHRD  ERRNO,T10NNBA,PKTSSR    ;NBA  SET
      040742  104456                                                                TRAP    C$ERHRD
      040744  001757                                                                .WORD   1007
      040746  043013                                                                .WORD   T10NNBA
      040750  012036                                                                .WORD   PKTSSR
 2602 040752                        60$:
 2603 040752  013701  042552                MOV     T10BFR,R1               ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
 2604 040756  012702  025252                MOV     #025252,R2              ;SET UP EXPECTED DATA
 2605 040762  020102                        CMP     R1,R2                   ;WAS ANY MESSAGE REC'D
 2606 040764  001404                        BEQ     70$                     ;BR, IF OK (EQUAL)
 2610 040766                                ERRHRD  ERRNO,T10MBF,EXPREC     ;MESSAGE BUFFER WAS MODIFIED
      040766  104456                                                                TRAP    C$ERHRD
      040770  001760                                                                .WORD   1008
      040772  042634                                                                .WORD   T10MBF
      040774  015464                                                                .WORD   EXPREC
 2611 040776                        70$:
 2612 040776  005737  002220                TST     FATFLG                  ;ANY FATAL ERRORS
 2613 041002  001403                        BEQ     80$                     ;BR, IF NO FATAL ERRORS
 2614 041004  004737  017172                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 2615 041010                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      041010                                                                  10001$:
      041010  104405                                                                TRAP    C$ESEG
 2616 041012                        80$:
 2617 041012                                ENDSUB                          ;////////////// END SUBTEST //////////////
      041012                                                                  L10076:
      041012  104403                                                                TRAP    C$ESUB
 2618
```

```
                                        ;*
                                        ;TEST 10 SUBTEST 2
                                        ;
                                        ;CHECKS THAT THE MESSAGE BUFFER RELEASE COMMAND WORKS
                                        ;PROPERLY AND THAT THERE IS AN INTERRUPT IF THE "IE"
                                        ;BIT IS SET IN THE COMMAND PACKET AND THE "ERI" BIT
                                        ;IS SET IN THE CHARACTERISTICS DATA PACKET
                                        ;
                                        ;-
041014                                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
041014                                                                                  T10.2:
041014  104402                                                                          TRAP    C$BSUB

041016  004737  043366                          JSR     PC,T10RST               ;SET PACKET TO INITIAL VALUES
041022                                          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
041022  012700  000000                                                                  MOV     #PRI00 R0
041026  104441                                                                          TRAP    C$SPRI
041030  012704  042530                          MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
041034  012764  000010  000006                  MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
041042                                  5$:
041042                                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
041042  104404                                                                          TRAP    C$BSEG

041044  004737  015764                          JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
041050  103405                                  BCS     10$                     ;BR IF SOFT INIT = OK
041052  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
041054                                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
041054  104455                                                                          TRAP    C$ERDF
041056  001761                                                                          .WORD   1009
041060  003646                                                                          .WORD   SFIERR
041062  012024                                                                          .WORD   SFIMSG
041064  005037  002220          10$:            CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
041070  005037  002222                          CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
041074  012737  000020  042546                  MOV     #000020,T10DATA+6       ;SET ERI IN CHARACTERISTICS DATA
041102  010465  000000                          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
041106  004737  016326                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
041112  103407                                  BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
041114  010001                                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
041116                                          ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
041116  104455                                                                          TRAP    C$ERDF
041120  001762                                                                          .WORD   1010
041122  043070                                                                          .WORD   T10SSR
041124  012036                                                                          .WORD   PKTSSR
041126  005237  002220                          INC     FATFLG                  ;SET FATAL ERROR FLAG
041132                                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
041132  104406                                                                          TRAP    C$CLP1
041134                                          ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
041134  104410                                                                          TRAP    C$ESCAPE
041136  000056                                                                          .WORD   10000$ .
041140  005737  002222                          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
041144  001004                                  BNE     22$                     ;BRANCH IF YES
041146                                          ERRHRD  ERRNO,T10NINT,PKTSSR
041146  104456                                                                          TRAP    C$ERHRD
041150  001763                                                                          .WORD   1011
041152  043157                                                                          .WORD   T10NINT
041154  012036                                                                          .WORD   PKTSSR
041156  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
```

TEST 10: BASIC PACKET PROTOCOL

```
2666 041162  012702  000200              MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
2667 041166  032701  000100              BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
2668 041172  001402                      BEQ     25$                     ;BRANCH IF NOT OFF-LINE
2669 041174  052702  000100              BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2670 041200  020201                25$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2671 041202  001404                      BEQ     30$                     ;OKAY IF MATCH
2675 041204                              ERRHRD  ERRNO,T10NBA,PKTSSR     ;NBA NOT ZERO
     041204  104456                                                              TRAP    C$ERHRD
     041206  001764                                                              .WORD   1012
     041210  042731                                                              .WORD   T10NBA
     041212  012036                                                              .WORD   PKTSSR
2676 041214                        30$:
2677 041214                              ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     041214                                                                    10000$:
     041214  104405                                                              TRAP    C$ESEG
2678 041216                              BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     041216  104404                                                              TRAP    C$BSEG
2679
2680 041220  005037  002222              CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2681 041224  012737  025252  042552      MOV     #025252,T10BFR          ;WIPE OUT MESSAGE BUFFER AREA
2682 041232  012714  100212              MOV     #100212,(R4)            ;SET COMMAND PACKET TO MESS BUF REL
2683 041236  010465  000000              MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2684 041242  004737  016326              JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2685 041246  103407                      BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
2686 041250  010001                      MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2690 041252                              ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     041252  104455                                                              TRAP    C$ERDF
     041254  001765                                                              .WORD   1013
     041256  043070                                                              .WORD   T10SSR
     041260  012036                                                              .WORD   PKTSSR
2691 041262  005237  002220              INC     FATFLG                  ;SET FATAL ERROR FLAG
2692 041266                        45$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041266  104406                                                              TRAP    C$CLP1
2693 041270  005737  002222              TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2694 041274  001004                      BNE     52$                     ;BRANCH IF YES
2698 041276                              ERRHRD  ERRNO,T10INT,PKTSSR
     041276  104456                                                              TRAP    C$ERHRD
     041300  001766                                                              .WORD   1014
     041302  043250                                                              .WORD   T10INT
     041304  012036                                                              .WORD   PKTSSR
2699 041306  016501  000002        52$:  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
2700 041312  012702  000200              MOV     #SSR,P2                 ;EXPECTED CONTENTS OF TSSR
2701 041316  032701  000100              BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
2702 041322  001402                      BEQ     55$                     ;BRANCH IF NOT OFF-LINE
2703 041324  052702  000100              BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2704 041330  020201                55$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2705 041332  001404                      BEQ     60$                     ;OKAY IF MATCH
2709 041334                              ERRHRD  ERRNO,T10NNBA,PKTSSR    ;NBA NOT SET
     041334  104456                                                              TRAP    C$ERHRD
     041336  001767                                                              .WORD   1015
     041340  043013                                                              .WORD   T10NNBA
     041342  012036                                                              .WORD   PKTSSR
2710 041344                        60$:
2711 041344  013701  042552              MOV     T10BFR,R1               ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
2712 041350  012702  025252              MOV     #025252,R2              ;SET UP EXPECTED DATA
2713 041354  020102                      CMP     R1,R2                   ;WAS ANY MESSAGE REC'D
2714 041356  001404                      BEQ     70$                     ;BR, IF OK (EQUAL)
```

```
2718 041360                                ERRHRD  ERRNO,T10MBF,EXPREC     ;MESSAGE BUFFER WAS MODIFIED
     041360  104456                                                                TRAP    C$ERHRD
     041362  001770                                                                .WORD   1016
     041364  042634                                                                .WORD   T10MBF
     041366  015464                                                                .WORD   EXPREC
2719
2720 041370                           70$:
2721 041370  005737  002220                 TST     FATFLG                  ;ANY FATAL ERRORS
2722 041374  001402                         BEQ     80$                     ;BR, IF NO FATAL ERRORS
2723 041376  004737  017172                 JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
2724 041402                           80$:
2725 041402                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     041402                                                                     10001$:
     041402  104405                                                                TRAP    C$ESEG
2726 041404                                ENDSUB                          ;////////////// END SUBTEST //////////////
     041404                                                                     L10077:
     041404  104403                                                                TRAP    C$ESUB
```

TEST 10: BASIC PACKET PROTOCOL

```
2728                                    ;*
2729                                    ;TEST 10 SUBTEST 3
2730                                    ;
2731                                    ;CHECKS THAT THE CPU GIVES UP OWNERSHIP OF THE MESSAGE BUFFER
2732                                    ;AFTER THE MESSAGE BUFFER RELEASE, AND THAT FOLLOWING COMMANDS
2733                                    ;WORK CORRECTLY
2734                                    ;
2735                                    ;
2736 041406                                    BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     041406                                                                                     T10.3:
     041406  104402                                                                                TRAP    C$BSUB
2737
2738 041410  004737  043366                    JSR     PC,T10RST               ;SET PACKET TO INITIAL VALUES
2739 041414                                    SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     041414  012700  000000                                                                     MOV     #PRI00,R0
     041420  104441                                                                             TRAP    C$SPRI
2740 041422  012704  042530                    MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
2741 041426  012764  000010  000006           MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
2742 041434                            5$:
2743 041434                                    BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     041434  104404                                                                             TRAP    C$BSEG
2744
2745 041436  004737  015764                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2746 041442  103405                            BCS     10$                     ;BR IF SOFT INIT = OK
2750 041444  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2751 041446                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     041446  104455                                                                             TRAP    C$ERDF
     041450  001771                                                                             .WORD   1017
     041452  003646                                                                             .WORD   SFIERR
     041454  012024                                                                             .WORD   SFIMSG
2752 041456  005037  002220            10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
2753 041462  005037  002222                    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2754 041466  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2755 041472  004737  016326                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2756 041476  103407                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
2757 041500  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2761 041502                                    ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     041502  104455                                                                                 TRAP    C$ERDF
     041504  001772                                                                                 .WORD   1018
     041506  043070                                                                                 .WORD   T10SSR
     041510  012036                                                                                 .WORD   PKTSSR
2762 041512  005237  002220                    INC     FATFLG                  ;SET FATAL ERROR FLAG
2763 041516                            15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041516  104406                                                                                 TRAP    C$CLP1
2764 041520                                    ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     041520  104410                                                                                 TRAP    C$ESCAPE
     041522  000056                                                                                 .WORD   10000$-.
2765 041524  005737  002222                    TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2766 041530  001004                            BNE     22$                     ;BRANCH IF YES
2770 041532                                    ERRHRD  ERRNO,T10NINT,PKTSSR
     041532  104456                                                                                 TRAP    C$ERHRD
     041534  001773                                                                                 .WORD   1019
     041536  043157                                                                                 .WORD   T10NINT
     041540  012036                                                                                 .WORD   PKTSSR
2771 041542  016501  000002            22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
2772 041546  012702  000200                    MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
2773 041552  032701  000100                    BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
```

TEST 10: BASIC PACKET PROTOCOL

```
2774 041556  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
2775 041560  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2776 041564  020201                  25$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2777 041566  001404                        BEQ     30$                     ;OKAY IF MATCH
2781 041570                                ERRHRD  ERRNO,T10NBA,PKTSSR     ;NBA NOT ZERO
     041570  104456                                                                TRAP    C$ERHRD
     041572  001774                                                                .WORD   1020
     041574  042731                                                                .WORD   T10NBA
     041576  012036                                                                .WORD   PKTSSR
2782 041600                          30$:
2783 041600                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     041600                                                                   10000$:
     041600  104405                                                                TRAP    C$ESEG
2784 041602                                BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     041602  104404                                                                TRAP    C$BSEG
2785 041604  004737  043366                JSR     PC,T10RST               ;RESET PACKETS AND COMMANDS
2786 041610  005037  002222                CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2787 041614  012737  025252  042552        MOV     #025252,T10BFR          ;WIPE OUT MESSAGE BUFFER AREA
2788 041622  012714  100212                MOV     #100212,(R4)            ;SET COMMAND PACKET TO MESS BUF REL
2789 041626  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2790 041632  004737  016326                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
2791 041636  103407                        BCS     45$                     ;BR IF CARRY SET (GOOD RETURN)
2792 041640  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2796 041642                                ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     041642  104455                                                                TRAP    C$ERDF
     041644  001775                                                                .WORD   1021
     041646  043070                                                                .WORD   T10SSR
     041650  012036                                                                .WORD   PKTSSR
2797 041652  005237  002220                INC     FATFLG                  ;SET FATAL ERROR FLAG
2798 041656                          45$:  CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     041656  104406                                                                TRAP    C$CLP1
2799 041660  005737  002222                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2800 041664  001404                        BEQ     52$                     ;BRANCH IF NO
2804 041666                                ERRHRD  ERRNO,T10INT,PKTSSR
     041666  104456                                                                TRAP    C$ERHRD
     041670  001776                                                                .WORD   1022
     041672  043250                                                                .WORD   T10INT
     041674  012036                                                                .WORD   PKTSSR
2805 041676  016501  000002          52$:  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
2806 041702  012702  000200                MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
2807 041706  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
2808 041712  001402                        BEQ     55$                     ;BRANCH IF NOT OFF-LINE
2809 041714  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
2810 041720  020201                  55$:  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
2811 041722  001404                        BEQ     60$                     ;OKAY IF MATCH
2815 041724                                ERRHRD  ERRNO,T10NNBA,PKTSSR    ;NBA NOT SET
     041724  104456                                                                TRAP    C$ERHRD
     041726  001777                                                                .WORD   1023
     041730  043013                                                                .WORD   T10NNBA
     041732  012036                                                                .WORD   PKTSSR
2816 041734                          60$:
2817 041734  013701  042552                MOV     T10BFR,R1               ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
2818 041740  012702  025252                MOV     #025252,R2              ;SET UP EXPECTED DATA
2819 041744  020102                        CMP     R1,R2                   ;WAS ANY MESSAGE REC'D
2820 041746  001404                        BEQ     70$                     ;BR, IF OK (EQUAL)
2824 041750                                ERRHRD  ERRNO,T10MBF,EXPREC     ;MESSAGE BUFFER WAS MODIFIED
     041750  104456                                                                TRAP    C$ERHRD
```

TEST 10: BASIC PACKET PROTOCOL

```
        041752  002000                                                          .WORD   1024
        041754  042634                                                          .WORD   T10MBF
        041756  015464                                                          .WORD   EXPREC
   2825
   2826 041760                          70$:    CKLOOP                          ;LOOP ON ERROR IF FLAG SET
        041760  104406                                                          TRAP    C$CLP1
   2827
   2828 041762  005037  002222                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
   2829 041766  004737  043366                  JSR     PC,T10RST               ;RESET THE PACKETS AND COMMANDS
   2830 041772  042714  100000                  BIC     #100000,(R4)            ;CLEAR THE ACK BIT
   2831 041776  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
   2832 042002  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
   2833 042006  103407                          BCS     75$                     ;BR IF CARRY SET (GOOD RETURN)
   2834 042010  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
   2838 042012                                  ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
        042012  104455                                                          TRAP    C$ERDF
        042014  002001                                                          .WORD   1025
        042016  043070                                                          .WORD   T10SSR
        042020  012036                                                          .WORD   PKTSSR
   2839 042022  005237  002220                  INC     FATFLG                  ;SET FATAL ERROR FLAG
   2840 042026                          75$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
        042026  104406                                                          TRAP    C$CLP1
   2841 042030                                  ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
        042030  104410                                                          TRAP    C$ESCAPE
        042032  000062                                                          .WORD   10001$-.
   2842 042034  005737  002222                  TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
   2843 042040  001006                          BNE     82$                     ;BRANCH IF YES
   2847 042042  016500  000002                  MOV     TSSR(R5),R0             ;GET TSSR FOR ERROR REPORT
   2848 042046                                  ERRHRD  ERRNO,T10NINT,PKTSSR
        042046  104456                                                          TRAP    C$ERHRD
        042050  002002                                                          .WORD   1026
        042052  043157                                                          .WORD   T10NINT
        042054  012036                                                          .WORD   PKTSSR
   2849 042056  016501  000002          82$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
   2850 042062  012702  000200                  MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
   2851 042066  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
   2852 042072  001402                          BEQ     85$                     ;BRANCH IF NOT OFF-LINE
   2853 042074  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
   2854 042100  020201                  85$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
   2855 042102  001404                          BEQ     90$                     ;OKAY IF MATCH
   2859 042104                                  ERRHRD  ERRNO,T10SSR,PKTSSR     ;NBA NOT ZERO
        042104  104456                                                          TRAP    C$ERHRD
        042106  002003                                                          .WORD   1027
        042110  043070                                                          .WORD   T10SSR
        042112  012036                                                          .WORD   PKTSSR
   2860 042114                          90$:
   2861 042114                                  ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        042114                                                                  10001$:
        042114  104405                                                          TRAP    C$ESEG
   2862 042116  005737  002220                  TST     FATFLG                  ;ANY FATAL ERRORS
   2863 042122  001403                          BEQ     95$                     ;BR, IF NO FATAL ERRORS
   2864 042124  004737  017172                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
   2865
   2866 042130                                  BGNSEG                          ;<<<<<<<<<<<<< BGN SEGMENT <<<<<<<<<<<<<
        042130  104404                                                          TRAP    C$BSEG
   2867 042132  005037  002222          95$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
   2868 042136  004737  043366                  JSR     PC,T10RST               ;RESET THE PACKETS AND COMMANDS
```

TEST 10: BASIC PACKET PROTOCOL SEQ 0195

```
 2869 042142  010465  000000                MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 2870 042146  004737  016326                JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 2871 042152  103407                        BCS     100$                    ;BR IF CARRY SET (GOOD RETURN)
 2872 042154  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 2876 042156                                ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      042156  104455                                                            TRAP    C$ERDF
      042160  002004                                                           .WORD   1028
      042162  043070                                                           .WORD   T10SSR
      042164  012036                                                           .WORD   PKTSSR
 2877 042166  005237  002220                INC     FATFLG                  ;SET FATAL ERROR FLAG
 2878 042172                        100$:   CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      042172  104406                                                            TRAP    C$CLP1
 2879 042174                                ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
      042174  104410                                                            TRAP    C$ESCAPE
      042176  000062                                                           .WORD   10002$-.
 2880 042200  005737  002222                TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 2881 042204  001006                        BNE     112$                    ;BRANCH IF YES
 2885 042206  016500  000002                MOV     TSSR(R5),R0             ;GET TSSR FOR ERROR REPORT
 2886 042212                                ERRHRD  ERRNO,T10NINT,PKTSSR
      042212  104456                                                            TRAP    C$ERHRD
      042214  002005                                                           .WORD   1029
      042216  043157                                                           .WORD   T10NINT
      042220  012036                                                           .WORD   PKTSSR
 2887 042222  016501  000002        112$:   MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 2888 042226  012702  000200                MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
 2889 042232  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 2890 042236  001402                        BEQ     125$                    ;BRANCH IF NOT OFF-LINE
 2891 042240  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 2892 042244  020201        125$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 2893 042246  001405                        BEQ     130$                    ;OKAY IF MATCH
 2897 042250                                ERRHRD  ERRNO,T10SSR,PKTSSR     ;NBA NOT ZERO
      042250  104456                                                            TRAP    C$ERHRD
      042252  002006                                                           .WORD   1030
      042254  043070                                                           .WORD   T10SSR
      042256  012036                                                           .WORD   PKTSSR
 2898 042260                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      042260                                                                10002$:
      042260  104405                                                            TRAP    C$ESEG
 2899 042262  005737  002220        130$:   TST     FATFLG                  ;CHECK FATAL FLAG FOR SET
 2900 042266  001402                        BEQ     140$                    ;BR, IF NOT SET (OK)
 2901 042270  004737  017172                JSR     PC,CKDROP               ;DROP DEVICE IF INDICATED
 2902 042274                        140$:
 2903 042274                                ENDSUB                          ;////////////// END SUBTEST //////////////
      042274                                                                L10100:
      042274  104403                                                            TRAP    C$ESUB
 2904 042276                        30$:
```

```
2906                                    ;+
2907                                    ;TEST 10 SUBTEST 4
2908                                    ;
2909                                    ;CHECKS THAT THE REGISTER MODIFICATION REFUSED (RMR) BIT IN
2910                                    ;THE TSSR WILL BE SET IF A WRITE CHARACTERISTICS COMMAND
2911                                    ;BEING EXECUTED AND ANOTHER "WC" COMMAND IS ATTEMPTED
2912                                    ;
2913                                    ;-
2914
2915 042276                                    BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
      042276                                                                                  T10.4:
      042276  104402                                                                           TRAP    C$BSUB
2916
2917 042300  004737  043440                    JSR     PC,T10RT2               ;SET SECOND PACKET UP
2918 042304  004737  043366                    JSR     PC,T10RST               ;SET PACKET TO INITIAL VALUES
2919 042310                                    SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      042310  012700  000000                                                                  MOV     #PRI00,R0
      042314  104441                                                                          TRAP    C$SPRI
2920 042316  012704  042530                    MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
2921 042322  012703  042572                    MOV     #T10PKT,R3              ;GET THE ADDRESS OF 2ND CMD PACKET
2922 042326  012764  000010  000006            MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
2923 042334  012763  000010  000006            MOV     #8.,PKBCNT(R3)          ;START WITH MINIMUM ALLOWABLE VALUE
2924 042342                                5$:
2925 042342                                    BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      042342  104404                                                                          TRAP    C$BSEG
2926 042344  004737  015764                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
2927 042350  103405                            BCS     10$                     ;BR IF SOFT INIT = OK
2931 042352  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
2932 042354                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      042354  104455                                                                          TRAP    C$ERDF
      042356  002007                                                                          .WORD   1031
      042360  003646                                                                          .WORD   SFIERR
      042362  012024                                                                          .WORD   SFIMSG
2933 042364  005037  002220            10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
2934 042370  005037  002222                    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
2935 042374  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
2936 042400  010365  000000                    MOV     R3,TSDB(R5)             ;SECOND COMMAND PACKET
2937 042404  004737  016240                    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
2938 042410  016501  000002                    MOV     TSSR(R5),R1             ;GET CONTENTS OF TSSR REGISTER
2939 042414  032701  000200                    BIT     #SSR,R1                 ;CHECK FOR SSR (TSSR) SET
2940 042420  001006                            BNE     15$                     ;BR, IF SSR SET (GOOD)
2944 042422                                    ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      042422  104455                                                                          TRAP    C$ERDF
      042424  002010                                                                          .WORD   1032
      042426  043070                                                                          .WORD   T10SSR
      042430  012036                                                                          .WORD   PKTSSR
2945 042432  005237  002220                    INC     FATFLG                  ;SET FATAL ERROR FLAG
2946 042436                                15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      042436  104406                                                                          TRAP    C$CLP1
2947 042440                                    ESCAPE  SEG                     ;BY PASS SUBTEST IF FATAL ERROR
      042440  104410                                                                          TRAP    C$ESCAPE
      042442  000056                                                                          .WORD   10000$ .
2948 042444  005737  002222                    TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
2949 042450  001004                            BNE     22$                     ;BRANCH IF YES
2950
2951
2955 042452                                    ERRHRD  ERRNO,T10NINT,PKTSSR
```

TEST 10: BASIC PACKET PROTOCOL

```
        042452  104456                                                          TRAP    C$ERHRD
        042454  002011                                                          .WORD   1033
        042456  043157                                                          .WORD   T1ONINT
        042460  012036                                                          .WORD   PKTSSR
  2956  042462  016501  000002        22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
  2957  042466  012702  110200                MOV     #SSR!RMR!SC,R2          ;EXPECTED CONTENTS OF TSSR
  2958  042472  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
  2959  042476  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
  2960  042500  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
  2961  042504  020201                25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
  2962  042506  001404                        BEQ     30$                     ;OKAY IF MATCH
  2966  042510                                ERRHRD  ERRNO,T1OSSR,PKTSSR     ;NBA NOT ZERO
        042510  104456                                                          TRAP    C$ERHRD
        042512  002012                                                          .WORD   1034
        042514  043070                                                          .WORD   T1OSSR
        042516  012036                                                          .WORD   PKTSSR
  2967  042520                        30$:
  2968  042520                                ENDSEG                          ;<<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<<
        042520                                                                  10000$:
        042520  104405                                                          TRAP    C$ESEG
  2969  042522                                ENDSUB                          ;////////////// END SUBTEST //////////////
        042522                                                                  L10101:
        042522  104403                                                          TRAP    C$ESUB
  2970  042524                                EXIT    TST                     ;ALL DONE WITH THIS TEST
        042524  104432                                                          TRAP    C$EXIT
        042526  000764                                                          .WORD   L10075-.
```

TEST 10: BASIC PACKET PROTOCOL

```
2972                                    ;*
2973                                    ;LOCAL STORAGE FOR THIS TEST
2974                                    ;
2975
2979 042530                             T10PACKET:                      ;COMMAND PACKET FOR TEST
2980 042530  100204                             .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
2981 042532  042540                             .WORD   T10DATA         ;ADDRESS OF CHARACTERISTICS BLOCK
2982 042534  000000                             .WORD   0
2983 042536  000010                             .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
2984
2985 042540                             T10DATA:                                ;CHARACTERISTICS DATA BLOCK
2986 042540  042552                             .WORD   T10BFR          ;ADDRESS OF MESSAGE BUFFER
2987 042542  000000                             .WORD   0
2988 042544  000016                             .WORD   14.             ;LENGTH OF MESSAGE BUFFER
2989 042546  000000  000000                     .WORD   0,0
2990
2991 042552                             T10BFR: .BLKW   8.              ;MESSAGE BUFFER
2992
2993                                    ;*
2994                                    ;
2995                                    ;TEST DATA FOR SUBTEST FOUR
2996                                    ;
2997 042572                             T10PKT:                         ;COMMAND PACKET FOR TEST
2998 042572  100204                             .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
2999 042574  042602                             .WORD   T10DTA          ;ADDRESS OF CHARACTERISTICS BLOCK
3000 042576  000000                             .WORD   0
3001 042600  000010                             .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
3002
3003 042602                             T10DTA:                         ;CHARACTERISTICS DATA BLOCK
3004 042602  042614                             .WORD   T10BUFR         ;ADDRESS OF MESSAGE BUFFER
3005 042604  000000                             .WORD   0
3006 042606  000016                             .WORD   14.             ;LENGTH OF MESSAGE BUFFER
3007 042610  000000  000000                     .WORD   0,0
3008
3009 042614                             T10BUFR:.BLKW   8.              ;MESSAGE BUFFER
3010
3011                                    ;*
3012                                    ;LOCAL TEXT MESSAGES FOR TEST
3013                                    ;-
3014
3015
3016 042634      115     145     163    T10MBF: .ASCIZ  'Message Buffer Modified after MESSAGE BUFFER RELEASE Command'
3017 042731      116     102     101    T10NBA: .ASCIZ  'NBA Not Clear After WRITE CHARACTERISTICS Command'
3018 043013      116     102     101    T10NNBA:        .ASCIZ  'NBA Set After MESSAGE BUFFER RELEASE Command'
3019 043070      103     157     156    T10SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
3020 043157      105     170     160    T10NINT:        .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS
3021 043250      125     156     145    T10INT: .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS
3022 043337      102     141     163    TST10ID:        .ASCIZ  'Basic Packet Protocol'
3023                                            .EVEN
3024
```

TEST 10: BASIC PACKET PROTOCOL

```
3026
3027                                    ;*
3028                                    ;
3029                                    ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
3030                                    ;
3031                                    ;-
3032
3033 043366                             T10RST:
3034 043366                                     SAVREG                  ;SAVE THE REGISTERS
3035 043372  012701  042530                       MOV     #T10PACKET,R1   ;START OF THE PACKET
3036 043376  012721  100204                       MOV     #100204,(R1)+   ;WRITE CHARACTERISTICS WITH ACK, IE
3037 043402  012721  042540                       MOV     #T10DATA,(R1)+  ;ADDRESS OF CHAR DATA BLOCK
3038 043406  005021                               CLR     (R1)+           ;EXTENDED ADDRESS
3039 043410  012721  000010                       MOV     #8.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
3040 043414  012721  042552                       MOV     #T10BFR,(R1)+   ;ADDRESS OF MESSAGE BUFFER
3041 043420  005021                               CLR     (R1)+
3042 043422  012721  000016                       MOV     #14.,(R1)+      ;LENGTH OF MESSAGE BUFFER
3043 043426  005021                               CLR     (R1)+
3044 043430  005011                               CLR     (R1)
3045 043432  005037  042552                       CLR     T10BFR          ;CLEAR 1ST LOC IN MESSAGE BUFFER
3046 043436  000207                               RTS     PC              ;RETURN
3047                                    ;*
3048                                    ;
3049                                    ;ROUTINE TO RESTORE COMMAND PACKET #2 TO START UP (DEFAULT) VALUES
3050                                    ;
3051                                    ;
3052
3053 043440                             T10RT2:
3054 043440                                     SAVREG                  ;SAVE THE REGISTERS
3055 043444  012701  042572                       MOV     #T10PKT,R1      ;START OF THE PACKET
3056 043450  012721  100204                       MOV     #100204,(R1)+   ;WRITE CHARACTERISTICS WITH ACK, IE
3057 043454  012721  042602                       MOV     #T10DTA,(R1)+   ;ADDRESS OF CHAR DATA BLOCK
3058 043460  005021                               CLR     (R1)+           ;EXTENDED ADDRESS
3059 043462  012721  000010                       MOV     #8.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
3060 043466  012721  042614                       MOV     #T10BUFR,(R1)+  ;ADDRESS OF MESSAGE BUFFER
3061 043472  005021                               CLR     (R1)+
3062 043474  012721  000016                       MOV     #14.,(R1)+      ;LENGTH OF MESSAGE BUFFER
3063 043500  005021                               CLR     (R1)+
3064 043502  005011                               CLR     (R1)
3065 043504  005037  042614                       CLR     T10BUFR         ;CLEAR 1ST LOC IN MESSAGE BUFFER
3066 043510  000207                               RTS     PC              ;RETURN
3067 043512                             ENDTST
     043512                                                                      L10075:
     043512  104401                                                                    TRAP    C$ETST
```

```
                                        .SBTTL  TEST  11: NON TAPE MOTION COMMANDS

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

043514                                  BGNTST
043514                                                                  T11::
043514  012700  045512                    MOV     #TST1,ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
043520  004737  016500                    JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
043524  012737  000024  002214            MOV     #20.,LOOPCNT            ;PERFORM 20 ITERATIONS
043532                          T11LOOP:
043532                                  BGNSUB                          ;/////////// BEGIN SUBTEST ///////////
043532                                                                          T11.1:
043532  104402                                                                  TRAP    C$BSUB

043534                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
043534  012700  000000                                                          MOV     #PRI00,R0
043540  104441                                                                  TRAP    C$SPRI
043542  004737  015764                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
043546  103405                          BCS     3$                      ;BR IF SOFT INIT = OK
043550  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043552                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
043552  104455                                                                  TRAP    C$ERDF
043554  002115                                                                  .WORD   1101
043556  003646                                                                  .WORD   SFIERR
043560  012024                                                                  .WORD   SFIMSG
043562                          3$:
043562  012704  044740                    MOV     #T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
043566  004737  010652                    JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
043572  103404                          BCS     4$                      ;BR, IF COMMAND ISSUED OK
043574                                  ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
043574  104456                                                                  TRAP    C$ERHRD
043576  002116                                                                  .WORD   1102
043600  005052                                                                  .WORD   WRTMSG
043602  012024                                                                  .WORD   SFIMSG
043604                          4$:
043604  004737  045544                    JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
043610  012704  044670                    MOV     #T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
043614                          5$:
043614                                  BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
043614  104404                                                                  TRAP    C$BSEG

043616  005037  002220          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
043622  005037  002222                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
043626  010465  000000                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
043632  004737  016326                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
043636  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
043640  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
043642                                  ERRDF   ERRNO,T11SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
043642  104455                                                                  TRAP    C$ERDF
```

```
        043644  002117                                                          .WORD   1103
        043646  045234                                                          .WORD   T11SSR
        043650  012036                                                          .WORD   PKTSSR
  3125  043652  005237  002220            INC     FATFLG          ;SET FATAL ERROR FLAG
  3126  043656                    15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
        043656  104406                                                          TRAP    C$CLP1
  3127  043660                            ESCAPE  SEG             ;BY PASS SUBTEST IF FATAL ERROR
        043660  104410                                                          TRAP    C$ESCAPE
        043662  000074                                                          .WORD   10000$-.
  3128  043664  005737  002222            TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
  3129  043670  001004                    BNE     22$             ;BRANCH IF YES
  3133  043672                            ERRHRD  ERRNO,T11NINT,PKTSSR
        043672  104456                                                          TRAP    C$ERHRD
        043674  002120                                                          .WORD   1104
        043676  045364                                                          .WORD   T11NINT
        043700  012036                                                          .WORD   PKTSSR
  3134  043702  016501  000002    22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
  3135  043706  012702  000200            MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
  3136  043712  032701  000100            BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
  3137  043716  001402                    BEQ     25$             ;BRANCH IF NOT OFF-LINE
  3138  043720  052702  000100            BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
  3139  043724  020201    25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
  3140  043726  001404                    BEQ     30$             ;OKAY IF MATCH
  3144  043730    -                       ERRHRD  ERRNO,T11NBA,PKTSSR     ;NBA NOT ZERO
        043730  104456                                                          TRAP    C$ERHRD
        043732  002121                                                          .WORD   1105
        043734  045002                                                          .WORD   T11NBA
        043736  012036                                                          .WORD   PKTSSR
  3145  043740                    30$:
  3146  043740  004737  011104    35$:    JSR     PC,CKRAM        ;CHECK RAM TO MEMORY
  3147  043744  103405                    BCS     59$             ;RAM OK GO ON
  3151  043746                            ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
        043746  104456                                                          TRAP    C$ERHRD
        043750  002122                                                          .WORD   1106
        043752  004741                                                          .WORD   PKTRAM
        043754  015500                                                          .WORD   RAMERR
  3152  043756                            ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        043756                                                          10000$:
        043756  104405                                                          TRAP    C$ESEG
  3153
  3154
  3155  043760                    59$:    ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        043760                                                          L10103:
        043760  104403                                                          TRAP    C$ESUB
  3156
  3157  043762  005737  002220            TST     FATFLG          ;ANY FATAL ERRORS ?
  3158  043766  001402                    BEQ     60$             ;BRANCH IF NOT
  3159  043770  004737  017172            JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
  3160  043774                    60$:
```

TEST 11: NON TAPE MOTION COMMANDS

```
3162                                    ;+
3163                                    ;
3164                                    ;TEST 11, SUBTEST 2
3165                                    ;
3166                                    ;CHECK THAT NON-ZERO MODE BITS BEING SET CAUSES
3167                                    ;INITIALIZE COMMAND TO BE REJECTED
3168                                    ;
3169                                    ;-
3170
3171 043774                                     BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
      043774                                                                                    T11.2:
      043774  104402                                                                           TRAP    C$BSUB
3172
3173 043776                                     SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      043776  012700  000000                                                                    MOV     #PRI00,R0
      044002  104441                                                                           TRAP    C$SPRI
3174 044004                                     BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
      044004  104404                                                                           TRAP    C$BSEG
3175
3176
3177 044006  004737  015764                     JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
3178 044012  103405                             BCS     3$                      ;BR IF SOFT INIT = OK
3182 044014  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3183 044016                                     ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      044016  104455                                                                           TRAP    C$ERDF
      044020  002123                                                                           .WORD   1107
      044022  003646                                                                           .WORD   SFIERR
      044024  012024                                                                           .WORD   SFIMSG
3184 044026                             3$:
3185 044026  012704  04474C                     MOV     #T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
3186 044032  004737  010652                     JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
3187 044036  103404                             BCS     4$                      ;BR, IF COMMAND ISSUED OK
3191 044040                                     ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      044040  104456                                                                           TRAP    C$ERHRD
      044042  002124                                                                           .WORD   1108
      044044  005052                                                                           .WORD   WRTMSG
      044046  012024                                                                           .WORD   SFIMSG
3192 044050                             4$:
3193 044050  004737  045544                     JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
3194 044054  012704  044670                     MOV     #T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3195 044060                             5$:
3196 044060  005037  002222             10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
3197 044064  052714  007400                     BIS     #P.MODE,(R4)            ;NON-ZERO COMMAND MODE BITS
3198 044070  010465  000000                     MOV     R4,T5DB(R5)             ;SET THE PACKET ADDRESS
3199 044074  004737  016326                     JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3200 044100  103405                             BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
3201 044102  010001                             MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3205 044104                                     ERRDF   ERRNO,T11SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
      044104  104455                                                                           TRAP    C$ERDF
      044106  002125                                                                           .WORD   1109
      044110  045234                                                                           .WORD   T11SSR
      044112  012036                                                                           .WORD   PKTSSR
3206 044114                             15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      044114  104406                                                                           TRAP    C$CLP1
3207 044116                                     ESCAPE  SEG                     ;BY PASS CHECKS IF FATAL ERROR
      044116  104410                                                                           TRAP    C$ESCAPE
      044120  000074                                                                           .WORD   10000$ .
```

TEST 11: NON-TAPE MOTION COMMANDS

```
3208 044122  005737  002222            TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
3209 044126  001004                    BNE     22$                     ;BRANCH IF YES
3213 044130                            ERRHRD  ERRNO,T11NINT,PKTSSR
     044130  104456                                                    TRAP    C$ERHRD
     044132  002126                                                    .WORD   1110
     044134  045364                                                    .WORD   T11NINT
     044136  012036                                                    .WORD   PKTSSR
3214 044140  016501  000002      22$:   MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
3215 044144  012702  100206            MOV     #SC!SSR!TSREJ,R2        ;EXPECTED CONTENTS OF TSSR
3216 044150  032701  000100            BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
3217 044154  001402                    BEQ     25$                     ;BRANCH IF NOT OFF-LINE
3218 044156  052702  000100            BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
3219 044162  020201      25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
3220 044164  001404                    BEQ     30$                     ;OKAY IF MATCH
3224 044166                            ERRHRD  ERRNO,T112REJ,PKTSSR    ;COMMAND NOT REJECTED
     044166  104456                                                    TRAP    C$ERHRD
     044170  002127                                                    .WORD   1111
     044172  045042                                                    .WORD   T112REJ
     044174  012036                                                    .WORD   PKTSSR
3225 044176              30$:
3226 044176  004737  011104      35$:   JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
3227 044202  103405                    BCS     59$                     ;RAM OK GO ON
3231 044204                            ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
     044204  104456                                                    TRAP    C$ERHRD
     044206  002130                                                    .WORD   1112
     044210  004741                                                    .WORD   PKTRAM
     044212  015500                                                    .WORD   RAMERR
3232 044214                            ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     044214                                                         10000$:
     044214  104405                                                    TRAP    C$ESEG
3233
3234 044216              59$:   ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     044216                                                         L10104:
     044216  104403                                                    TRAP    C$ESUB
```

```
3236                                    ;*
3237                                    ;
3238                                    ;TEST 11, SUBTEST 3
3239                                    ;
3240                                    ;CHECK THAT THE GET STATUS COMMAND IS ACCEPTED
3241                                    ;
3242                                    ;
3243
3244 044220                                    BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
     044220                                                                                  T11.3:
     044220  104402                                                                            TRAP    C$BSUB
3245
3246 044222                                    SETPRI  @PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     044222  012700  000000                                                                    MOV     @PRI00,R0
     044226  104441                                                                            TRAP    C$SPRI
3247 044230                                    BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     044230  104404                                                                            TRAP    C$BSEG
3248 044232  004737  015764                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
3249 044236  103405                            BCS     3$                      ;BR IF SOFT INIT = OK
3253 044240  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3254 044242                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     044242  104455                                                                            TRAP    C$ERDF
     044244  002131                                                                            .WORD   1113
     044246  003646                                                                            .WORD   SFIERR
     044250  012024                                                                            .WORD   SFIMSG
3255 044252                            3$:
3256 044252  012704  044740                    MOV     @T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
3257 044256  004737  010652                    JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
3258 044262  103404                            BCS     4$                      ;BR, IF COMMAND ISSUED OK
3262 044264                                    ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     044264  104456                                                                            TRAP    C$ERHRD
     044266  002132                                                                            .WORD   1114
     044270  005052                                                                            .WORD   WRTMSG
     044272  012024                                                                            .WORD   SFIMSG
3263 044274                            4$:
3264 044274  004737  045544                    JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
3265 044300  012704  044670                    MOV     @T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3266 044304                            5$:
3267 044304  005037  002222            10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
3268 044310  010465  000000                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
3269 044314  004737  016326                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3270 044320  103405                            BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
3271 044322  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3275 044324                                    ERRDF   ERRNO,T11SR2,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     044324  104455                                                                            TRAP    C$ERDF
     044326  002133                                                                            .WORD   1115
     044330  045310                                                                            .WORD   T11SR2
     044332  012036                                                                            .WORD   PKTSSR
3276 044334                            15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     044334  104406                                                                            TRAP    C$CLP1
3277 044336                                    ESCAPE  SL3                     ;BY-PASS SUBTEST IF FATAL ERROR
     044336  104410                                                                            TRAP    C$ESCAPE
     044340  000074                                                                            .WORD   10000$ .
3278 044342  005737  002222                    TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
3279 044346  001004                            BNE     22$                     ;BRANCH IF YES
3283 044350                                    ERRHRD  ERRNO,T11NINT,PKTSSR
     044350  104456                                                                            TRAP    C$ERHRD
```

TEST 11: NON-TAPE MOTION COMMANDS

```
        044352  002134                                                          .WORD   1116
        044354  045364                                                          .WORD   T11NINT
        044356  012036                                                          .WORD   PKTSSR
  3284  044360  016501  000002          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
  3285  044364  012702  000200                  MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
  3286  044370  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
  3287  044374  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
  3288  044376  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
  3289  044402  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
  3290  044404  001404                          BEQ     30$                     ;OKAY IF MATCH
  3294  044406                                  ERRHRD  ERRNO,T113REJ,PKTSSR    ;COMMAND NOT ACCEPTED
        044406  104456                                                          TRAP    C$ERHRD
        044410  002135                                                          .WORD   1117
        044412  045123                                                          .WORD   T113REJ
        044414  012036                                                          .WORD   PKTSSR
  3295  044416                          30$:
  3296  044416  004737  011104          35$:    JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
  3297  044422  103405                          BCS     59$                     ;RAM OK GO ON
  3301  044424                                  ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
        044424  104456                                                          TRAP    C$ERHRD
        044426  002136                                                          .WORD   1118
        044430  004741                                                          .WORD   PKTRAM
        044432  015500                                                          .WORD   RAMERR
  3302  044434                          ENDSEG                                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        044434                                                                  10000$:
        044434  104405                                                          TRAP    C$ESEG
  3303
  3304
  3305  044436                  59$:    ENDSUB                                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
        044436                                                                  L10105:
        044436  104403                                                          TRAP    C$ESUB
```

```
3307                                    ;*
3308                                    ;
3309                                    ;TEST 11, SUBTEST 4
3310                                    ;
3311                                    ;SUBTEST TO VERIFY THAT A GET STATUS COMMAND IS
3312                                    ;REJECTED IF A NON-ZERO COMMAND MODE FIELD IS USED
3313                                    ;
3314                                    ;-
3315
3316 044440                             BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     044440                                                                     T11.4:
     044440  104402                                                             TRAP    C$BSUB
3317
3318 044442                             SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     044442  012700  000000                                                     MOV     #PRI00,R0
     044446  104441                                                             TRAP    C$SPRI
3319 044450                             BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     044450  104404                                                             TRAP    C$BSEG
3320 044452  004737  015764             JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
3321 044456  103405                     BCS     3$                      ;BR IF SOFT INIT = OK
3325 044460  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3326 044462                             ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     044462  104455                                                             TRAP    C$ERDF
     044464  002137                                                             .WORD   1119
     044466  003646                                                             .WORD   SFIERR
     044470  012024                                                             .WORD   SFIMSG
3327 044472                             3$:
3328 044472  012704  044740             MOV     #T11PK2,R4              ;WRITE CHARACTERISTICS PACKET
3329 044476  004737  010652             JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
3330 044502  103404                     BCS     4$                      ;BR, IF COMMAND ISSUED OK
3334 044504                             ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
     044504  104456                                                             TRAP    C$ERHRD
     044506  002140                                                             .WORD   1120
     044510  005052                                                             .WORD   WRTMSG
     044512  012024                                                             .WORD   SFIMSG
3335 044514                             4$:
3336 044514  004737  045544             JSR     PC,T11REST              ;SET UP PACKET FOR COMMAND
3337 044520  012704  044670             MOV     #T11PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
3338 044524                             5$:
3339 044524  005037  002222             10$:    CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
3340 044530  052714  007000             BIS     #007000,(R4)            ;SET TO NON-ZERO MODE
3341 044534  010465  000000             MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
3342 044540  004737  016326             JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
3343 044544  103405                     BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
3344 044546  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
3348 044550                             ERRDF   ERRNO,T11SR2,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
     044550  104455                                                             TRAP    C$ERDF
     044552  002141                                                             .WORD   1121
     044554  045310                                                             .WORD   T11SR2
     044556  012036                                                             .WORD   PKTSSR
3349 044560                             15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
     044560  104406                                                             TRAP    C$CLP1
3350 044562                             ESCAPE  SUB                     ;BY PASS SUBTEST IF FATAL ERROR
     044562  104410                                                             TRAP    C$FSCAPE
     044564  000076                                                             .WORD   L10106-.
3351 044566  005737  002222             TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
3352 044572  001004                     BNE     22$                     ;BRANCH IF YES
```

TEST 11: NON-TAPE MOTION COMMANDS

```
3356 044574                                ERRHRD  ERRNO,T11NINT,PKTSSR
     044574  104456                                                        TRAP    C$ERHRD
     044576  002142                                                        .WORD   1122
     044600  045364                                                        .WORD   T11NINT
     044602  012036                                                        .WORD   PKTSSR
3357 044604  016501  000002        22$:    MOV     TSSR(R5),R1           ;GET THE CONTENTS OF TSSR
3358 044610  012702  100206              MOV     #SC!SSR!TSREJ,R2      ;EXPECTED CONTENTS OF TSSR
3359 044614  032701  000100              BIT     #OFL,R1               ;IS OFF-LINE BIT SET ?
3360 044620  001402                      BEQ     25$                   ;BRANCH IF NOT OFF-LINE
3361 044622  052702  000100              BIS     #OFL,R2               ;SET OFF-LINE IN EXPECTED DATA
3362 044626  020201        25$:    CMP     R2,R1                 ;DOES EXPECTED MATCH RECEIVED ?
3363 044630  001404                      BEQ     30$                   ;OKAY IF MATCH
3367 044632                              ERRHRD  ERRNO,T114REJ,PKTSSR  ;COMMAND NOT REJECTED
     044632  104456                                                        TRAP    C$ERHRD
     044634  002143                                                        .WORD   1123
     044636  045153                                                        .WORD   T114REJ
     044640  012036                                                        .WORD   PKTSSR
3368 044642                      30$:
3369
3370 044642  004737  011104        35$:    JSR     PC,CKRAM              ;CHECK RAM TO MEMORY
3371 044646  103404                      BCS     59$                   ;RAM OK GO ON
3375 044650                              ERRHRD  ERRNO,PKTRAM,RAMERR   ;THEY DON'T MATCH
     044650  104456                                                        TRAP    C$ERHRD
     044652  002144                                                        .WORD   1124
     044654  004741                                                        .WORD   PKTRAM
     044656  015500                                                        .WORD   RAMERR
3376 044660                      59$:
3377 044660                              ENDSEG                        ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<
     044660                                                                 10000$:
     044660  104405                                                            TRAP    C$ESEG
3378 044662                              ENDSUB                        ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\
     044662                                                                 L10106:
     044662  104403                                                            TRAP    C$ESUB
3379
```

```
3381 044664                                    EXIT    TST             ;ALL DONE THIS TEST
     044664  104432                                                                 TRAP    C$EXIT
     044666  000762                                                                 .WORD   L10102-.
3382
3383                                           ;*
3384                                           ;LOCAL STORAGE FOR THIS TEST
3385                                           ;-
3386
3390 044670                                    T11PACKET:                      ;COMMAND PACKET FOR TEST
3391 044670  100204                                    .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
3392 044672  044700                                    .WORD   T11DATA         ;ADDRESS OF CHARACTERISTICS BLOCK
3393 044674  000000                                    .WORD   0
3394 044676  000010                                    .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
3395
3396 044700                                    T11DATA:                                ;CHARACTERISTICS DATA BLOCK
3397 044700  044712                                    .WORD   T11BFR          ;ADDRESS OF MESSAGE BUFFER
3398 044702  000000                                    .WORD   0
3399 044704  000016                                    .WORD   14.             ;LENGTH OF MESSAGE BUFFER
3400 044706  000000  000000                            .WORD   0,0
3401
3402 044712                                    T11BFR: .BLKW   8.              ;MESSAGE BUFFER
3403
3404
3406         044740                                    .=<.+10>&177770
3408 044740                                    T11PK2:                         ;COMMAND PACKET FOR TEST
3409 044740  100204                                    .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
3410 044742  044750                                    .WORD   T11DTA          ;ADDRESS OF CHARACTERISTICS BLOCK
3411 044744  000000                                    .WORD   0
3412 044746  000010                                    .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
3413
3414 044750                                    T11DTA:                                 ;CHARACTERISTICS DATA BLOCK
3415 044750  044762                                    .WORD   T11BF2          ;ADDRESS OF MESSAGE BUFFER
3416 044752  000000                                    .WORD   0
3417 044754  000016                                    .WORD   14.             ;LENGTH OF MESSAGE BUFFER
3418 044756  000000  000000                            .WORD   0,0
3419
3420 044762                                    T11BF2: .BLKW   8.              ;MESSAGE BUFFER
3421
3422
3423
3424                                           ;*
3425                                           ;LOCAL TEXT MESSAGES FOR TEST
3426                                           ;-
3427
3428 045002      111     116     111   T11NBA: .ASCIZ  'INITIALIZE Command Not Accepted'
3429 045042      111     116     111   T112REJ:        .ASCIZ  'INITIALIZE Not Rejected With Non-Zero Mode Field'
3430 045123      107     105     124   T113REJ:        .ASCIZ  'GET STATUS Not Accepted'
3431 045153      107     105     124   T114REJ:        .ASCIZ  'GET STATUS Not Rejected With Non-Zero Mode Field'
3432 045234      103     157     156   T11SSR: .ASCIZ  'Contents of TSSR Incorrect After INITIALIZE'
3433 045310      103     157     156   T11SR2: .ASCIZ  'Contents of TSSR Incorrect After GET STATUS'
3434 045364      105     170     160   T11NINT:        .ASCIZ  'Expected Interrupt Not Received On INITIALIZE'
3435 045442      111     156     143   T11TSBA:        .ASCIZ  'Incorrect TSBA Address After INITIALIZE'
3436 045512      116     157     156   TST11ID:        .ASCIZ  'Non-Tape Motion Commands'
3437                                           .EVEN
3438
```

TEST 11: NON TAPE MOTION COMMANDS

```
3440
3441                                    ;*
3442                                    ;
3443                                    ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
3444                                    ;INITIALIZE COMMAND
3445                                    ;
3446                                    ;
3447
3448 045544                             T11REST:
3449 045544                                     SAVREG                          ;SAVE THE REGISTERS
3450 045550  012701  044670                       MOV     #T11PACKET,R1       ;START OF THE PACKET
3451 045554  012721  100213                       MOV     #100213,(R1)+       ;INITIALIZE WITH ACK, IE
3452 045560  005021                               CLR     (R1)+               ;ADDRESS OF CHAR DATA BLOCK
3453 045562  005021                               CLR     (R1)+               ;EXTENDED ADDRESS
3454 045564  005021                               CLR     (R1)+               ;SIZE OF DATA BLOCK IN BYTES
3455 045566  005021                               CLR     (R1)+               ;ADDRESS OF MESSAGE BUFFER
3456 045570  005021                               CLR     (R1)+
3457 045572  005021                               CLR     (R1)+               ;LENGTH OF MESSAGE BUFFER
3458 045574  005021                               CLR     (R1)+
3459 045576  005011                               CLR     (R1)
3460 045600  005037  044712                       CLR     T11BFR              ;CLEAR 1ST LOC IN MESSAGE BUFFER
3461 045604  000207                               RTS     PC                  ;RETURN
3462                                    ;*
3463                                    ;
3464                                    ;ROUTINE TO RESTORE COMMAND PACKET TO START UP (DEFAULT) VALUES
3465                                    ;GET STATUS COMMAND
3466                                    ;
3467                                    ;
3468
3469 045606                             T11RT2:
3470 045606                                     SAVREG                          ;SAVE THE REGISTERS
3471 045612  012701  044670                       MOV     #T11PACKET,R1       ;START OF THE PACKET
3472 045616  012721  100217                       MOV     #100217,(R1)+       ;GET STATUS WITH ACK, IE
3473 045622  005021                               CLR     (R1)+               ;ADDRESS OF CHAR DATA BLOCK
3474 045624  005021                               CLR     (R1)+               ;EXTENDED ADDRESS
3475 045626  005021                               CLR     (R1)+               ;SIZE OF DATA BLOCK IN BYTES
3476 045630  005021                               CLR     (R1)+               ;ADDRESS OF MESSAGE BUFFER
3477 045632  005021                               CLR     (R1)+
3478 045634  005021                               CLR     (R1)+               ;LENGTH OF MESSAGE BUFFER
3479 045636  005021                               CLR     (R1)+
3480 045640  005011                               CLR     (R1)
3481 045642  005037  044712                       CLR     T11BFR              ;CLEAR 1ST LOC IN MESSAGE BUFFER
3482 045646  000207                               RTS     PC                  ;RETURN
3483 045650                                     ENDTST
     045650                                                                     L10102:
     045650  104401                                                                 TRAP    C$ETST
3484 045652                             ENDMOD
```

```
 1                                              .TITLE  TSV6 - PARAMETER CODING
 7
12
18
19 045652                                       BGNMOD  TSV6
   045652                               TSV6::
20
21                                              .SBTTL  HARDWARE PARAMETER CODING SECTION
22
23                                              ;++
24                                              ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
25                                              ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
26                                              ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
27                                              ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
28                                              ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
29                                              ; WITH THE OPERATOR.
30                                              ;--
31 045652                                       BGNHRD
   045652  000010                               .WORD L10107-L$HARD/2
   045654                               L$HARD::
32
33 045654                                       GPRMA   HPM1,0,0,160010,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
   045654  000031                               .WORD   T$CODE
   045656  045674                               .WORD   HPM1
   045660  160010                               .WORD   T$LOLIM
   045662  177776                               .WORD   T$HILIM
34 045664                                       GPRMA   HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
   045664  001031                               .WORD   T$CODE
   045666  045730                               .WORD   HPM2
   045670  000000                               .WORD   T$LOLIM
   045672  000776                               .WORD   T$HILIM
35                                              ;GPRMD  HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
36 045674                                       ENDHRD
                                                .EVEN
   045674                               L10107:
37 045674      104     105     126     HPM1:   .ASCIZ  'DEVICE ADDRESS (TSBA/TSDB) '
38 045730      111     116     124     HPM2:   .ASCIZ  'INTERRUPT VECTOR    '
39 045754      111     116     124     HPM3:   .ASCIZ  'INTERRUPT PRIORITY      '
40                                              .EVEN
```

SOFTWARE PARAMETER CODING SECTION

```
                                        .SBTTL  SOFTWARE PARAMETER CODING SECTION

                                ;++
                                ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--
046004                                  BGNSFT
046004  000003                          .WORD L10110-L$SOFT/2
046006                          L$SOFT::
                                ;       GPRML   SPM1,0,-1,YES                   ; GET TRANSPORT TEST FLAG.
046006                                  GPRML   SPM4,2,-1,YES                   ; GET ITERATION CONTROL.
046006  001130                          .WORD   T$CODE
046010  046044                          .WORD   SPM4
046012  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
046014                                  ENDSFT
                                        .EVEN
046014                          L10110:

046014  105  116  101  SPM1:    .ASCIZ  'ENABLE TRANSPORT TESTS '
046044  111  116  110  SPM4:    .ASCIZ  'INHIBIT ITERATIONS     '
046074  120  105  122  SPM6:    .ASCIZ  'PER TEST ERROR LIMIT   '
046124  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).
                                ;

046154                          PATCH::

                                        .BLKW   32.

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

F1

SEQ 0212

SYMBOL TABLE

```
ADDSSR  012116 G        C$AU  = 000052        DEVDRO  023414        FRESIZ  003122 G        INTFLA  016135
ADR     = 000020 G      C$AUTO= 000061        DEVNRD  023333        FUSI    004113          INTMAS  016134
AMBTSS  006631          C$BRK = 000022        DEVNXR  023251        F$AU    = 000015        INTR    016206 G
ASSEMB= 000010          C$BSEG= 000004        DEVONL  023201        F$AUTO= 000020          INTREC  002222 G
A1716 = 000003          C$BSUB= 000002        DEVSUM  023144        F$BGN   = 000040        INTVEC  016136
BADDAT  003152 G        C$CEFG= 000045        DFPTBL  002154 G      F$CLEA= 000007          INTX    004274
BADSSR  015670 G        C$CLCK= 000062        DIAGMC= 000000        F$DU    = 000016        INVERT  021176 G
BDVPCR= 177520 G        C$CLEA= 000012        DICEA = 000001        F$END   = 000041        IOKCKI= 000200
BENBSW  002226 G        C$CLOS= 000035        DSBINT  016174        F$HARD= 000004          IOKSTP= 000001
BIE     = 040000        C$CLP1= 000006        DUAD12  004637        F$HW    = 000013        IPRI    002210 G
BIT0    = 000001 G      C$CVEC= 000036        DUFLG   003106 G      F$INIT= 000006          ISR     = 000100 G
BIT00 = 000001 G        C$DCLN= 000044        DUMMY   003056        F$JMP   = 000050        IVEC    002206 G
BIT01 = 000002 G        C$DODU= 000051        EF.CON= 000036 G      F$MOD   = 000000        IXE     = 004000 G
BIT02 = 000004 G        C$DRPT= 000024        EF.NEW= 000035 G      F$MSG   = 000011        I$AU    = 000041
BIT03 = 000010 G        C$DU  = 000053        EF.PWR= 000034 G      F$PROT= 000021          I$AUTO= 000041
BIT04 = 000020 G        C$EDIT= 000003        EF.RES= 000037 G      F$PWR   = 000017        I$CLN   = 000041
BIT05 = 000040 G        C$ERDF= 000055        EF.STA= 000040 G      F$RPT   = 000012        I$DU    = 000041
BIT06 = 000100 G        C$ERHR= 000056        EMAXDU  016767        F$SEG   = 000003        I$HRD   = 000041
BIT07 = 000200 G        C$ERRO= 000060        EN      = 000000        F$SOFT= 000005          I$INIT= 000041
BIT08 = 000400 G        C$ERSF= 000054        ENAINT  016142        F$SRV   = 000010        I$MOD   = 000041
BIT09 = 001000 G        C$ERSO= 000057        ENVIRN  020620        F$SUB   = 000002        I$MSG   = 000041
BIT1  = 000002 G        C$ESCA= 000010        EPRTSW  002176 G      F$SW    = 000014        I$PROT= 000040
BIT10 = 002000 G        C$ESEG= 000005        EPRT1   006354        F$TEST= 000001          I$PTAB= 000041
BIT11 = 004000 G        C$ESUB= 000003        EPRT2   006354        GDDAT   003154 G        I$PWR   = 000041
BIT12 = 010000 G        C$ETST= 000001        ERCM    011723        GERRMA  002172 G        I$RPT   = 000041
BIT13 = 020000 G        C$EXIT= 000032        ERRHI   002234 G      GETPAT  020164 G        I$SEG   = 000041
BIT14 = 040000 G        C$GETB= 000026        ERRK    016746        GETSEL  020246 G        I$SETU= 000041
BIT15 = 100000 G        C$GETW= 000027        ERRLO   002236 G      G$CNTO= 000200          I$SFT   = 000041
BIT2  = 000004 G        C$GMAN= 000043        ERRNO = 002144        G$DELM= 000372          I$SRV   = 000041
BIT3  = 000010 G        C$GPHR= 000042        ERRVEC= 000004 G      G$DISP= 000003          I$SUB   = 000041
BIT4  = 000020 G        C$GPLO= 000030        ERTABE  003372        G$EXCP= 000400          I$TST   = 000041
BIT5  = 000040 G        C$GPRI= 000040        ERTABL  003172        G$HILI= 000002          J$JMP   = 000167
BIT6  = 000100 G        C$INIT= 000011        ESUM    016750        G$LOLI= 000001          KIPAR0= 172340
BIT7  = 000200 G        C$INLP= 000020        EVL     = 000004 G    G$NO    = 000000        KIPAR1= 172342
BIT8  = 000400 G        C$MANI= 000050        EXBCNT= 000010        G$OFFS= 000400          KIPAR2= 172344
BIT9  = 001000 G        C$MEM = 000031        EXPBRE  015472 G      G$OFSI= 000376          KIPAR3= 172346
BOE     = 000400 G      C$MSG = 000023        EXPD    002230 G      G$PRMA= 000001          KIPAR4= 172350
BRINIT  004453          C$OPEN= 000034        EXPGOT  004527        G$PRMD= 000002          KIPAR5= 172352
BSEL0 = 000000          C$PNTB= 000014        EXPGT2  004563        G$PRML= 000000          KIPAR6= 172354
BSEL1 = 000001          C$PNTF= 000017        EXPMSG  002320 G      G$RADA= 000140          KIPAR7= 172356
CHKAMB  016034          C$PNTS= 000016        EXPREC  015464 G      G$RADB= 000000          KIPDR0= 172300
CHKMAN  020470 G        C$PNTX= 000015        EXTA    005766        G$RADD= 000040          KIPDR1= 172302
CHKTSS  016326          C$QIO = 000377        EXTEND  005764        G$RADL= 000120          KIPDR2= 172304
CKDROP  017172          C$RDBU= 000007        EXTFEA  002224 G      G$RADO= 000020          KIPDR3= 172306
CKEMAX  017072          C$REFG= 000047        E$END = 002100        G$XFER= 000004          KIPDR4= 172310
CKMSG   011350 G        C$RESE= 000033        E$LOAD= 000035        G$YES = 000010          KIPDR5= 172312
CKMSG2  011470 G        C$REVI= 000003        FATERR= 000060        HIADDR= 001400          KIPDR6= 172314
CKRAM   011104 G        C$RFLA= 000021        FATFLG  002220 G      HOE     = 100000 G      KIPDR7= 172316
CKRAM2  011214 G        C$RPT = 000025        FERCM   011712        HPM1    045674          KTENAB  003120 G
CMDPKT  021250 G        C$SEFG= 000046        FIFEXP  012160 G      HPM2    045730          KTFLG   003126 G
CMPMEM  017650          C$SPRI= 000041        FIF1MS  012232        HPM3    045754          KTINIT  021044
CONFIG  017240          C$SVEC= 000037        FIF2MS  012301        IBE     = 010000 G      KTOFF   017264
COUNT   002306 G        C$TPRI= 000013        FILLME  017412        IDU     = 000040 G      KTON    017246
CSRADD  002204 G        DATA    002310 G      FNOINT  004211        IER     = 020000 G      LEHRMA  002170 G
CTAB    003160 G        DATASC  020222        FORCER  002174 G      IFAULT  004252          LISTAL= 000001
CTABE   003172 G        DEBUGM  011622        FREE    003120 G      INCERK  017034          LOE     = 040000 G
CTABM   003160 G        DEVCNT  002216 G      FREEHI  003124        INTCPC  016140          LOOPCN  002214 G
```

SYMBOL TABLE

```
LOOPCO  013116          L10001  002174          L10073  037026          NXR     003734          PRIO5 = 000240 G
LOOPFL  003156 G        L10002  005762          L10074  037232          NXRERR  005732 G        PRIO6 = 000300 G
LOT   = 000010 G        L10003  012034          L10075  043512          NXRX    003773          PRIO7 = 000340 G
L$ACP   002110 G        L10004  012052          L10076  041012          NXTU    022040          PRMESS  014232
L$APT   002036 G        L10005  012070          L10077  041404          OFL   = 000100          PRMNO   002316 G
L$AU    022370 G        L10006  012076          L10100  042274          ONEFIL= 000000          PRMSGE  014542 G
L$AUT   002070 G        L10007  012114          L10101  042522          O$APTS= 000000          PRMSGO  014722
L$AUTO  022574 G        L10010  012132          L10102  045650          O$AU  = 000001          PRMSG1  014767
L$CCP   002106 G        L10011  012156          L10103  043760          O$BGNR= 000001          PRMSG2  015025
L$CLEA  022654 G        L10012  012230          L10104  044216          O$BGNS= 000001          PROASC  014410
L$CO    002032 G        L10013  012400          L10105  044436          O$DU  = 000001          PR1ASC  014455
L$DEPO  002011 G        L10014  013114          L10106  044662          O$ERRT= 000000          PST32W  003146 G
L$DESC  003404 G        L10015  013742          L10107  045674          O$GNSW= 000001          PUNIT   022322
L$DESP  002076 G        L10016  013764          L10110  046014          O$POIN= 000001          PW.D11= 000021
L$DEVP  002060 G        L10017  015470          MEMADD  013744 G        O$SETU= 000000          PW.D13= 000022
L$DISP  002124 G        L10020  015476          MEMCK   021266 G        PASRPT  022072          PW.D22= 000020
L$DLY   002116 G        L10021  015504          MENASC  020437          PATCH   046154 G        PW.NOP= 000000
L$DTP   002040 G        L10022  015516          MENERR  020364          PATDAT  020220          PW.NO1= 000023
L$DTYP  002034 G        L10023  015540          MENRES  020466          PC.ERA= 002400          PW.RDE= 000024
L$DU    022466 G        L10024  015566          MMVEC = 000250          PC.IER= 002000          PW.RDR= 000001
L$DUT   002072 G        L10025  015726          MSA.FR= 000006          PC.NOO= 001000          PW.RDS= 000005
L$DVTY  003376 G        L10026  016236          MSA.NO= 000000          PC.REL= 000000          PW.RFI= 000003
L$EF    002052 G        L10030  022320          MSA.NR= 000004          PC.REW= 000400          PW.WCT= 000006
L$ENVI  002044 G        L10031  022464          MSA.VO= 000002          PKBCNT= 000006          PW.WFI= 000004
L$ETP   002102 G        L10032  022572          MSGEXP  012134 G        PKHI  = 000004          PW.WFM= 000007
L$EXP1  002046 G        L10033  022652          MSGLOO  013054 G        PKLOW = 000002          PW.WMI= 000010
L$EXP4  002064 G        L10034  022700          MSGSTA  012340 G        PKTADD  007550          PW.WNP= 000011
L$EXP5  002066 G        L10035  023142          MSGSUB  013732 G        PKTFRM  007512          PW.WTR= 000002
L$HARD  045654 G        L10036  023702          MS.ATT= 000006          PKTGET  012054 G        P.ACK = 100000
L$HIME  002120 G        L10037  023550          MS.EXT= 000200          PKTMES  012100 G        P.CMD = 000037
L$HPCP  002016 G        L10040  023632          MS.RSD= 000001          PKTRAM  004741 G        P.CONT= 000012
L$HPTP  002022 G        L10041  024400          MS.RSF= 000020          PKTSSR  012036 G        P.CVC = 040000
L$HW    002154 G        L10042  025072          MS.RST= 000010          PNT   = 001000 G        P.FMT = 000140
L$ICP   002104 G        L10043  026426          M2901   026152          PRAMPK  013766          P.FORM= 000011
L$INIT  021546 G        L10044  025334          M8186   005550          PRASC   014513          P.GETS= 000017
L$LADP  002026 G        L10045  025634          M8189   005641          PRBEXP  015460          P.IE  = 000200
L$LAST  046404 G        L10046  026132          NBA   = 002000          PRBMSG  015326          P.INIT= 000013
L$LOAD  002100 G        L10047  027532          NEWPAS  022026          PRBREC  015462          P.MODE= 007400
L$LUN   002074 G        L10050  027000          NODEV   003110 G        PRBTOT  015413          P.OPP = 020000
L$MREV  002050 G        L10051  027350          NOINIT  004331          PRBYTE  015112 G        P.POSI= 000010
L$NAME  002000 G        L10052  031002          NOINTR  004215          PRI   = 002000 G        P.READ= 000001
L$PRIO  002042 G        L10053  030062          NOITS   002166 G        PRIADD  010154          P.SWB = 010000
L$PROT  021536 G        L10054  030406          NOMAN   020524          PRIAO   010224          P.WRIT= 000005
L$PRT   002112 G        L10055  034370          NOMEM   005454          PRIBXO  007606 G        P.WRTC= 000004
L$REPP  002062 G        L10056  031334          NP.IR = 000200          PRIEQU  010054          P.WRTS= 000006
L$REV   002010 G        L10057  031620          NP.LOO= 000040          PRIPKT  007364 G        QVP     002202 G
L$RPT   022702 G        L10060  032064          NP.OUT= 000100          PRIRAM  010062          RAMASC  014146
L$SOFT  046006 G        L10061  032312          NP.WRP= 000020          PRITAD  010270          RAMDAT  002240 G
L$SPC   002056 G        L10062  032556          NSI     004146          PRITSS  006020          RAMERR  015500 G
L$SPCP  002020 G        L10063  033116          NSINIT  004403          PRITO   010352          RAMEXP  015520 G
L$SPTP  002024 G        L10064  035274          NUL     004523          PRIT1   010415          RAMFOR  010112
L$STA   002030 G        L10065  040410          NULCR   004524          PRIXOR  007736 G        RAMSIZ  002300 G
L$SW    002164 G        L10066  035536          NXM   = 004000          PRIO0 = 000000 G        RAMTAD  015506 G
L$TEST  002114 G        L10067  035776          NXMFLG  003132 G        PRIO1 = 000040 G        RCVHIA  002302 G
L$TIML  002014 G        L10070  036202          NXMHI   003136 G        PRIO2 = 000100 G        RCVLOA  002304 G
L$UNIT  002012 G        L10071  036370          NXMLO   003134 G        PRIO3 = 000140 G        RDERR   005202
L10000  002162          L10072  036574          NXMTST  021442          PRIO4 = 000200 G        RECMSG  002464 G
```

H1

SEQ 0214

SYMBOL TABLE

```
RECV     002232 G    S1.IEO= 010000      TST6ID  030763      T1.1    023504      T4      025074 G
REGSAV   020130      S1.IFM= 001000      TST7ID  034273      T1.2    023564      T4LOOP  025114
RETERR   005366      S1.IHE= 000400      TST8ID  035257      T10     040412 G    T4.1    025074
REWIND   011004 G    S1.IID= 004000      TST9ID  040311      T10BFR  042552      T4.2    025336
RMCHBE= 000167       S1.I1R= 020000      TSV2    002000 G    T10BUF  042614      T4.3    025636
RMCHEN= 000200       S1.I2R= 040000      TSV3    002174 G    T10DAT  042540      T5      026430 G
RMMSGB= 000215       S1.PAR= 100000      TSV4    021536 G    T10DTA  042602      T5ADDR  027470
RMMSGE= 000234       S2.ATI= 000010      TSV5    023464 G    T10INT  043250      T5LOOP  026446
RMPKTB= 000201       S2.BTI= 000004      TSV6    045652 G    T10LOO  040430      T5MEM   027432
RMPKTE= 000210       S2.DIM= 000200      TTIBFR= 177562 G    T10MBF  042634      T5.1    026452
RMR    = 010000      S2.ILW= 000100      TTICSR= 177560 G    T10NBA  042731      T5.2    027020
RWPACK   011100      S2.INR= 000020      TTIVEC= 000060 G    T10NIN  043157      T6      027534 G
SC     = 100000      S2.OUT= 000040      T$ARGC= 000003      T10NNB  043013      T6INT   030553
SCE    = 020000      S2.UND= 000003      T$CODE= 001130      T10PAC  042530      T6LOOP  027552
SCHERR   005274      TBLEND= 003056 G    T$ERRN= 002144      T10PKT  042572      T6NBA   030450
SCME     005007      TCOASC  006472      T$EXCP= 000000      T10RST  043366      T6NINT  030631
SDELAY   010650      TCOCOD  006672      T$FLAG= 000040      T10RT2  043440      T6PACK  030440
SELASC   020432      TEMP1   003112 G    T$GMAN= 000000      T10SSR  043070      T6SSR   030475
SELDAT= 000004       TEMP2   003114 G    T$HILI= 000776      T10.1   040430      T6TSBA  030711
SEL2   = 000002      TERCLS= 000016      T$LAST= 000001      T10.2   041014      T6.1    027552
SETMAP   017306      TESTNO= 000013      T$LOLI= 000000      T10.3   041406      T6.2    030076
SETU     022124      TEXASC  006431      T$LSYM= 010000      T10.4   042276      T7      031004 G
SFFMSG   012072 G    TFCASC  006533      T$LTNO= 000013      T11     043514 G    T7BFR   033166
SFHERR   003701      TIMEXP  015542 G    T$NEST= 177777      T11BFR  044712      T7DATA  033150
SFIERR   003646      TIMSGO  015570      T$NS0  = 000000      T11BF2  044762      T7INT   034121
SFIMSG   012024 G    TINERR  012011      T$NS1  = 000005      T11DAT  044700      T7LOOP  031022
SFPTBL   002164 G    TMPBFR  002630 G    T$NS2  = 000002      T11DTA  044750      T7NBA   033300
SIFLAG   003150 G    TNAM    016674      T$NS3  = 000003      T11LOO  043532      T7NINT  034030
SIMSG    011756      TRANST  002164 G    T$PTNU= 000000      T11NBA  045002      T7PACK  033140
SKIPT    003374      TSBA   = 000000 G    T$SAVL= 177777      T11NIN  045364      T7REST  034322
SOFINI   015764 G    TSBAH  = 000001 G    T$SEGL= 177777      T11PAC  044670      T7SP    033160
SPACE    010462 G    TSBAM2  026242      T$SEKO= 010000      T11PK2  044740      T7SSR   033741
SPM1     046014      TSBAM3  026324      T$SUBN= 000004      T11RES  045544      T7TSBA  034210
SPM4     046044      TSDB   = 000000 G    T$TAGL= 177777      T11RT2  045606      T7.1    031022
SPM6     046074      TSDBH  = 000001 G    T$TAGN= 010111      T11SR2  045310      T7.2    031350
SPM7     046124      TSFCOD  007232      T$TEMP= 000000      T11SSR  045234      T7.3    031622
SR0    = 177572      TSREJ  = 000006      T$TEST= 000013      T11TSB  045442      T7.4    032066
SR1    = 177574      TSSDEF  006602      T$TSTM= 177777      T11.1   043532      T7.5    032314
SR2    = 177576      TSSR   = 000002 G    T$TSTS= 000001      T11.2   043774      T7.6    032560
SR3    = 172516      TSSRBI  003476 G    T$$AU  = 010031      T11.3   044220      T72DAT  033206
SSR    = 000200      TSSRFO  006411      T$$AUT= 010033      T11.4   044440      T72DON= 033222
STATCO   012402      TSSRH  = 000003 G    T$$CLE= 010034      T112RE  045042      T72NBA  033222
SVCGBL= 000000       TSSX    004014      T$$DU  = 010032      T113RE  045123      T72REJ  033353
SVCINS= 000000       TSTBLK  002750 G    T$$HAR= 010107      T114RE  045153      T73REJ  033452
SVCSUB= 000001       TSTCNT  002212 G    T$$HW  = 010000      T2      023704 G    T74REJ  033545
SVCTAG= 000000       TSTEND  016710      T$$INI= 010030      T2LOOP  023722      T75REJ  033643
SVCTST= 000001       TSTFLA  002312 G    T$$MSG= 010025      T2SSR   024300      T8      034372 G
S$LSYM= 010000       TSTLOO  016446 G    T$$PRO= 010027      T2TSBA  024166      T8BFR   035022
S0.IDB= 000010       TSTPTR  002314 G    T$$RPT= 010035      T2TSSR  024233      T8DATA  035010
S0.IFB= 000002       TSTSET  016500 G    T$$SEG= 010000      T23A    003140 G    T8LOOP  034410
S0.IFP= 000001       TST1ID  023662      T$$SOF= 010110      T23B    003142 G    T8NVCK  035077
S0.ILD= 000020       TST10I  043337      T$$SRV= 010026      T3      024402 G    T8PACK  035000
S0.ION= 000040       TST11I  045512      T$$SUB= 010106      T3BFLG  003144 G    T8SSR   035170
S0.IRD= 000100       TST2ID  024352      T$$SW  = 010001      T3LOOP  024420      T8VCK   035042
S0.IRW= 000004       TST3ID  025045      T$$TES= 010102      T3SSR   024774      T9      035276 G
S0.ISP= 000200       TST4ID  026404      T1      023464 G    T3TSBA  024664      T9BFR   037262
S1.ICE= 002000       TST5ID  027402      T1LOOP  023502      T3TSSR  024730      T9DATA  037250
```

SYMBOL TABLE

```
T9INT   040137          UAM   = 000200 G      WRTCHR  010652 G      XSOONL= 000100          X2.EXT= 000200
T9LOOP  035320          UNITN   002200 G      WRTERR  005107        XSOPED= 000010          X2.OPM= 100000
T9NBA   037316          UNREC = 000006        WRTMSG  005052        XSORLL= 010000          X2.RCE= 040000
T9NINT  040046          USI     004117        WSMBK   021260 G      XSORLS= 040000          X2.REV= 000077
T9PACK  037240          WAITF   016240 G      XFERAS  015730        XSOTMK= 100000          X2.SPA= 035400
T9REST  040336          WC.IFA= 000200        XNXM    016366        XSOVCK= 000020          X2.UNI= 000007
T9SSR   037757          WC.IFE= 000002        XORBFO  007670        XSOWLE= 004000          X2.WCF= 002000
T9TSBA  040226          WC.IGO= 000001        XORFOR  010006        XSOWLK= 000004          X3.DCK= 000010
T9.1    035320          WC.IRE= 000010        XST0  = 000006 G      XXCOMM  003116 G        X3.MBZ= 000006
T9.2    035566          WC.IRW= 000004        XST1  = 000010 G      X$ALWA= 000000          X3.MDE= 177400
T9.3    036000          WC.IOT= 000100        XST2  = 000012 G      X$FALS= 000040          X3.OPI= 000100
T9.4    036204          WC.I1T= 000040        XST3  = 000014 G      X$OFFS= 000400          X3.REV= 000040
T9.5    036372          WC.I5R= 000020        XST4  = 000016 G      X$TRUE= 000020          X3.RIB= 000001
T9.6    036576          WF.IED= 000010        XSOBOT  000002        X1.COR= 020000          X3.SPA= 000200
T9.7    037030          WF.IER= 000004        XSOEOT= 000001        X1.DLT= 100000          X3.TRF= 000020
T92DAT  037302          WF.IHI= 000200        XSOIE = 000040        X1.MBZ= 017375          X4.HSP= 100000
T92DON= 037316          WF.IRE= 000040        XSOILA= 000400        X1.RBP= 000400          X4.MBZ= 017400
T92REJ  037371          WF.IWF= 000020        XSOILC= 001000        X1.SPA= 040000          X4.RCE= 040000
T93REJ  037470          WF.IWR= 000100        XSOLET= 020000        X1.UNC= 000002          X4.TSM= 020000
T94REJ  037563          WF.I3R= 000002        XSOMOT= 000200        X2.BUF= 000100          X4.WRC= 000377
T95REJ  037661          WF.I4R= 000001        XSONEF= 002000

. ABS.  046404      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  28880 WORDS  ( 113 PAGES)
DYNAMIC MEMORY:  20060 WORDS  ( 77 PAGES)
ELAPSED TIME:  00:37:00
CNTSAAO.BIC,CNTSAAO.SEQ/-SP=SVC34/ML,TSV1A,TSV22A,TSV3A,TSV4,TSV5A,TSV6
```