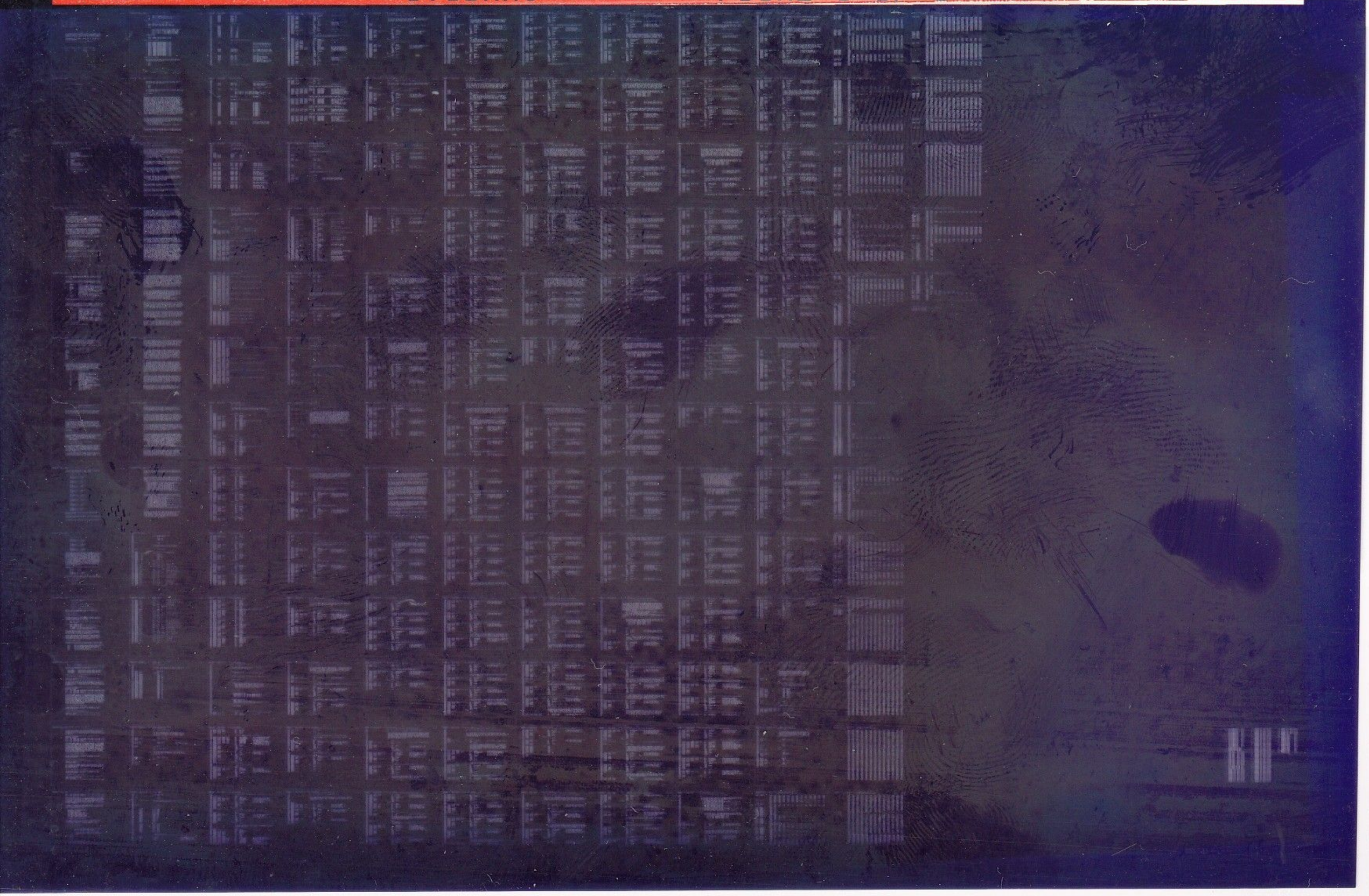
CDS·11 CDS11 MEM SIM DIAG AH-T002A-MC
CVCDAAO FICHE 1 OF 1



.REM &

IDENTIFICATION

PRODUCT CODE: AC-T000A-MC

PRODUCT NAME: CVCDAA0 CDS-11 MEM SIM DIAG

PRODUCT DATE: SEPTEMBER 1981

MAINTAINER: DIAGNOSTIC ENGINEERING

REVISION HISTORY

REVISION	DATE	REASONS
A	SEPTEMBER 1981	FIRST RELEASE

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THE CDS-11 MEMORY SIMULATOR DIAGNOSTIC WILL TEST ALL THE LOGIC ON THE MEMORY SIMULATOR MODULE EXCEPT THAT LOGIC PERTAINING TO THE SYSTEM BUS. THE DIAGNOSTIC WILL TEST ALL BITS IN CONTROL REGISTER 0, CONTROL REGISTER 2, CONTROL REGISTER 4, AND CONTROL REGISTER 6. THE DIAGNOSTIC WILL TEST THE 1K BY 4 MAP PROTECTION RAM'S AND ASSOCIATED LOGIC, MODULE SELECT RAM 0, MODULE SELECT RAM 1, AND THE FOUR 4K BY 16 MEMORY SIMULATOR RAM'S. THE DIAGNOSTIC IS CAPABLE OF TESTING ONE TO EIGHT MEMORY SIMULATOR MODULES SEQUENTIALLY.

IN ORDER TO TEST THE SYSTEM BUS LOGIC, ANOTHER DIAGNOSTIC MUST BE USED. THIS OTHER DIAGNOSTIC, VIA THE STATE ANALYZER MODULE AND THE TARGET EMULATOR MODULE, WILL TEST THE SYSTEM BUS LOGIC ON THE MEMORY SIMULATOR MODULE.

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.

NOTE: THIS PROGRAM HAS NOT BEEN TESTED IN THE APT ENVIRONMENT, HOWEVER, THE APT INTERFACE HAS BEEN PROVIDED IN THE DIAGNOSTIC.

1.2 SYSTEM REQUIREMENTS

1. LSI-11 OR EQUIVALENT TYPE CPU WITH Q-BUS
2. MINIMUM OF 16K WORDS OF MEMORY
3. CONSOLE TERMINAL AND CONTROLLER
4. CDS-11 BACKPLANE AND CABLES
5. MEMORY SIMULATOR MODULE(S) (M8740)
6. MXV11 MODULE AND CDS-11 ROMS
7. STORAGE DEVICE WITH CONTROLLER (OPTIONAL)
8. XXDP+ MEDIA FOR STORAGE DEVICE (OPTIONAL)

1.3 RELATED DOCUMENTS AND STANDARDS

CHQUS? XXDP+ USER'S MANUAL (THE ''?'' IN CHQUS INDICATES THE REVISION LEVEL OF THE DOCUMENT. AT THE TIME THIS PROGRAM WAS WRITTEN, THE REVISION LEVEL WAS ''E''.

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

ALL HARDWARE THAT IS SPECIFIED IN SECTION 1.2 OF THIS DOCUMENT MUST BE OPERATIONAL AND FREE OF ALL FAULTS.

1.5 ASSUMPTIONS

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

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

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

2.2 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. YOJ 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
IBE*	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

EVL EXECUTE EVALUATION (ON DIAGNOSTICS WHICH HAVE EVALUATION SUPPORT)

*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 6 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). YOU WILL THEN BE ASKED THE FOLLOWING QUESTIONS FOR EACH UNIT.

CSR ADDRESS:
DEVICE NUMBER:

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).

THERE ARE NO SOFTWARE QUESTIONS IN THIS PROGRAM.

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 FICTIONAL DEVICE, THE XY11. SUPPOSE THIS DEVICE CONSISTS OF A CONTROL MODULE WITH EIGHT UNITS (SUB-DEVICES) ATTACHED TO IT. THESE UNITS ARE DESCRIBED BY THE OCTAL NUMBERS 0 THROUGH 7. THERE

IS ONE HARDWARE PARAMETER THAT CAN VARY AMONG UNITS CALLED THE Q-FACTOR. THIS Q-FACTOR MAY BE 0 OR 1. BELOW IS A SIMPLE WAY TO BUILD A TABLE FOR ONE XY11 WITH EIGHT UNITS.

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

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

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

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

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

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

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

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

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

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

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

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

```
FEATURE.

        # UNITS (D) ?  8<CR>

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

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

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

AS YOU CAN SEE IN THE ABOVE DIALOGUE, THE RUNTIME SERVICES WILL BUILD AS MANY ENTRIES AS IT CAN WITH THE INFORMATION GIVEN IN ANY ONE PASS THROUGH THE QUESTIONS. IN THE FIRST PASS, TWO ENTRIES ARE BUILT SINCE TWO SUB-DEVICES AND Q-FACTORS WERE SPECIFIED. THE SERVICES ASSUME THAT THE CSR ADDRESS IS 160000 FOR BOTH SINCE IT WAS SPECIFIED ONLY ONCE. IN THE SECOND PASS, FOUR ENTRIES WERE BUILT. THIS IS BECAUSE FOUR SUB-DEVICES WERE SPECIFIED. THE "-" CONSTRUCT TELLS THE RUNTIME SERVICES TO INCREMENT THE DATA FROM THE FIRST NUMBER TO THE SECOND. IN THIS CASE, SUB-DEVICES 2, 3, 4 AND 5 WERE SPECIFIED. (IF THE SUB-DEVICE WERE SPECIFIED BY ADDRESSES, THE INCREMENT WOULD BE BY 2 SINCE ADDRESSES MUST BE ON AN EVEN BOUNDARY.) THE CSR ADDRESSES AND Q-FACTORS FOR THE FOUR ENTRIES ARE ASSUMED TO BE 160000 AND 0 RESPECTIVELY SINCE THEY WERE ONLY SPECIFIED ONCE. THE LAST TWO UNITS ARE SPECIFIED IN THE THIRD PASS.

THE WHOLE PROCESS COULD HAVE BEEN ACCOMPLISHED IN ONE PASS AS SHOWN BELOW.

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

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

AS YOU CAN SEE FROM THIS EXAMPLE, NULL REPLIES (COMMAS ENCLOSING A NULL FIELD) TELL THE RUNTIME SERVICES TO REPEAT THE LAST REPLY.

2.7 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

1. BOOT XXDP+

2. ANSWER ANY QUESTIONS ASKED AND GIVE THE DATE.

3. TYPE "R NAME", WHERE NAME IS THE NAME OF THE BIN OR BIC FILE FOR THIS PROGRAM

4. TYPE "START"

5. ANSWER THE "CHANGE HW" QUESTION WITH "Y"

6. ANSWER ALL THE HARDWARE QUESTIONS

7. 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 "IBE" 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", "IBE" 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

THE ERROR PRINTOUTS WILL USE THE FOLLOWING WORDS TO INDICATE ERROR INFORMATION. A DESCRIPTION OF THE WORDS PRINTED OUT ARE AS FOLLOWS:

REG:	ONE OF THE MEMORY SIMULATOR MODULE'S CONTROL REGISTERS
LOAD:	DATA THAT WAS LOADED INTO THE CONTROL REGISTER
READ:	DATA THAT WAS READ FROM THE CONTROL REGISTER
MASK:	BITS IN THE CONTROL REGISTER THAT ARE NOT CHECKED
GOOD:	EXPECTED CONTROL REGISTER DATA
BAD:	DATA "READ" FROM THE CONTROL REGISTER WITH THE "MASK"

BITS CLEARED
XXXXXX: SIX OCTAL DIGITS INDICA ING THE DA A FOR THE ABOVE WORDS

THERE ARE FOUR ERROR NUMBERS ASSOCIATED WITH THIS DIAGNOSTIC. THE ERROR NUMBERS AND THEIR MEANINGS ARE DESCRIBED BELOW:

ERROR NUMBER 1 - ERROR DETECTED CHECKING CONTROL REGISTER 0
ERROR NUMBER 2 - ERROR DETECTED CHECKING CONTROL REGISTER 2
ERROR NUMBER 3 - ERROR DETECTED CHECKING CONTROL REGISTER 4
ERROR NUMBER 4 - ERROR DETECTED CHECKING CONTROL REGISTER 6

EXAMPLES OF EACH TYPE OF CONTROL REGISTER ERROR PRINTOUT ARE SHOWN BELOW:

** CONTROL REGISTER 0 ERROR MESSAGES **

```
CVCDA DVC FTL ERR 00001 ON UNIT 00 TST 001 SUB 000 PC: XXXXXX
CONTROL REG 0 ERROR
LOAD: XXXXXX READ: XXXXXX MASK: XXXXXX GOOD: XXXXXX BAD: XXXXXX
```

THE ABOVE ERROR PRINTOUT IS GIVEN WHEN THE PROGRAM IS CHECKING CONTROL REGISTER 0 AND NO OTHER CONTROL REGISTER INFORMATION IS NEEDED TO DETERMINE THE FAULT. IF MORE CONTROL REGISTER INFORMATION IS NEEDED IN DETERMINING THE FAULT OF CONTROL REGISTER 0, THE FOLLOWING ERROR REPORT WILL BE GIVEN.

```
CVCDA DVC FTL ERR 00001 ON UNIT 00 TST 014 SUB 000 PC: XXXXXX
MAP PROTECT LOGIC ERROR
CONTROL REG 0 ERROR
REG 0 = LOAD: XXXXXX READ: XXXXXX MASK: XXXXXX GOOD: XXXXXX BAD: XXXXXX
CONTROL REG 2 ERROR
REG 2 = LOAD: XXXXXX READ: XXXXXX MASK: XXXXXX GOOD: XXXXXX BAD: XXXXXX
REG 4 = LOAD: XXXXXX READ: XXXXXX
REG 6 = LOAD: XXXXXX READ: XXXXXX MASK: XXXXXX GOOD: XXXXXX BAD: XXXXXX
```

IN THE ABOVE ERROR, REFER TO THE LINE INDICATING "REG 0 =" FOR CONTROL REGISTER 0 ERROR INFORMATION. THE REMAINING CONTROL REGISTER INFORMATION IS GIVEN TO INDICATE WHAT WAS LOADED INTO THOSE CONTROL REGISTERS PREVIOUS TO THE ERROR. THIS IS DONE TO AID THE USER IN DETERMINING THE FAULT ON ERRORS WHICH NEED PREVIOUS CONTROL REGISTER SETUP. THE "BAD" INFORMATION FOR CONTROL REGISTERS OTHER THEN CONTROL REGISTER 0 SHOULD EQUAL THE "GOOD" INFORMATION. THE "BAD" INFORMATION IS PROVIDED IN CASE THE USER PROCEEDS FROM AN ERROR, IN WHICH CASE, THE "BAD" INFORMATION MAY AID THE USER IF ANOTHER CONTROL REGISTER HAD FAILED. THE ERROR MESSAGE "MAP PROTECTION LOGIC ERROR" INDICATES THE AREA OF LOGIC BEING TESTED WHEN THE ERROR WAS DETECTED.

```
TIME OUT ERROR ADDRESSING CONTROL REG 0
```

THE ABOVE ERROR PRINTOUT IS GIVEN WHEN THE PROGRAM IS TRYING TO ADDRESS CONTROL REGISTER 0 AND CAN'T. THE PROGRAM THEN JUMPS TO TIME OUT VECTOR #4.

** CONTROL REGISTER 2 ERROR MESSAGES **

```
CDCDA DVC FTL ERR 00002 ON UNIT 00 TST 004 SUB 000 PC: XXXXXX
CONTROL REG 2 ERROR
LOAD: XXXXXX READ: XXXXXX MASK: XXXXXX GOOD: XXXXXX BAD: XXXXXX
```

THE ABOVE ERROR PRINTOUT IS GIVEN WHEN THE PROGRAM IS CHECKING CONTROL REGISTER 2 AND NO OTHER CONTROL REGISTER INFORMATION IS NEEDED TO DETERMINE THE FAULT. IF MORE CONTROL REGISTER INFORMATION IS NEEDED IN DETERMINING THE FAULT OF CONTROL REGISTER 2, THE FOLLOWING ERROR REPORT WILL BE GIVEN.

```
CVCDA DVC FTL ERR 00002 ON UNIT 00 TST 014 SUB 000 PC: XXXXXX
MAP PROTECT LOGIC ERROR
CONTROL REG 2 ERROR
REG 0 = LOAD: XXXXXX READ: XXXXXX MASK: XXXXXX GOOD: XXXXXX BAD: XXXXXX
REG 2 = LOAD: XXXXXX READ: XXXXXX MASK: XXXXXX GOOD: XXXXXX BAD: XXXXXX
REG 4 = LOAD: XXXXXX READ: XXXXXX
REG 6 = LOAD: XXXXXX READ: XXXXXX MASK: XXXXXX GOOD: XXXXXX BAD: XXXXXX
```

IN THE ABOVE ERROR, REFER TO THE LINE INDICATING "REG 2 =" FOR CONTROL REGISTER 2 ERROR INFORMATION. THE REMAINING CONTROL REGISTER INFORMATION IS GIVEN TO INDICATE WHAT WAS LOADED INTO THOSE CONTROL REGISTERS PREVIOUS TO THE ERROR. THIS IS DONE TO AID THE USER IN DETERMINING THE FAULT ON ERRORS WHICH NEED PREVIOUS CONTROL REGISTER SETUP. THE "BAD" INFORMATION FOR CONTROL REGISTERS OTHER THEN CONTROL REGISTER 2 SHOULD EQUAL THE "GOOD" INFORMATION. THE "BAD" INFORMATION IS PROVIDED IN CASE THE USER PROCEEDS FROM AN ERROR, IN WHICH CASE, THE "BAD" INFORMATION MAY AID THE USER IF ANOTHER CONTROL REGISTER HAD FAILED. THE ERROR MESSAGE "MAP PROTECTION LOGIC ERROR" INDICATES THE AREA OF LOGIC BEING TESTED WHEN THE ERROR WAS DETECTED.

```
TIME OUT ERROR ADDRESSING CONTROL REG 2
```

THE ABOVE ERROR PRINTOUT IS GIVEN WHEN THE PROGRAM IS TRYING TO ADDRESS CONTROL REGISTER 2 AND CAN'T. THE PROGRAM THEN JUMPS TO TIME OUT VECTOR #4.

```
** CONTROL REGISTER 4 ERROR MESSAGE **

CVCDA DVC FTL ERR 00003 ON UNIT 00 TST 006 SUB 000 PC: XXXXXX
CONTROL REG 4 ERROR
LOAD: XXXXXX READ: XXXXXX
```

ALL SIXTEEN BITS OF CONTROL REGISTER 4 ARE READ/WRITE BITS, THEREFORE, ONLY TWO WORDS ARE NEEDED TO REPORT AN ERROR. THESE WORDS INDICATE THE DATA THAT WAS LOADED INTO CONTROL REGISTER 4 AND DATA THAT WAS READ FROM CONTROL REGISTER 4.

```
TIME OUT ERROR ADDRESSING CONTROL REG 4
```

THE ABOVE ERROR PRINTOUT IS GIVEN WHEN THE PROGRAM IS TRYING TO ADDRESS CONTROL REGISTER 4 AND CAN'T. THE PROGRAM THEN JUMPS TO TIME OUT VECTOR #4.

```
** CONTROL REGISTER 6 ERROR MESSAGE **

CVCDA DVC FTL ERR 00004 ON UNIT 00 TST 011 SUB 000 PC: XXXXXX
ERROR TYPE MESSAGE (SEE BELOW)
CONTROL REG 6 ERROR
REG 0 = LOAD: XXXXXX READ: XXXXXX MASK: XXXXXX GOOD: XXXXXX BAD: XXXXXX
```

REG 2 = LOAD: XXXXXX READ: XXXXXX MASK: XXXXXX GOOD: XXXXXX BAD: XXXXXX
REG 4 = LOAD: XXXXXX READ: XXXXXX
REG 6 = LOAD: XXXXXX READ: XXXXXX MASK: XXXXXX GOOD: XXXXXX BAD: XXXXXX

IN THE ABOVE ERROR, REFER TO THE LINE INDICATING "REG 6 =" FOR CONTROL REGISTER 6 ERROR INFORMATION. THE REMAINING CONTROL REGISTER INFORMATION IS GIVEN TO INDICATE WAS WAS LOADED INTO THOSE REGISTERS PREVIOUS TO THE ERROR. THIS IS DONE TO AID THE USER IN DETERMINING THE FAULT ON ERRORS WHICH NEED PREVIOUS CONTROL REGISTER SETUP. THE "BAD" INFORMATION FOR CONTROL REGISTERS OTHER THEN CONTROL REGISTER 6 SHOULD EQUAL THE "GOOD" INFORMATION.

THE ERROR TYPE MESSAGE IN THE ABOVE ERROR REPORT WILL BE ONE OF THOSE LISTED BELOW. THESE MESSAGES ARE REPORTED TO HELP THE USER IDENTIFY THE AREA OF LOGIC BEING TESTED IN WHICH THE ERROR WAS DETECTED. THESE ERROR TYPE MESSAGES ARE AS FOLLOWS:

DATA ERROR IN MAP PROTECTION RAM
ADDRESS SHORT IN MAP PROTECTION RAM
MAP PROTECT LOGIC ERROR
DATA ERROR IN MODULE SELECT RAM 0
DATA ERROR IN MODULE SELECT RAM 1
CHIP ENABLE ERROR - MODULE SELECT RAMS
MODULE SELECT RAM ADDRESSING ERROR
DATA ERROR IN MEMORY SIMULATOR RAM
CHIP ENABLE ERROR - MEMORY SIMULATOR RAM

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.

5.0 DEVICE INFORMATION TABLES

CONTROL REGISTER 0 (163010)

15	ID H	BIT 15 = 1 READ DEVICE TYPE IN BITS 15-8. MEMORY SIMULATOR DEVICE TYPE EQUALS 1 (400) BIT 15 = 0 READ DEVICE NUMBER INTO BITS 11-8. BITS 11-8 ARE USED TO SELECT THE DEVICE NUMBER OF THE MEMORY SIMULATOR. THESE BITS MUST BE EQUAL TO THE SETTING OF SWITCHES DEV 3, DEV 2, DEV 1 AND DEV 0.
11	SIG11 H	
10	SIG10 H	
9	SIG9 H	
8	SIG8 H	
7	UNUSED	
6	CK H	CLOCK HIGH (R/W)
5	WRV H	WRITE VIOLATION (READ ONLY)
4	RDV H	READ VIOLATION (READ ONLY)
3	8BIT H	BIT 3 = 1 8 BIT MODE. BIT 3 = 0 16 BIT MODE (R/W)

```
          2  MP H    MAP PROTECT SELECT (R/W)
          1  CTS H   BIT 1 = 1 MEMORY ACCESS FROM SYSTEM BUS. BIT 1 = 0
                     MEMORY ACCESS FROM LSI-11 BUS (R/W)
          0  RST H   BIT 0 = 1 RESET READ/WRITE VIOLATION F/F'S (R/W)

CONTROL REGISTER 2 (163012)

        15-8         BITS 15-8 ARE NOT AVAILABLE IN CONTROL REGISTER 2

          7  MSBRK H MEMORY SIMULATOR BREAK (READ ONLY)
          6  WREN H  MAP PROTECT RAM SIGNAL WRE H (READ ONLY)
          5  ESR H   MAP PROTECT RAM SIGNAL MPIN H (READ ONLY)

          4  UNUSED

                     BITS 3 AND 2 ARE USED TO SELECT THE MEMORY SIMULATOR
                     RAM, MAP PROTECT RAM AND THE MODULE SELECT RAMS
                     BIT 3 = 0 BIT 2 = 0 SELECT MEMORY SIMULATOR RAM
                     BIT 3 = 0 BIT 2 = 1 SELECT MODULE SELECT MEMORY 0
                     BIT 3 = 1 BIT 2 = 0 SELECT MAP PROTECT MEMORY
                     BIT 3 = 1 BIT 2 = 1 SELECT MODULE SELECT MEMORY 1
          3  MSEL1 H (R/W)
          2  MSEL0 H (R/W)

          1  MSAD17 H MEMORY SIMULATOR ADDRESS 17 (R/W)
          0  MSAD16 H MEMORY SIMULATOR ADDRESS 16 (R/W)

CONTROL REGISTER 4 (163014)

         15  MSAD15 H MEMORY SIMULATOR ADDRESS 15 (R/W)
         14  MSAD14 H MEMORY SIMULATOR ADDRESS 14 (R/W)
         13  MSAD13 H MEMORY SIMULATOR ADDRESS 13 (R/W)
         12  MSAD12 H MEMORY SIMULATOR ADDRESS 12 (R/W)
         11  MSAD11 H MEMORY SIMULATOR ADDRESS 11 (R/W)
         10  MSAD10 H MEMORY SIMULATOR ADDRESS 10 (R/W)
          9  MSAD9 H  MEMORY SIMULATOR ADDRESS  9 (R/W)
          8  MSAD8 H  MEMORY SIMULATOR ADDRESS  8 (R/W)
          7  MSAD7 H  MEMORY SIMULATOR ADDRESS  7 (R/W)
          6  MSAD6 H  MEMORY SIMULATOR ADDRESS  6 (R/W)
          5  MSAD5 H  MEMORY SIMULATOR ADDRESS  5 (R/W)
          4  MSAD4 H  MEMORY SIMULATOR ADDRESS  4 (R/W)
          3  MSAD3 H  MEMORY SIMULATOR ADDRESS  3 (R/W)
          2  MSAD2 H  MEMORY SIMULATOR ADDRESS  2 (R/W)
          1  MSAD1 H  MEMORY SIMULATOR ADDRESS  1 (R/W)
          0  MSAD0 H  MEMORY SIMULATOR ADDRESS  0 (R/W)

CONTROL REGISTER 6 (163016)

        WHEN THE MAP PROTECTION RAM IS SELECTED VIA CONTROL REGISTER 2,
        THE FOLLOWING BITS ARE LOADED INTO OR READ FROM THE MAP PROTEC-
        TION RAMS VIA CONTROL REGISTER 6.

          3   MUTB H
          2   RDE H
          1   WRE H
          0   MPIN H
```

WHEN MODULE SELECT RAM 0 OR 1 IS SELECTED VIA CONTROL REGISTER 2, THE FOLLOWING BITS ARE LOADED INTO OR READ FROM MODULE SELECT RAMS 0 OR 1 VIA CONTROL REGISTER 6.

3 EN3 H
2 EN2 H
1 EN1 H
0 EN0 H

WHEN THE MEMORY SIMULATOR RAMS ARE SELECTED VIA CONTROL REGISTER 2, ALL 16 BITS ARE LOADED INTO AND READ FROM THE SIMULATOR MEMORY RAMS VIA CONTROL REGISTER 6.

6.0 TEST SUMMARIES

TEST 1:

THIS TEST WILL CHECK THAT THE MEMORY SIMULATOR CAN BE SELECTED AND INITIALIZED TO A KNOWN STATE. THIS TEST WILL BE EXECUTED AT THE BEGINNING OF EVERY TEST THAT FOLLOWS. THE TEST WILL LOAD THE DEVICE NUMBER INTO CONTROL REGISTER 0 AND CHECK THAT THE DEVICE NUMBER CAN BE READ BACK CORRECTLY. THE R/W BITS IN THE LOW BYTE ARE CHECKED TO BE CLEARED. THE READ ONLY BITS, WRV H AND RDV H, ARE NOT CHECKED. THE TEST WILL LOAD THE DEVICE NUMBER AND THE SIGNAL I/D H INTO CONTROL REGISTER 0 AND CHECK THAT THE DEVICE TYPE AND THE R/W BITS IN THE LOW BYTE CAN BE READ BACK CORRECTLY. THE TEST WILL THEN LOAD THE DEVICE NUMBER AND THE SIGNAL RST H (RST H WILL CLEAR RDV AND WRV FLIP-FLOPS) INTO CONTROL REGISTER 0 AND CHECK THAT THE DEVICE NUMBER AND THE SIGNAL RST H ARE SET IN CONTROL REGISTER 0. THE OTHER R/W BITS AND THE READ ONLY BITS (RDV H AND WRV H) ARE CHECKED TO BE CLEARED IN CONTROL REGISTER 0. THE LAST PART OF REGISTER 0 TEST WILL BE TO CLEAR THE SIGNAL RST H IN CONTROL REGISTER 0 AND CHECK THAT THE SIGNAL RST H WAS CLEARED IN CONTROL REGISTER 0. THE OTHER BITS ARE CHECKED NOT TO HAVE CHANGED. THE TEST WILL CLEAR THE READ/WRITE BITS MSEL1 H, MSEL2 H, MSAD17 H, AND MSAD16 H IN CONTROL REGISTER 2 AND CHECK THAT THESE BITS ARE CLEARED BY READING CONTROL REGISTER 2. THE TEST WILL ALSO CLEAR ALL MSAD BITS IN CONTROL REGISTER 4 AND CHECK THAT ALL THE BITS WERE CLEARED.

TEST 2:

THIS TEST WILL CHECK CONTROL REGISTER 0 WITH BASIC PATTERNS. THE TEST WILL CHECK THAT THE R/W BITS RST H, CTS H, MP H, 8 BIT H AND CK H CAN BE SET AND CLEARED. THE READ ONLY BITS RDV H AND WRV H WILL BE CHECKED TO BE ZERO WHEN THE SIGNAL RST H IS SET TO A ONE. THE FOLLOWING TEST PATTERNS WILL BE USED TO CHECK CONTROL REGISTER 0:

1. SET ALL R/W BITS TO A ONE
2. SET ALL R/W BITS TO A ZERO
3. SET RST H, MP H AND CK H TO EQL 1. SET CTS H + 8 BIT H EQL 0.
4. SET CTS H + 8 BIT H EQL 1. SET RST H, MP H, + CK H EQL 0.

TEST 3:

THIS TEST WILL CHECK THE READ/WRITE BITS IN CONTROL REGISTER 0 USING A BINARY COUNT PATTERN. THE READ ONLY BITS WILL NOT BE CHECKED WHEN THE SIGNAL RST H IS A ZERO, HOWEVER, WHEN RST H IS SET TO A ONE, THE

READ ONLY BITS WILL BE CHECKED TO BE ZERO.

TEST 4:

THIS TEST WILL CHECK THE READ/WRITE BITS IN CONTROL REGISTER 2 USING BASIC PATTERNS. THIS TEST WILL CHECK THAT THE SIGNALS MSAD16 H, MSAD17 H MSEL0 H, AND MSEL1 H CAN BE SET AND CLEARED IN CONTROL REGISTER 2. THE READ ONLY BITS ESR H, WREN H, AND MSBRK H ARE IGNORED DURING THIS TEST ALONG WITH THE UNDEFINED BITS IN CONTROL REGISTER 2. THE TEST PATTERNS USED DURING THIS TEST ARE AS FOLLOWS:

1. SET ALL R/W BITS TO A ONE
2. SET ALL R/W BITS TO A ZERO
3. SET MSAD16 H + MSEL0 H EQL 1. SET MSAD17 H + MSEL1 H EQL 0.
4. SET MSAD17 H + MSEL1 H EQL 1. SET MSAD16 H + MSEL0 H EQL 0.

TEST 5:

THIS TEST WILL CHECK THE READ/WRITE BITS IN CONTROL REGISTER 2 USING A BINARY COUNT PATTERN. THE BITS BEING TESTED ARE MSAD16 H, MSAD17 H, MSEL0 H AND MSEL1 H. THE READ ONLY BITS, ESR H, WREN H, MSBRK H, AND THE UNDEFINED BITS, ARE IGNORED DURING THIS TEST.

TEST 6:

THIS TEST WILL CHECK ALL BITHE R/W BITS IN CONTROL REGISTER 4 USING BASIC PATTERNS. ALL SIXTEEN BITS IN CONTROL REGISTER 4 ARE READ/WRITE BITS. THESE BITS ARE CALLED MSAD15 H TO MSAD0 H. THE TEST PATTERNS USED TO TEST CONTROL REGISTER 4 ARE AS FOLLOWS:

1. SET ALL BITS TO A ONE.
2. SET ALL BITS TO A ZERO

TEST 7:

THIS TEST WILL CHECK ALL THE R/W BITS IN CONTROL REGISTER 4 USING AN ALTERNATING ONES AND ZEROES PATTERN AND AN ALTERNATING ZEROES AND ONES PATTERN. THIS TEST CHECKS THAT NO ADJACENT BITS ARE SHORTED TO EACH OTHER. THE TEST PATTERNS USED ARE AS FOLLOWS:

1. LOAD CONTROL REGISTER 4 WITH 125252
2. LOAD CONTROL REGISTER 4 WITH 052525

TEST 8:

THIS TEST WILL CHECK THE LOW BYTE OF CONTROL REGISTER 4 USING A BINARY COUNT PATTERN. THE HIGH BYTE OF CONTROL REGISTER 4 WILL BE CHECKED TO BE CLEARED DURING THIS TEST.

TEST 9:

THIS TEST WILL CHECK THE HIGH BYTE OF CONTROL REGISTER 4 USING A BINARY COUNT PATTERN. THE LOW BYTE OF CONTROL REGISTER 4 WILL BE CHECKED TO BE CLEARED DURING THIS TEST.

TEST 10:

THIS TEST WILL CHECK THAT THE SIGNAL MSBRK H IN CONTROL REGISTER 2 CAN BE SET TO A 0 WHEN THE SIGNALS RST H AND MP H ARE SET TO A ONE

IN CONTROL REGISTER 0. SETTING THE SIGNAL RST H WILL PRESET THE RDV AND WRV FLIP-FLOP'S, SUCH THAT, THE SIGNAL BRK L WILL BE HIGH. THE SIGNAL MP H WILL ALLOW THE SIGNAL BRK L AS A HIGH TO BE INVERTED, GENERATING THE SIGNAL MSBRK H AS A LOW. THE SIGNAL MSBRK H AS A LOW WILL BE READ INTO CONTROL REGISTER 2 AS A ZERO. THE TEST WILL THEN CLEAR THE SIGNAL RST H IN CONTROL REGISTER 0 AND CHECK THE SIGNAL MSBRK H IN CONTROL REGISTER 2 TO REMAIN A ZERO.

TEST 11:

THIS TEST WILL CHECK EACH ADDRESS OF THE 1K BY 4 MAP PROTECTION RAM WITH A DATA PATTERN OF ALL ONES AND THEN ALL ZEROES. THE TEST WILL SELECT AND INITIALIZE THE MEMORY SIMULATOR MODULE AS DESCRIBED IN TEST 1. THE TEST WILL SET THE SIGNAL MSEL1 H IN CONTROL REGISTER 2 WHICH WILL CAUSE THE SIGNAL SMPM L TO BE ASSERTED ON A WRITE OR READ TO CONTROL REGISTER 6. THE TEST WILL THEN SELECT THE MEMORY SIMULATOR ADDRESS BY LOADING THE ADDRESS INTO CONTROL REGISTER 4 BITS 15-8 AND CONTROL REGISTER 2 BITS 1 AND 0. THE TEST WILL THEN WRITE ALL ONES INTO THE MAP PROTECTION RAM VIA CONTROL REGISTER 6 AND THEN READ THE RAM LOCATION BACK VIA CONTROL REGISTER 6. THE RAM BITS MUTB H, MPIN H, WRE H, AND RDE H WILL BE CHECKED FOR ALL ONES. THE TEST WILL THEN WRITE THE LOCATION ADDRESSED BY CONTROL REGISTER 2 AND 4 WITH A DATA PATTERN OF ALL ZEORES. THE TEST WILL READ THE LOCATION AND CHECK THAT ALL THE RAM BITS WERE READ BACK AS ZEROES. THE TEST WILL THEN INCREMENT THE ADDRESS TO THE NEXT LOCATION AND REPEAT THE TEST SEQUENCE UNTILL ALL ADDRESSES OF THE 1K BY 4 MAP PROTECTION RAM HAVE BEEN TESTED WITH ALL ONES AND ZEROES.

TEST 12:

THIS TEST WILL CHECK THE 1K BY 4 MAP PROTECTION RAM USING AN ALTERNATING ONES AND ZEROES PATTERN AND AN ALTERNATING ZEROES AND ONES PATTERN. THE TEST WILL ALSO CHECK THAT THE MAP PROTECT BITS MPIN H AND WRE H CAN BE READ BACK INTO CONTROL REGISTER 2 AS SIGNALS ESR H AND WREN H RESPECTIVELY. THE SIGNAL MSBRK H IN CONTROL REGISTER 2 WILL BE CHECKED TO BE 0 DURING THIS TEST.

THE TEST WILL SET THE SIGNALS RST H AND MP H IN CONTROL REGISTER 0. THE TEST WILL THEN READ CONTROL REGISTER 0 AND CHECK THAT THE SIGNALS CTS H, 8 BIT H, RDV H, AND WRV H ARE ZERO AND THAT RST H AND MP H ARE ONES. THE TEST WILL THEN CLEAR THE SIGNAL RST H AND CHECK THAT RST H WAS THE ONLY BIT THAT CLEARED IN CONTROL REGISTER 0. THE TEST WILL THEN SET THE SIGNAL MSEL1 H TO A ONE IN CONTROL REGISTER 2. CONTROL REGISTER 2 BITS MSAD17 H AND MSAD16 H WILL BE SET TO A ONE OR ZERO DEPENDING UPON THE ADDRESS TO BE TESTED. THE TEST WILL THEN READ CONTROL REGISTER 2 CHECKING THAT THE BITS WERE LOADED CORRECTLY AND THAT THE SIGNAL MSBRK H IS A ZERO. MSBRK H IS READ BACK AS A RESULT OF THE SIGNAL MP H BEING ASSERTED IN CONTROL REGISTER 0. THE SIGNAL MSEL1 H BEING SET AND THE SIGNAL MSEL0 H BEING CLEARED WILL CAUSE THE SIGNAL SMPM L TO BE ASSERTED ON A WRITE OR READ TO CONTROL REISTER 6. THE SIGNAL SMPM L BEING ASSERTED WILL ENABLE THE 1K BY 4 MAP PROTECTION RAMS TO BE WRITTEN OR READ. THE TEST WILL NOW LOAD THE REMAINING PART OF THE 18 BIT MEMORY SIMULATOR ADDRESS INTO CONTROL REGISTER 4 BITS 15-8 AND CHECK THAT CONTROL REGISTER 4 WAS LOADED CORRECTLY. THE TEST WILL NOW WRITE THE LOCATION SELECTED WITH ONES IN BITS MPIN H AND RDE H AND ZEROES IN BITS WRE H AND MUTB H. THIS IS DONE

VIA WRITING TO CONTROL REGISTER 6 WHICH WILL ASSERT THE SIGNAL SMPM L. THE SIGNAL SMPM L BEING ASSERTED WILL ENABLE THE MAP PROTECTION RAM TO BE WRITTEN OR READ. THE PROGRAM WILL READ BACK THE LOCATION WRITTEN VIA CONTROL REGISTER 6 AND CHECK THE DATA TO BE CORRECT. THE TEST WILL THEN READ CONTROL REGISTER 2 AND CHECK THAT BIT ESR H IS SET TO A ONE AND BIT WREN H IS SET TO A 0. THESE BITS ARE ENABLED INTO CONTROL REGISTER 2 VIA THE SIGNAL MP H IN CONTROL REGISTER 0. THE TEST WILL NOW WRITE THE MAP PROTECTION RAM BITS WRE H AND MUTB H WITH ONES AND BITS MPIN H AND RDE H WITH ZEROES. THE TEST WILL READ THE RAM LOCATION VIA CONTROL REGISTER 6 AND CHECK THAT THE CORRECT PATTERN WAS READ BACK. THE TEST WILL THEN READ CONTROL REGISTER 2 AND CHECK THAT THE BIT WREN H IS SET TO A ONE AND BIT ESR H IS SET TO A ZERO. THIS TEST SEQUENCE WILL BE REPEATED FOR EACH ADDRESS OF THE 1K BY 4 MAP PROTECTION RAM.

TEST 13:

THIS TEST WILL CHECK THE MAP PROTECTION RAM FOR ADDRESS SHORTS. THIS TEST WILL CHECK THAT WRITING A LOCATION IN THE MAP PROTECTION RAM WILL NOT WRITE ANOTHER LOCATION IN THE RAM. THE TEST WILL FILL THE MAP PROTECTION RAM WITH ALL ZEROES CHECKING EACH LOCATION TO BE ZERO AS IT IS WRITTEN. THE TEST WILL THEN RESET THE ADDRESS POINTER TO THE FIRST LOCATION OF THE RAM AND DO THE FOLLOWING STEPS:

1. READ THE LOCATION AND CHECK IT TO BE ZERO
2. WRITE THE LOCATION WITH ALL ONES AND CHECK IT TO BE ALL ONES
3. INCREMENT ADDRESS POINTER TO NEXT ADDRESS OF RAM
4. REPEAT STEPS 1-3 UNTIL ALL LOCATIONS HAVE BEEN CHECKED

TEST 14:

THIS TEST WILL CHECK THAT THE WRV FLIP-FLOP CAN BE SET VIA THE SIGNALS WRE H AND CK H. THE TEST WILL CHECK THAT THE WRV FLIP-FLOP, ONCE SET, CAN NOT BE CLOCKED TO A ZERO BY CHANGING THE STATE OF WRE H AND CLOCKING THE SIGNAL CK H AGAIN. THE TEST WILL CHECK THAT THE WRV FLIP-FLOP CAN BE CLEARED WHEN THE SIGNAL RST H IS PULSED. THE TEST WILL ALSO CHECK THAT THE SIGNAL MSBRK H CAN BE SET TO A ONE AND ZERO AS A RESULT OF THE WRV FLIP-FLOP BEING SET AND CLEARED.

TEST 15:

THIS TEST WILL CHECK THAT THE RDV FLIP-FLOP CAN BE SET VIA THE SIGNALS RDE H AND CK H. THE TEST WILL CHECK THAT THE RDV FLIP-FLOP, ONCE SET, CAN NOT BE CLOCKED TO A ZERO BY CHANGING THE STATE OF RDE H AND CLOCKING THE SIGNAL CK H AGAIN. THE TEST WILL CHECK THAT THE RDV FLIP-FLOP CAN BE CLEARED WHEN THE SIGNAL RST H IS PULSED. THE TEST WILL ALSO CHECK THAT THE SIGNAL MSBRK H CAN BE SET TO A ONE AND ZERO AS A RESULT OF THE RDV FLIP-FLOP BEING SET AND CLEARED.

TEST 16:

THIS TEST WILL CHECK THAT THE RDV AND WRV FLIP-FLOPS CAN BE CLEARED WHEN THE SIGNAL RST H IS ASSERTED. THE TEST WILL SET THE RDV AND WRV FLIP-FLOP'S VIA THE SIGNALS WRE H, RDE H AND CK H. THE SIGNALS WRE H AND RDE H ARE OUTPUTS OF THE MAP PROTECTION RAM. THE SIGNAL CK H IS IN CONTROL REGISTER 0. THE SIGNAL CK H IS USED TO CLOCK THE SIGNALS WRE H AND RDE H INTO THE WRV AND RDV FLIP-FLOP'S RESPECTIVELY. THE

SIGNALS WRE H AND RDE H SET TO A ZERO WILL CAUSE THE WRV AND RDV FLIP-FLOPS TO BE SET WHEN THE SIGNAL CK H IS TOGGLED IN CONTROL REGISTER 0. THE TEST WILL THEN READ CONTROL REGISTER 0 AND CHECK THAT THE SIGNALS RDV H AND WRV H ARE SET. THE TEST WILL THEN SET THE SIGNAL RST H IN CONTROL REGISTER 0 AND CHECK CHECK THAT THE SIGNALS WRV H AND RDV H CLEARED. THE SIGNAL MSBRK H WILL ALSO BE CHECKED DURING THIS TEST TO BE SET WHEN THE WRV AND RDV FLIP-FLOP'S ARE SET AND CLEARED WHEN THE RDV AND WRV FLIP-FLOPS ARE CLEARED.

TEST 17:

THIS TEST WILL CHECK MODULE SELECT RAM 0. A BINARY COUNT PATTERN WILL BE LOADED INTO EACH ADDRESS OF MODULE SELECT RAM 0. FOR EACH PATTERN LOADED, THE TEST WILL CHECK FOR THE APPROPRIATE ENABLES ON THE OUTPUT OF THE RAM. IF MORE THEN ONE BIT IS SET ON THE INPUT TO THE RAM, ONLY ONE BIT WILL BE SET ON THE OUTPUT OF THE RAM. A ONE IN THE LOW ORDER BITS OF THE RAM WILL FORCE THE HIGH ORDER BITS TO A ZERO. MODULE SELECT RAM 0 IS SELECTED BY SETTING THE SIGNAL MSEL0 H TO A ONE AND MSEL1 H TO A ZERO IN CONTROL REGISTER 2. WHEN A WRITE OR READ IS ISSUED TO CONTROL REGISTER 6, THE SIGNAL SMDS0 L IS ASSERTED WHICH SELECT MODULE SELECT RAM 0.

TEST 18:

THIS TEST WILL CHECK MODULE SELECT RAM 1. A BINARY COUNT PATTERN WILL BE LOADED INTO EACH ADDRESS OF MODULE SELECT RAM 1. FOR EACH PATTERN LOADED, THE TEST WILL CHECK FOR THE APPROPRIATE ENABLES ON THE OUTPUT OF THE RAM. IF MORE THEN ONE BIT IS SET ON THE INPUT TO THE RAM, ONLY ONE BIT WILL BE SET ON THE OUTPUT OF THE RAM. A ONE IN THE LOW ORDER BITS OF THE RAM WILL FORCE THE HIGH ORDER BITS TO A ZERO. MODULE SELECT RAM 1 IS SELECTED BY SETTING THE SIGNALS MSEL0 H AND MSEL1 H TO A ONE IN CONTROL REGISTER 2 AND THEN DOING A WRITE OR READ TO CONTROL REGISTER 6 WHICH WILL CAUSE THE SIGNAL SMDS1 L TO BE ASSERTED. THE SIGNAL SMDS1 L WILL ENABLE MODULE SELECT RAM 1 TO BE WRITTEN OR READ. MSAD BITS 17 AND 16 ARE USED TO ADDRESS MODULE SELECT RAM 1.

TEST 19:

THIS TEST WILL CHECK THAT MODULE SELECT RAM 0 AND 1 ARE ACTUALLY SELECTED WHEN THE SIGNALS MSEL0 H AND MSEL1 H ARE SET TO SELECT THE RAM'S. THE TEST WILL SELECT MODULE SELECT RAM 0, ADDRESS 0, AND WRITE A DATA PATTERN OF 1 INTO THE RAM LOCATION. THE TEST WILL THEN SELECT MODULE SELECT RAM 1, ADDRESS 0, AND WRITE A DATA PATTERN OF 10 INTO THE RAM LOCATION. THE TEST WILL THEN RESELECT MODULE SELECT RAM 0 AND CHECK THE PATTERN TO BE 1. THE TEST WILL THEN SELECT MODULE SELECT RAM 1 AND CHECK THE DATA PATTERN TO BE 10.

TEST 20:

THIS TEST WILL CHECK THAT EACH ADDRESS IN MODULE SELECT RAM 0 CAN BE ADDRESSED CORRECTLY. THIS IS DONE BY WRITING A SPECIFIC DATA PATTERN INTO THE RAM AND THEN READING THE RAM CHECKING THAT NO LOCATIONS CHANGED. THE DATA PATTERNS LOADED INTO THE RAM, STARTING AT THE LOWEST ADDRESS ARE AS FOLLOWS: 10,4,2,1,0,1,10,4.

TEST 21:

THIS TEST WILL CHECK THAT EACH ADDRESS IN MODULE SELECT RAM 1 CAN BE ADDRESSED CORRECTLY. THIS IS DONE BY WRITING A SPECIFIC DATA PATTERN INTO THE RAM AND THEN READING THE RAM CHECKING THAT NO LOCATIONS CHANGED. THE DATA PATTERNS LOADED INTO THE RAM, STARTING AT THE LOWEST ADDRESS ARE AS FOLLOWS: 1, 2, 4, AND 10.

TEST 22:

THIS TEST WILL CHECK THAT ALL MEMORY SIMULATOR RAM'S CAN BE WRITTEN AND READ WITH DATA PATTERNS 125252 AND 052525. THIS TEST CHECKS THAT ALL BITS IN THE RAM LOCATION CAN BE WRITTEN TO A ONE AND ZERO AND THAT NO ADJACENT BITS ARE SHORTED TOGETHER. THIS TEST IS EXECUTED IN 16 BIT MODE. THIS TEST DOES NOT CHECK FOR INTERNAL ADDRESS SHORTS, ADDRESS DROP OUT, OR THAT THE CORRECT MEMORY SIMULATOR RAM IS SELECTED. ALL 16K WORDS OF MEMORY SIMULATOR RAM ARE CHECKED WITH THE ALTERNATING ONES AND ZEROES PATTERN AND ALTERNATING ZEROES AND ONES PATTERN.

TEST 23:

THIS TEST WILL CHECK THAT EACH 4K MEMORY SIMULATOR RAM IS SELECTED BY WRITING A DIFFERENT DATA PATTERN INTO ADDRESS ZERO OF EACH RAM. THE TEST WILL THEN READ ADDRESS ZERO OF EACH RAM CHECKING THE DATA PATTERN TO BE THAT WHICH WAS WRITTEN PREVIOUSLY. IF A DATA FRROR OCCURS DURING THE RE-READING OF THE RAM'S, THE RAM SELECT LOGIC IS PROBABLY AT FAULT. THE DATA PATTERNS WRITTEN INTO THE MEMORY SIMULATOR RAM'S 0, 1, 2, AND 3 ARE 11111, 22222, 33333, AND 44444 RESPECTIVELY.

TEST 24:

THIS TEST WILL CHECK THAT EACH ADDRESS OF THE MEMORY SIMULATOR RAM CAN BE ADDRESSED CORRECTLY AND THAT NO ADDRESSES ARE SHORTED TOGETHER. TO DO THIS, THE PROGRAM WRITES EACH LOCATION OF THE MEMORY SIMULATOR RAM WITH DATA EQUAL TO THE ADDRESS OF THE LOCATION. AS EACH LOCATION IS WRITTEN, THE PROGRAM READS THE LOCATION AND CHECKS THE DATA TO BE EQUAL TO THE ADDRESS. WHEN ALL MEMORY SIMULATOR RAM LOCATIONS HAVE BEEN WRITTEN, THE PROGRAM WILL RE-READ THE RAMS CHECKING THAT ALL LOCATIONS CONTAIN AS DATA THEIR ADDRESS. THE PROGRAM WILL THEN RESET THE POINTER TO THE BEGINNING ADDRESS OF THE RAM'S AND DD THE FOLLOWING:

1. READ THE LOCATION AND CHECK IT TO CONTAIN ITS ADDRESS
2. WRITE THE LOCATION WITH THE ONE'S COMPLEMENT OF THE ADDRESS AND CHECK THAT THE ONE'S COMPLEMENT CAN BE READ BACK.
3. REPEAT STEPS 1 AND 2 FOR EACH ADDRESS OF THE RAM'S

THE TEST WILL THEN RESET THE POINTER TO THE BEGINNING ADDRESS OF THE RAM AND CHECK EACH LOCATION TO CONTAIN THE ONE'S COMPLEMENT OF ITS ADDRESS.

TEST 25:

THIS TEST WILL WRITE THE MEMORY SIMULATOR RAM'S WITH A COMPLEMENTING DATA PATTERN. THE INITIAL DATA PATTERN WILL BE 125252, EACH CONSECUTIVE LOCATION WILL CONTAIN THE COMPLEMENTED DATA OF THE PREVIOUS LOCATION (I.E. 125252,052525,125252....ETC.). THE ADDRESSES TO THE MEMORY SIMULATOR RAMS WILL BE LOADED AS ODD ADDRESSES. THIS IS DONE TO INSURE THAT ODD ADDRESSES IN 16 BIT MODE DO NOT DISABLE THE WRITING OR READING OF THE LOW BYTE OF THE MEMORY SIMULATOR RAM. AFTER

ALL THE RAM LOCATIONS HAVE BEEN WRITTEN WITH THE COMPLEMENTING DATA PATTERN, THE TEST WILL REREAD THE MEMORY SIMULATOR RAMS USING EVEN ADDRESSES AND CHECKING THAT THE RAM'S CONTAIN THE COMPLEMENTING DATA PATTERN.

TEST 26:

THIS TEST WILL WRITE INTO AND READ FROM THE MEMORY SIMULATOR RAMS USING THE "MOVB" INSTRUCTION. THIS TEST IS DONE IN 16 BIT MODE. THE PURPOSE OF THIS TEST IS TO CHECK THAT WHEN WRITING THE LOW BYTE OF AN ADDRESS, THE HIGH BYTE IS NOT EFFECTED, AND WHEN WRITING THE HIGH BYTE OF AN ADDRESS, THE LOW BYTE IS NOT EFFECTED. THE TEST SEQUENCE IS AS FOLLOWS:

1. WRITE THE ADDRESS TO BE TESTED INTO CONTROL REGISTER 4
2. WRITE THE MODULE SELECT RAMS TO ENABLE THE MEMORY SIMULATOR
3. WRITE AND CHECK LOW BYTE FOR DATA PATTERN OF 125
4. WRITE AND CHECK HIGH BYTE FOR DATA PATTERN OF 252
5. READ LOCATION AND CHECK WORD FOR DATA PATTERN OF 125125
6. WRITE AND CHECK LOW BYTE FOR DATA PATTERN OF 252
7. READ LOCATION AND CHECK WORD FOR DATA PATTERN OF 125252
8. WRITE AND CHECK HIGH BYTE FOR DATA PATTERN OF 125
9. READ LOCATION AND CHECK WORD FOR DATA PATTERN OF 052652
10. REPEAT STEPS 1-9 FOR ADDRESS 0 OF EACH MEMORY SIMULATOR RAM

TEST 27:

THIS TEST WILL CHECK THAT THE MEMORY SIMULATOR RAMS CAN BE WRITTEN AND READ IN 8 BIT MODE. ADDRESS 0 OF EACH MEMORY SIMULATOR RAM WILL BE TESTED IN 8 BIT MODE. THIS TEST WILL CHECK 8 BIT MODE AS FOLLOWS:

1. SET MEMORY SIMULATOR TO 16 BIT MODE
2. LOAD ADDRESS 0 OF MEMORY SIMULATOR RAM SELECTED
3. WRITE DATA PATTERN 125252 INTO RAM AND CHECK THE DATA PATTERN
4. SET MEMORY SIMULATOR TO 8 BIT MODE
5. WRITE DATA PATTERN 031463 INTO RAM + CHECK LOW BYTE FOR 063
6. SET MEMORY SIMULATOR TO 16 BIT MODE
7. READ DATA FROM MEMORY SIMULATOR RAM AND CHECK DATA TO BE 125063
8. SET MEMORY SIMULATOR TO 8 BIT MODE
9. LOAD ADDRESS 1 OF MEMORY SIMULATOR RAM SELECTED
10. WRITE DATA PATTERN 146314 INTO RAM + CHECK HIGH BYTE FOR 146000
11. SET MEMORY SIMULATOR TO 16 BIT MODE
12. READ DATA FROM MEMORY SIMULATOR RAM AND CHECK DATA TO BE 146063
13. REPEAT STEPS 1-12 FOR ADDRESS 0 OF EACH MEMORY SIMULATC

TEST 28:

THIS TEST WILL CHECK THAT INIT L CAN CLEAR THE LOW BYTE OF CONTROL REGISTER 0, AND THAT INIT H CAN PRESET THE WRV AND RDV FLIP-FLOPS. THIS IS DONE BY CLEARING(1) RDV AND WRV FLIP-FLOPS AND THEN ISSUING A BRESET INSTRUCTION WHICH SHOULD PRESET(0) THE RDV AND THE WRV FLIP-FLOPS. THEN ALL ONES ARE LOADED INTO THE LOW BYTE OF REGISTER 0 AND A BRESET INSTRUCTION IS AGAIN ISSUED WHICH SHOULD CLEAR THE LOW BYTE OF REGISTER 0.

```
                                .TITLE PROGRAM HEADER AND TABLES
                                .SBTTL  PROGRAM HEADER

                                        .ENABL  ABS
                                        .ENABL  AMA
                                        .DSABL  GBL
002000                                  .       =       2000

002000                                  BGNMOD

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

002000                                  POINTER BGNSETUP


002000                                  HEADER  CVCDA,A,0,60.,0,PRI07
002000                          L$NAME::                ;DIAGNOSTIC NAME
002000  103                             .ASCII /C/
002001  126                             .ASCII /V/
002002  103                             .ASCII /C/
002003  104                             .ASCII /D/
002004  101                             .ASCII /A/
002005  000                             .BYTE   0
002006  000                             .BYTE   0
002007  000                             .BYTE   0
002010                          L$REV::                 ;REVISION LEVEL
002010  101                             .ASCII  /A/
002011                          L$DEPO::                ;0
002011  060                             .ASCII  /0/
002012                          L$UNIT::                ;NUMBER OF UNITS
002012  000001                          .WORD   T$PTHV
002014                          L$TIML::                ;LONGEST TEST TIME
002014  000074                          .WORD   60.
002016                          L$HPCP::                ;POINTER TO H.W. QUES.
002016  021424                          .WORD   L$HARD
002020                          L$SPCP::                ;POINTER TO S.W. QUES.
002020  000000                          .WORD   0
002022                          L$HPTP::                ;PTR. TO DEF. H.W. PTABLE
002022  002216                          .WORD   L$HW
002024                          L$SPTP::                ;PTR. TO S.W. PTABLE
002024  000000                          .WORD   0
002026                          L$LADP::                ;DIAG. END ADDRESS
002026  021526                          .WORD   L$LAST
002030                          L$STA::                 ;RESERVED FOR APT STATS
002030  000000                          .WORD   0
002032                          L$CO::
002032  000000                          .WORD   0
002034                          L$DTYP::                ;DIAGNOSTIC TYPE
002034  000000                          .WORD   0
002036                          L$APT::                 ;APT EXPANSION
002036  000000                          .WORD   0
002040                          L$DTP::                 ;PTR. TO DISPATCH TABLE
002040  002124                          .WORD   L$DISPATCH
```

```
1147  002042
1148  002042  000340                L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
                                      .WORD   PRI07
1149  002044
1150  002044  000000                L$ENVI::                ;FLAGS DESCRIBE HOW IT WAS SETUP
                                      .WORD   0
1151  002046
1152  002046  000000                L$EXP1::                ;EXPANSION WORD
                                      .WORD   0
1153  002050
1154  002050     003                L$MREV::                ;SVC REV AND EDIT #
                                      .BYTE   C$REVISION
1155  002051     003                                        .BYTE   C$EDIT
1156  002052
1157  002052  000000                L$EF::                  ;DIAG. EVENT FLAGS
                                      .WORD   0
1158  002054  000000                                        .WORD   0
1159  002056                        L$SPC::
1160  002056  000000                                        .WORD   0
1161  002060                        L$DEVP::                ; POINTER TO DEVICE TYPE LIST
1162  002060  002314                                        .WORD   L$DVTYP
1163  002062                        L$REPP::                ;PTR. TO REPORT CODE
1164  002062  000000                                        .WORD   0
1165  002064                        L$EXP4::
1166  002064  000000                                        .WORD   0
1167  002066                        L$EXP5::
1168  002066  000000                                        .WORD   0
1169  002070                        L$AUT::                 ;PTR. TO ADD UNIT CODE
1170  002070  000000                                        .WORD   0
1171  002072                        L$DUT::                 ;PTR. TO DROP UNIT CODE
1172  002072  000000                                        .WORD   0
1173  002074                        L$LUN::                 ;LUN FOR EXERCISERS TO FILL
1174  002074  000000                                        .WORD   0
1175  002076                        L$DESP::                ;POINTER TO DIAG. DESCRIPTION
1176  002076  002324                                        .WORD   L$DESC
1177  002100                        L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
1178  002100  104035                                        EMT     E$LOAD
1179  002102                        L$ETP::                 ;POINTER TO ERRTBL
1180  002102  000000                                        .WORD   0
1181  002104                        L$ICP::                 ;PTR. TO INIT CODE
1182  002104  005566                                        .WORD   L$INIT
1183  002106                        L$CCP::                 ;PTR. TO CLEAN-UP CODE
1184  002106  005746                                        .WORD   L$CLEAN
1185  002110                        L$ACP::                 ;PTR. TO AUTO CODE
1186  002110  005744                                        .WORD   L$AUTO
1187  002112                        L$PRT::                 ;PTR. TO PROTECT TABLE
1188  002112  005560                                        .WORD   L$PROT
1189  002114                        L$TEST::                ;TEST NUMBER
1190  002114  000000                                        .WORD   0
1191  002116                        L$DLY::                 ;DELAY COUNT
1192  002116  000000                                        .WORD   0
1193  002120                        L$HIME::                ;PTR. TO HIGH MEM
1194  002120  000000                                        .WORD   0
1195
```

```
1196                                    .SBTTL  DISPATCH TABLE
1197
1198                                    ;++
1199                                    ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
1200                                    ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
1201                                    ;--
1202
1203  002122                                    DISPATCH 28.
1204  002122  000034                           .WORD   28
1205  002124                            L$DISPATCH::
1206  002124  006012                           .WORD   T1
1207  002126  006020                           .WORD   T2
1208  002130  006224                           .WORD   T3
1209  002132  006340                           .WORD   T4
1210  002134  006522                           .WORD   T5
1211  002136  006600                           .WORD   T6
1212  002140  006664                           .WORD   T7
1213  002142  006752                           .WORD   T8
1214  002144  007022                           .WORD   T9
1215  002146  007066                           .WORD   T10
1216  002150  007164                           .WORD   T11
1217  002152  007364                           .WORD   T12
1218  002154  007740                           .WORD   T13
1219  002156  010330                           .WORD   T14
1220  002160  011206                           .WORD   T15
1221  002162  012064                           .WORD   T16
1222  002164  012572                           .WORD   T17
1223  002166  013010                           .WORD   T18
1224  002170  013204                           .WORD   T19
1225  002172  013550                           .WORD   T20
1226  002174  014060                           .WORD   T21
1227  002176  014314                           .WORD   T22
1228  002200  014704                           .WORD   T23
1229  002202  015352                           .WORD   T24
1230  002204  016234                           .WORD   T25
1231  002206  016706                           .WORD   T26
1232  002210  017650                           .WORD   T27
1233  002212  020622                           .WORD   T28
1234
```

```
                                        .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 HARDWARE P-TABLES,
                                        ; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
                                        ;--

002214                                          BGNHW   DFPTBL
002214  000002                                  .WORD   L10000-L$HW/2
002216                                  L$HW::
002216                                  DFPTBL::

002216  163010                                  .WORD   163010                  ;CSR ADDRESS
002220  000000                                  .WORD   0                       ;DEVICE SELECTION NUMBER


002222                                          ENDHW
002222                                  L10000:

                                        .SBTTL  SOFTWARE P-TABLE

                                        ;++
                                        ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
                                        ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
                                        ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
                                        ; AT RUN TIME.
                                        ;--

002222                                          BGNSW   SFPTBL
002222  000000                                  .WORD   L10001-L$SW/2
002224                                  L$SW::
002224                                  SFPTBL::


002224                                          ENDSW
002224                                  L10001:

002224                                          ENDMOD
```

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


002224                                  BGNMOD

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


002224                                  EQUALS
                                ;
                                ; BIT DIFINITIONS
                                ;
100000                          BIT15== 100000
040000                          BIT14== 40000
020000                          BIT13== 20000
010000                          BIT12== 10000
004000                          BIT11== 4000
002000                          BIT10== 2000
001000                          BIT09== 1000
000400                          BIT08== 400
000200                          BIT07== 200
000100                          BIT06== 100
000040                          BIT05== 40
000020                          BIT04== 20
000010                          BIT03== 10
000004                          BIT02== 4
000002                          BIT01== 2
000001                          BIT00== 1
                                ;
001000                          BIT9==  BIT09
000400                          BIT8==  BIT08
000200                          BIT7==  BIT07
000100                          BIT6==  BIT06
000040                          BIT5==  BIT05
000020                          BIT4==  BIT04
000010                          BIT3==  BIT03
000004                          BIT2==  BIT02
000002                          BIT1==  BIT01
000001                          BIT0==  BIT00
                                ;
                                ; EVENT FLAG DEFINITIONS
                                ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                ;
000040                          EF.START==      32.             ; START COMMAND WAS ISSUED
000037                          EF.RESTART==    31.             ; RESTART COMMAND WAS ISSUED
000036                          EF.CONTINUE==   30.             ; CONTINUE COMMAND WAS ISSUED
000035                          EF.NEW==        29.             ; A NEW PASS HAS BEEN STARTED
000034                          EF.PWR--        28.             ; A POWER-FAIL/POWER-UP OCCURRED
                                ;
                                ;
                                ; PRIORITY LEVEL DEFINITIONS
                                ;
```

```
1331            000340                  PRI07== 340
1332            000300                  PRI06== 300
1333            000240                  PRI05== 240
1334            000200                  PRI04== 200
1335            000140                  PRI03== 140
1336            000100                  PRI02== 100
1337            000040                  PRI01== 40
1338            000000                  PRI00== 0
1339                                    ;
1340                                    ;OPERATOR FLAG BITS
1341                                    ;
1342            000004                  EVL==     4
1343            000010                  LOT==    10
1344            000020                  ADR==    20
1345            000040                  IDU==    40
1346            000100                  ISR==   100
1347            000200                  UAM==   200
1348            000400                  BOE==   400
1349            001000                  PNT==  1000
1350            002000                  PRI==  2000
1351            004000                  IXE==  4000
1352            010000                  IBE== 10000
1353            020000                  IER== 20000
1354            040000                  LOE== 40000
1355            100000                  HOE== 100000
1356
1357                                    ;
1358                                    ;MEMORY SIMULATOR CONTROL REGISTER 0
1359                                    ;
1360
1361            100000                  IDH==   BIT15                   ;BIT15=1 READ DEVICE TYPE IN 15-8
1362                                                                    ;BITS 14-12 WILL READ BACK AS 0
1363                                                                    ;MS DEVICE TYPE EQUALS 400 (BIT8=1)
1364                                                                    ;
1365                                                                    ;BIT15=0 READ DEVICE NUMBER INTO
1366                                                                    ;BITS 11-8
1367
1368            004000                  SIG11H==BIT11                   ;BITS 11-8 ARE USED TO SELECT THE
1369            002000                  SIG10H==BIT10                   ;DEVICE NUMBER TO ASSERT THE SIGNAL
1370            001000                  SIG9H== BIT9                    ;DEVE L. WHEN SELECTING MS THESE BITS
1371            000400                  SIG8H== BIT8                    ;MUST = THE SETTING OF DEV 3 - DEV 0
1372
1373                                                                    ;BIT 7 UNUSED (ALWAYS READ AS A 0)
1374
1375            000100                  CKH==   BIT6                    ;CLOCK HIGH
1376            000040                  WRVH==  BIT5                    ;WRITE VIOLATION (READ ONLY)
1377            000020                  RDVH==  BIT4                    ;READ VIOLATION  (READ ONLY)
1378            000010                  BIT8H== BIT3                    ;SELECT MS AS 8 BIT MODE NOT 16
1379            000004                  MPH==   BIT2                    ;MAP PROTECT SELECT
1380            000002                  CTSH==  BIT1                    ;BIT1=1 MEM ACCESS FROM LSI-11 BUS
1381                                                                    ;BIT1=0 MEM ACCESS FROM SYSTEM BUS
1382            000001                  RSTH==  BIT0                    ;RESET MEMORY SIMULATOR MODULE
1383
1384                                    ;
1385                                    ;MEMORY SIMULATOR CONTROL REGISTER 2
1386                                    ;
```

```
1387
1388                                                    ;BITS 15-8 ARE UNUSED BITS
1389
1390    000200          MSBRKH==BIT7                    ;MEMORY SIMULATOR BREAK (READ ONLY)
1391    000100          WRENH== BIT6                    ;WRITE ENABLE (READ ONLY)
1392    000040          ESRH==  BIT5                    ;ENABLE SIMULATOR RAM
1393
1394                                                    ;BIT 4 IS UNUSED (ALWAYS READ AS A 0)
1395
1396    000010          MSEL1== BIT3                    ;MEMORY SELECTS
1397    000004          MSEL0== BIT2                    ;
1398
1399                    ;       MSEL1=0 MSEL0=0 - SELECT SIMULATOR MEMORY - SSM L
1400                    ;       MSEL1=0 MSEL0=1 - SELECT MODULE SELECT MEMORY 0 - SMDS0 L
1401                    ;       MSEL1=1 MSEL0=0 - SELECT MAP PROTECT MEMORY - SMPM L
1402                    ;       MSEL1=1 MSEL0=1 - SELECT MODULE SELECT MEMORY 1 - SMDS1 L
1403
1404    000002          MSAD17==BIT1                    ;MS ADDRESS 17
1405    000001          MSAD16==BIT0                    ;MS ADDRESS 16
1406
1407                    ;
1408                    ;MEMORY SIMULATOR CONTROL REGISTER 4
1409                    ;
1410
1411    100000          MSAD15==BIT15                   ;MS ADDRESS 15
1412    040000          MSAD14==BIT14                   ;MS ADDRESS 14
1413    020000          MSAD13==BIT13                   ;MS ADDRESS 13
1414    010000          MSAD12==BIT12                   ;MS ADDRESS 12
1415    004000          MSAD11==BIT11                   ;MS ADDRESS 11
1416    002000          MSAD10==BIT10                   ;MS ADDRESS 10
1417    001000          MSAD9== BIT9                    ;MS ADDRESS 9
1418    000400          MSAD8== BIT8                    ;MS ADDRESS 8
1419    000200          MSAD7== BIT7                    ;MS ADDRESS 7
1420    000100          MSAD6== BIT6                    ;MS ADDRESS 6
1421    000040          MSAD5== BIT5                    ;MS ADDRESS 5
1422    000020          MSAD4== BIT4                    ;MS ADDRESS 4
1423    000010          MSAD3== BIT3                    ;MS ADDRESS 3
1424    000004          MSAD2== BIT2                    ;MS ADDRESS 2
1425    000002          MSAD1-= BIT1                    ;MS ADDRESS 1
1426    000001          MSAD0=- BIT0                    ;MS ADDRESS 0
1427
1428                    ;
1429                    ;MEMORY SIMULATOR MAP PROTECT BITS - CONTROL REGISTER 6
1430                    ;
1431
1432    000001          MPINH== BIT0                    ;MAPPED INTO MEMORY SIMULATOR
1433    000002          WREH==  BIT1                    ;WRITE ENABLED SIMULATOR MEMORY
1434    000004          RDEH==  BIT2                    ;READ ENABLED SIMULATOR MEMORY
1435    000010          MUTBH-= BIT3                    ;
```

```
1436                                    .SBTTL  GLOBAL DATA SECTION
1437
1438                                    ;++
1439                                    ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1440                                    ; IN MORE THAN ONE TEST.
1441                                    ;--
1442
1443
1444  002224                                    ERRTBL
1445  002224                            L$ERRTBL::
1446  002224  000000                    ERRTYP::        .WORD   0
1447  002226  000000                    ERRNBR::        .WORD   0
1448  002230  000000                    ERRMSG::        .WORD   0
1449  002232  000000                    ERRBLK::        .WORD   0
1450
1451                                    ;
1452                                    ;GLOBAL DATA FOR MEMORY SIMULATOR
1453                                    ;
1454
1455  002234  163010                    REG0::  .WORD   163010          ;CONTROL REGISTER 0
1456  002236  163012                    REG2::  .WORD   163012          ;CONTROL REGISTER 2
1457  002240  163014                    REG4::  .WORD   163014          ;CONTROL REGISTER 4
1458  002242  163016                    REG6::  .WORD   163016          ;CONTROL REGISTER 6
1459
1460  002244  000000                    IDDEV:: .WORD   0               ;MEMORY SIMULATOR DEVICE # (11-2)
1461  002246  000000                    UNITNB::.WORD   0               ;
1462  002250  000000                    IDTYPE::.WORD   0               ;MEMORY SIMULATOR DEVICE TYPE (15-8)
1463
1464  002252  000000                    R0LOAD::.WORD   0               ;WORD LOADED INTO REGISTER 0
1465  002254  000000                    R0GOOD::.WORD   0               ;EXPECTED REG 0
1466  002256  000000                    R0MASK::.WORD   0               ;REGISTER 0 MASK WORD
1467  002260  000000                    R0READ::.WORD   0               ;ACTUAL REG 0 READ
1468  002262  000000                    R0BAD:: .WORD   0               ;REG 0 READ MINUS R0MASK BITS
1469
1470  002264  000000                    R2LOAD::.WORD   0               ;WORD LOADED INTO REGISTER 2
1471  002266  000000                    R2GOOD::.WORD   0               ;EXPECTED REGISTER 2
1472  002270  000000                    R2MASK::.WORD   0               ;REGISTER 2 MASK WORD
1473  002272  000000                    R2READ::.WORD   0               ;ACTUAL REG 2 READ
1474  002274  000000                    R2BAD:: .WORD   0               ;REG 2 READ MINUS R2MASK BITS
1475
1476  002276  000000                    R4LOAD::.WORD   0               ;WORD LOADED INTO REGISTER 4
1477  002300  000000                    R4READ::.WORD   0               ;WORD READ OUT OF REGISTER 4
1478
1479  002302  000000                    R6LOAD::.WORD   0               ;WORD LOADED INTO REGISTER 6
1480  002304  000000                    R6GOOD::.WORD   0               ;EXPECTED REGISTER 6
1481  002306  000000                    R6MASK::.WORD   0               ;REGSITER 6 MAS  ORD
1482  002310  000000                    R6READ::.WORD   0               ;ACTUAL REGISTE  6 READ
1483  002312  000000                    R6BAD:: .WORD   0               ;REG 6 READ MINUS MASK WORD
```

```
                                        .SBTTL  GLOBAL TEXT SECTION

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


                                        ;
                                        ; NAMES OF DEVICES SUPPORTED BY PROGRAM
                                        ;
002314                                          DEVTYP  <CDS-11>
002314                                  L$DVTYP::
002314  042103  026523  030461                   .ASCIZ  /CDS-11/
002322     000
        002324                                  .EVEN


                                        ; TEST DESCRIPTION
                                        ;
002324                                          DFSCRIPT        <MEMORY SIMULATOR DIAG.>
002324                                  L$DESC::
002324  042515  047515  054522                   .ASCIZ  /MEMORY SIMULATOR DIAG./
002332  051440  046511  046125
002340  052101  051117  042040
002346  040511  027107     000
        002354                                  .EVEN


                                        ;
                                        ;ASCII MESSAGES USED BY ERROR CALLS
                                        ;

002354  040504  040524  042440  MSGMP:: .ASCIZ  /DATA ERROR IN MAP PROTECTION RAM/
002362  051122  051117  044440
002370  020116  040515  020120
002376  051120  052117  041505
002404  044524  047117  051040
002412  046501     000
002415     101  042104  042522  MSGMPS::.ASCIZ  /ADDRESS SHORT IN MAP PROTECTION RAM/
002422  051523  051440  047510
002430  052122  044440  020116
002436  040515  020120  051120
002444  052117  041505  044524
002452  047117  051040  046501
002460     000
002461     115  050101  050040  MSGMPL::.ASCIZ  /MAP PROTECT LOGIC ERROR/
002466  047522  042524  052103
002474  046040  043517  041511
002502  042440  051122  051117
002510     000
002511     104  052101  020101  MSGMD0::.ASCIZ  /DATA ERROR IN MODULE SELECT RAM 0/
002516  051105  047522  020122
002524  047111  046440  042117
002532  046125  020105  042523
002540  042514  052103  051040
```

```
1540  002546  046501  030040     000
1541  002553     104  052101  020101  MSGMD1::.ASCIZ  /DATA ERROR IN MODULE SELECT RAM 1/
1542  002560  051105  047522  020122
1543  002566  047111  046440  042117
1544  002574  046125  020105  042523
1545  002602  042514  052103  051040
1546  002610  046501  030440     000
1547  002615     103  044510  020120  MSGMDC::.ASCIZ  /CHIP ENABLE ERROR - MODULE SELECT RAM'S/
1548  002622  047105  041101  042514
1549  002630  042440  051122  051117
1550  002636  026440  046440  042117
1551  002644  046125  020105  042523
1552  002652  042514  052103  051040
1553  002660  046501  051447     000
1554  002665     115  042117  046125  MSGMDA::.ASCIZ  /MODULE SELECT RAM ADDRESSING ERROR/
1555  002672  020105  042523  042514
1556  002700  052103  051040  046501
1557  002706  040440  042104  042522
1558  002714  051523  047111  020107
1559  002722  051105  047522  000122
1560  002730  040504  040524  042440  MSGMSR::.ASCIZ  /DATA ERROR IN MEMORY SIMULATOR RAM/
1561  002736  051122  051117  044440
1562  002744  020116  042515  047515
1563  002752  054522  051440  046511
1564  002760  046125  052101  051117
1565  002766  051040  046501     000
1566  002773     103  044510  020120  MSGMSC::.ASCIZ  /CHIP ENABLE ERROR - MEMORY SIMULATOR RAM/
1567  003000  047105  041101  042514
1568  003006  042440  051122  051117
1569  003014  026440  046440  046505
1570  003022  051117  020131  044523
1571  003030  052515  040514  047524
1572  003036  020122  040522  000115
1573                                  .EVEN
1574
1575
1576                                  ;
1577                                  ; FORMAT STATEMENTS USED IN PRINT CALLS
1578                                  ;
1579
1580  003044  040445  047503  052116  EMSGR0::.ASCIZ  /%ACONTROL REG 0 ERROR%N/
1581  003052  047522  020114  042522
1582  003060  020107  020060  051105
1583  003066  047522  022522  000116
1584  003074  040445  047503  052116  EMSGR2::.ASCIZ  /%ACONTROL REG 2 ERROR%N/
1585  003102  047522  020114  042522
1586  003110  020107  020062  051105
1587  003116  047522  022522  000116
1588  003124  040445  047503  052116  EMSGR4::.ASCIZ  /%ACONTROL REG 4 ERROR%N/
1589  003132  047522  020114  042522
1590  003140  020107  020064  051105
1591  003146  047522  022522  000116
1592  003154  040445  047503  052116  EMSGR6::.ASCIZ  /%ACONTROL REG 6 ERROR%N/
1593  003162  047522  020114  042522
1594  003170  020107  020066  051105
1595  003176  047522  022522  000116
```

```
1596  003204  040445  042522  030107  REG0EQ::.ASCIZ  /%AREG0 = /
1597  003212  036440  000040
1598  003216  040445  042522  031107  REG2EQ::.ASCIZ  /%AREG2 = /
1599  003224  036440  000040
1600  003230  040445  042522  032107  REG4EQ::.ASCIZ  /%AREG4 = /
1601  003236  036440  000040
1602  003242  040445  042522  033107  REG6EQ::.ASCIZ  /%AREG6 = /
1603  003250  036440  000040
1604  003254  040445  047514  042101  FRMTR0::.ASCIZ  /%ALOAD: %06%S1%AREAD: %06%S1%AMASK: %06%S1%AGOOD: %06%S1%ABAD: %06%N/
1605  003262  020072  047445  022466
1606  003270  030523  040445  042522
1607  003276  042101  020072  047445
1608  003304  022466  030523  040445
1609  003312  040515  045523  020072
1610  003320  047445  022466  030523
1611  003326  040445  047507  042117
1612  003334  020072  047445  022466
1613  003342  030523  040445  040502
1614  003350  035104  022440  033117
1615  003356  047045     000
1616  003361     045  046101  040517  FRMTR4::.ASCIZ  /%ALOAD: %06%S1%AREAD: %06%N/
1617  003366  035104  022440  033117
1618  003374  051445  022461  051101
1619  003402  040505  035104  022440
1620  003410  033117  047045     000
1621  003415     045  052101  046511  MSGTM0::.ASCIZ  /%ATIME OUT ERROR ADDRESSING CONTROL REG 0%N/
1622  003422  020105  052517  020124
1623  003430  051105  047522  020122
1624  003436  042101  051104  051505
1625  003444  044523  043516  041440
1626  003452  047117  051124  046117
1627  003460  051040  043505  030040
1628  003466  047045     000
1629  003471     045  052101  046511  MSGTM2::.ASCIZ  /%ATIME OUT ERROR ADDRESSING CONTROL REG 2%N/
1630  003476  020105  052517  020124
1631  003504  051105  047522  020122
1632  003512  042101  051104  051505
1633  003520  044523  043516  041440
1634  003526  047117  051124  046117
1635  003534  051040  043505  031040
1636  003542  047045     000
1637  003545     045  052101  046511  MSGTM4::.ASCIZ  /%ATIME OUT ERROR ADDRESSING CONTROL REG 4%N/
1638  003552  020105  052517  020124
1639  003560  051105  047522  020122
1640  003566  042101  051104  051505
1641  003574  044523  043516  041440
1642  003602  047117  051124  046117
1643  003610  051040  043505  032040
1644  003616  047045     000
1645
1646
1647          003622                  .EVEN
1648
1649
1650
```

```
                        .SBTTL  GLOBAL ERROR REPORT SECTION

                        ;++
                        ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
                        ; USED BY MORE THAN TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
                        ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
                        ;--


003622                          BGNMSG  ROEROR
003622                  ROEROR::
003622                          PRINTB  #EMSGR0
003622  012746  003044          MOV     #EMSGR0,-(SP)
003626  012746  000001          MOV     #1,-(SP)
003632  010600                  MOV     SP,R0
003634  104414                  TRAP    C$PNTB
003636  062706  000004          ADD     #4,SP
003642                          PRINTX  #FRMTR0,R0LOAD,R0READ,R0MASK,R0GOOD,R0BAD
003642  013746  002262          MOV     R0BAD,-(SP)
003646  013746  002254          MOV     R0GOOD,-(SP)
003652  013746  002256          MOV     R0MASK,-(SP)
003656  013746  002260          MOV     R0READ,-(SP)
003662  013746  002252          MOV     R0LOAD,-(SP)
003666  012746  003254          MOV     #FRMTR0,-(SP)
003672  012746  000006          MOV     #6,-(SP)
003676  010600                  MOV     SP,R0
003700  104415                  TRAP    C$PNTX
003702  062706  000016          ADD     #16,SP
                        
                        
                        
                        
003706                          ENDMSG
003706                  L10002:
003706  104423                  TRAP    C$MSG

003710                          BGNMSG  R2EROR
003710                  R2EROR::
003710                          PRINTB  #EMSGR2
003710  012746  003074          MOV     #EMSGR2,-(SP)
003714  012746  000001          MOV     #1,-(SP)
003720  010600                  MOV     SP,R0
003722  104414                  TRAP    C$PNTB
003724  062706  000004          ADD     #4,SP
003730                          PRINTX  #FRMTR0,R2LOAD,R2READ,R2MASK,R2GOOD,R2BAD
003730  013746  002274          MOV     R2BAD,-(SP)
003734  013746  002266          MOV     R2GOOD,-(SP)
003740  013746  002270          MOV     R2MASK,-(SP)
003744  013746  002272          MOV     R2READ,-(SP)
003750  013746  002264          MOV     R2LOAD,-(SP)
003754  012746  003254          MOV     #FRMTR0,-(SP)
003760  012746  000006          MOV     #6,-(SP)
003764  010600                  MOV     SP,R0
003766  104415                  TRAP    C$PNTX
003770  062706  000016          ADD     #16,SP
```

```
1707  003774                                  ENDMSG
1708  003774                            L10003:
1709  003774  104423                          TRAP    C$MSG
1710
1711  003776                                  BGNMSG  R4EROR
1712  003776                            R4EROR::
1713  003776                                  PRINTB  #EMSGR4
1714  003776  012746  003124                  MOV     #EMSGR4,-(SP)
1715  004002  012746  000001                  MOV     #1,-(SP)
1716  004006  010600                          MOV     SP,R0
1717  004010  104414                          TRAP    C$PNTB
1718  004012  062706  000004                  ADD     #4,SP
1719  004016                                  PRINTX  #FRMTR4,R4LOAD,R4READ
1720  004016  013746  002300                  MOV     R4READ,-(SP)
1721  004022  013746  002276                  MOV     R4LOAD,-(SP)
1722  004026  012746  003361                  MOV     #FRMTR4,-(SP)
1723  004032  012746  000003                  MOV     #3,-(SP)
1724  004036  010600                          MOV     SP,R0
1725  004040  104415                          TRAP    C$PNTX
1726  004042  062706  000010                  ADD     #10,SP
1727  004046                                  ENDMSG
1728  004046                            L10004:
1729  004046  104423                          TRAP    C$MSG
1730
1731  004050                                  BGNMSG  ALINFO
1732  004050                            ALINFO::
1733  004050                                  PRINTB  #EMSGR6
1734  004050  012746  003154                  MOV     #EMSGR6,-(SP)
1735  004054  012746  000001                  MOV     #1,-(SP)
1736  004060  010600                          MOV     SP,R0
1737  004062  104414                          TRAP    C$PNTB
1738  004064  062706  000004                  ADD     #4,SP
1739  004070  004737  004240                  JSR     PC,PRNTAL               ;GO PRINT ALL THE REGISTERS
1740  004074                                  ENDMSG
1741  004074                            L10005:
1742  004074  104423                          TRAP    C$MSG
1743
1744  004076                                  BGNMSG  ALR0IN
1745  004076                            ALR0IN::
1746  004076                                  PRINTB  #EMSGR0
1747  004076  012746  003044                  MOV     #EMSGR0,-(SP)
1748  004102  012746  000001                  MOV     #1,-(SP)
1749  004106  010600                          MOV     SP,R0
1750  004110  104414                          TRAP    C$PNTB
1751  004112  062706  000004                  ADD     #4,SP
1752  004116  004737  004240                  JSR     PC,PRNTAL
1753  004122                                  ENDMSG
1754  004122                            L10006:
1755  004122  104423                          TRAP    C$MSG
1756
1757  004124                                  BGNMSG  ALR2IN
1758  004124                            ALR2IN::
1759  004124                                  PRINTB  #EMSGR2
1760  004124  012746  003074                  MOV     #EMSGR2,-(SP)
1761  004130  012746  000001                  MOV     #1,-(SP)
1762  004134  010600                          MOV     SP,R0
```

```
1763  004136  104414                          TRAP    C$PNTB
1764  004140  062706  000004                  ADD     #4,SP
1765  004144  004737  004240                  JSR     PC,PRNTAL               ;GO PRINT ALL THE REGISTERS
1766  004150                                  ENDMSG
1767  004150                          L10007:
1768  004150  104423                          TRAP    C$MSG
1769
1770  004152                                  BGNMSG  R0TM
1771  004152                          R0TM::
1772  004152                                  PRINTB  #MSGTM0
1773  004152  012746  003415                  MOV     #MSGTM0,-(SP)
1774  004156  012746  000001                  MOV     #1,-(SP)
1775  004162  010600                          MOV     SP,R0
1776  004164  104414                          TRAP    C$PNTB
1777  004166  062706  000004                  ADD     #4,SP
1778  004172                                  ENDMSG
1779  004172                          L10010:
1780  004172  104423                          TRAP    C$MSG
1781
1782  004174                                  BGNMSG  R2TM
1783  004174                          R2TM::
1784  004174                                  PRINTB  #MSGTM2
1785  004174  012746  003471                  MOV     #MSGTM2,-(SP)
1786  004200  012746  000001                  MOV     #1,-(SP)
1787  004204  010600                          MOV     SP,R0
1788  004206  104414                          TRAP    C$PNTB
1789  004210  062706  000004                  ADD     #4,SP
1790  004214                                  ENDMSG
1791  004214                          L10011:
1792  004214  104423                          TRAP    C$MSG
1793
1794  004216                                  BGNMSG  R4TM
1795  004216                          R4TM::
1796  004216                                  PRINTB  #MSGTM4
1797  004216  012746  003545                  MOV     #MSGTM4,-(SP)
1798  004222  012746  000001                  MOV     #1,-(SP)
1799  004226  010600                          MOV     SP,R0
1800  004230  104414                          TRAP    C$PNTB
1801  004232  062706  000004                  ADD     #4,SP
1802  004236                                  ENDMSG
1803  004236                          L10012:
1804  004236  104423                          TRAP    C$MSG
1805
1806  004240                          PRNTAL::PRINTX  #REGOEQ
1807  004240  012746  003204                  MOV     #REGOEQ,-(SP)
1808  004244  012746  000001                  MOV     #1,-(SP)
1809  004250  010600                          MOV     SP,R0
1810  004252  104415                          TRAP    C$PNTX
1811  004254  062706  000004                  ADD     #4,SP
1812  004260                                  PRINTX  #FRMTR0,R0LOAD,R0READ,R0MASK,R0GOOD,R0BAD
1813  004260  013746  002262                  MOV     R0BAD,-(SP)
1814  004264  013746  002254                  MOV     R0GOOD,-(SP)
1815  004270  013746  002256                  MOV     R0MASK,-(SP)
1816  004274  013746  002260                  MOV     R0READ,-(SP)
1817  004300  013746  002252                  MOV     R0LOAD,-(SP)
1818  004304  012746  003254                  MOV     #FRMTR0,-(SP)
```

```
1819  004310  012746  000006          MOV     #6,-(SP)
1820  004314  010600                  MOV     SP,R0
1821  004316  104415                  TRAP    C$PNTX
1822  004320  062706  000016          ADD     #16,SP
1823  004324                          PRINTX  #REG2EQ
1824  004324  012746  003216          MOV     #REG2EQ,-(SP)
1825  004330  012746  000001          MOV     #1,-(SP)
1826  004334  010600                  MOV     SP,R0
1827  004336  104415                  TRAP    C$PNTX
1828  004340  062706  000004          ADD     #4,SP
1829  004344                          PRINTX  #FRMTR0,R2LOAD,R2READ,R2MASK,R2GOOD,R2BAD
1830  004344  013746  002274          MOV     R2BAD,-(SP)
1831  004350  013746  002266          MOV     R2GOOD,-(SP)
1832  004354  013746  002270          MOV     R2MASK,-(SP)
1833  004360  013746  002272          MOV     R2READ,-(SP)
1834  004364  013746  002264          MOV     R2LOAD,-(SP)
1835  004370  012746  003254          MOV     #FRMTR0,-(SP)
1836  004374  012746  000006          MOV     #6,-(SP)
1837  004400  010600                  MOV     SP,R0
1838  004402  104415                  TRAP    C$PNTX
1839  004404  062706  000016          ADD     #16,SP
1840  004410                          PRINTX  #REG4EQ
1841  004410  012746  003230          MOV     #REG4EQ,-(SP)
1842  004414  012746  000001          MOV     #1,-(SP)
1843  004420  010600                  MOV     SP,R0
1844  004422  104415                  TRAP    C$PNTX
1845  004424  062706  000004          ADD     #4,SP
1846  004430                          PRINTX  #FRMTR4,R4LOAD,R4READ
1847  004430  013746  002300          MOV     R4READ,-(SP)
1848  004434  013746  002276          MOV     R4LOAD,-(SP)
1849  004440  012746  003361          MOV     #FRMTR4,-(SP)
1850  004444  012746  000003          MOV     #3,-(SP)
1851  004450  010600                  MOV     SP,R0
1852  004452  104415                  TRAP    C$PNTX
1853  004454  062706  000010          ADD     #10,SP
1854  004460                          PRINTX  #REG6EQ
1855  004460  012746  003242          MOV     #REG6EQ,-(SP)
1856  004464  012746  000001          MOV     #1,-(SP)
1857  004470  010600                  MOV     SP,R0
1858  004472  104415                  TRAP    C$PNTX
1859  004474  062706  000004          ADD     #4,SP
1860  004500                          PRINTX  #FRMTR0,R6LOAD,R6READ,R6MASK,R6GOOD,R6BAD
1861  004500  013746  002312          MOV     R6BAD,-(SP)
1862  004504  013746  002304          MOV     R6GOOD,-(SP)
1863  004510  013746  002306          MOV     R6MASK,-(SP)
1864  004514  013746  002310          MOV     R6READ,-(SP)
1865  004520  013746  002302          MOV     R6LOAD,-(SP)
1866  004524  012746  003254          MOV     #FRMTR0,-(SP)
1867  004530  012746  000006          MOV     #6,-(SP)
1868  004534  010600                  MOV     SP,R0
1869  004536  104415                  TRAP    C$PNTX
1870  004540  062706  000016          ADD     #16,SP
1871  004544  000207                  RTS     PC
1872
```

```
                                .SBTTL  GLOBAL SUBROUTINES SECTION

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

                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;    SUBROUTINE TO....SELECT AND INITIALIZE MEMORY SIMULATOR


                                ; INPUTS:
                                ;    LOCATION IDDEV CONTAINS USER DEFINED DEVICE NUMBER IN BITS 11-8



                                ; IMPLICIT INPUTS:


                                ; OUTPUTS.
                                ;       R0LOAD CONTAINS USER DEFINED UNIT NUMBER IN BITS 11-8
                                ;       R0MASK CONTAINS CONTROL REGISTER 0 MASK WORD (000000)
                                ;        R2LOAD CONTAINS ALL 0'S IN R/W BITS MSEL1 H, MSEL2 H, MSAD17 H, MSAD16 H
                                ;        R2MASK CONTAINS CONTROL REGISTER 2 MASK WORD (177760)
                                ;        R4LOAD CONTAINS ALL ZEROES IN R/W BITS MSAD 15:0


                                ; IMPLICIT OUTPUTS:


                                ; SUBORDINATE ROUTINES USED:
                                ;    LDRDR0 ROUTINE TO LOAD, READ AND COMPARE REGISTER 0
                                ;    LDRD0R ROUTINE TO LOAD, READ AND COMPARE REGISTER 0 (USED FOR DEVICE TYPE)
                                ;    LDRDR2 ROUTINE TO LOAD, READ AND COMPARE REGISTER 2
                                ;    LDRDR4 ROUTINE TO LOAD, READ AND COMAPRE REGISTER 4


                                ; FUNCTIONAL SIDE EFFECTS:
                                ;    MEMORY SIMULATOR SELECTED
                                ;    CONTROL REGISTER LOW BYTE EQUALS 0
                                ;    CONTROL REGISTER 2 R/W BITS CLEARED (MSEL1,MSEL2,MSAD17 AND MSAD16)
                                ;    CONTROL REGISTER 4 R/W BITS CLEARED (MSAD 15:0)


                                ; CALLING SEQUENCE:
                                ;    JSR PC,INITMS

                                ;--



004546                          INITMS::BGNSEG                          ;ROUTINE TO INIT MS MODULE
004546  104404                          TRAP    C$BSEG
004550                                  SETVEC  #4,#1$,#PRIO7           ;SETUP VECTOR
004550  012746  000340                  MOV     #PRIO7,-(SP)
```

```
1929  004554  012746  004674                  MOV     #1$,-(SP)
1930  004560  012746  000004                  MOV     #4,-(SP)
1931  004564  012746  000003                  MOV     #3,-(SP)
1932  004570  104437                          TRAP    C$SVEC
1933  004572  062706  000010                  ADD     #10,SP
1934
1935                                          ;LOAD DEVICE NUMBER INTO REGISTER 0 AND CHECK IT
1936
1937  004576  013737  002244  002252          MOV     IDDEV,ROLOAD            ;GET USER DEFINED DEVICE NUMBER
1938  004604  012737  000060  002256          MOV     #RDVH!WRVH,ROMASK       ;SETUP TO MASK READ ONLY BITS
1939  004612  013737  002252  002254          MOV     ROLOAD,ROGOOD           ;PUT DATA LOADED INTO EXPECTED
1940  004620  013777  002252  175406          MOV     ROLOAD,@REG0            ;WRITE WORD TO REG 0
1941  004626  017737  175402  002260          MOV     @REG0,ROREAD            ;READ REGISTER CONTENTS BACK
1942  004634  013737  002260  002262          MOV     ROREAD,ROBAD            ;COPY REG 0 READ TO ALLOW MASKING
1943  004642  043737  002256  002262          BIC     ROMASK,ROBAD            ;CLEAR UNWANTED BITS OF REG 0
1944  004650  023737  002254  002262          CMP     ROGOOD,ROBAD            ;COMPARE EXPECTED WITH THAT READ
1945  004656  001414                          BEQ     2$                      ;IF COMPARE WAS GOOD THEN CONT
1946  004660                                  ERRDF   1,,ROEROR               ;DEVICE # OR LB NOT = EXPECTED
1947  004660  104455                          TRAP    C$ERDF
1948  004662  000001                          .WORD   1
1949  004664  000000                          .WORD   0
1950  004666  003622                          .WORD   ROEROR
1951  004670                                  CKLOOP
1952  004670  104406                          TRAP    C$CLP1
1953  004672  000406                          BR      2$                      ;BRANCH AROUND TIME OUT ERROR
1954  004674  005726                  1$:     TST     (SP)+                   ;CLEAN UP STACK
1955  004676  005726                          TST     (SP)+                   ;CLEAN UP STACK
1956  004700                                  ERRDF   1,,ROTM                 ;TIME OUT ERROR REG 0
1957  004700  104455                          TRAP    C$ERDF
1958  004702  000001                          .WORD   1
1959  004704  000000                          .WORD   0
1960  004706  004152                          .WORD   ROTM
1961  004710                          2$:     CLRVEC  #4                      ;CLEAR VECTOR
1962  004710  012700  000004                  MOV     #4,R0
1963  004714  104436                          TRAP    C$CVEC
1964  004716                                  ENDSEG
1965  004716                          10000$:
1966  004716  104405                          TRAP    C$ESEG
1967
1968                                          ;READ DEVICE TYPE IN REGISTER 0
1969
1970  004720                                  BGNSEG
1971  004720  104404                          TRAP    C$BSEG
1972  004722  052737  100000  002252          BIS     #IDH,ROLOAD             ;SETUP TO READ DEVICE TYPE
1973  004730  013737  002250  002254          MOV     IDTYPE,ROGOOD           ;SETUP EXPECTED DATA
1974  004736  004737  005352                  JSR     PC,LDRDOR               ;LOAD, READ AND COMPARE REG 0
1975  004742  001404                          BEQ     3$                      ;IF EQUAL THEN DEVICE TYPE COMPARED
1976  004744                                  ERRDF   1,,ROEROR               ;DEVICE TYPE NOT EQUAL EXPECTED
1977  004744  104455                          TRAP    C$ERDF
1978  004746  000001                          .WORD   1
1979  004750  000000                          .WORD   0
1980  004752  003622                          .WORD   ROEROR
1981  004754                          3$:     ENDSEG
1982  004754                          10001$:
1983  004754  104405                          TRAP    C$ESEG
1984
```

```
1985                                                ;SET SIGNAL RSTH TO 0 WRV AND RDV F/F'S - 0 SIGNAL I/D H
1986
1987  004756                                        BGNSEG
1988  004756  104404                               TRAP    C$BSEG
1989  004760  013737  002244  002252               MOV     IDDEV,ROLOAD            ;GOT USER DEFINED DEVICE NUMBER
1990  004766  052737  000001  002252               BIS     #RSTH,ROLOAD            ;SET BIT TO ISSUE SIGNAL RST H
1991  004774  042737  000060  002256               BIC     #WRVH!RDVH,ROMASK       ;CLEAR READ ONLY BITS RDVH + WRV H
1992  005002  004737  005344                       JSR     PC,LDRDRO               ;GO ISSUE SIGNAL RST H TO 0 WRV + RDV
1993  005006  001404                               BEQ     4$                      ;IF OK THEN CONTINUE
1994  005010                                        ERRDF   1,,ROEROR               ;RSTH NOT SET OR RDV OR WRV F/F'S SET
1995  005010  104455                               TRAP    C$ERDF
1996  005012  000001                               .WORD   1
1997  005014  000000                               .WORD   0
1998  005016  003622                               .WORD   ROEROR
1999  005020                              4$:       ENDSEG
2000  005020                              10002$:
2001  005020  104405                               TRAP    C$ESEG
2002
2003                                                ;CLEAR SIGNAL RST H TO CHECK THAT RDV + WRV REMAIN A ZERO
2004
2005  005022                                        BGNSEG
2006  005022  104404                               TRAP    C$BSEG
2007  005024  042737  000001  002252               BIC     #RSTH,ROLOAD            ;SETUP TO CLEAR RSTH BIT IN REG 0
2008  005032  004737  005344                       JSR     PC,LDRDRO               ;LOAD, READ AND COMPARE REG 0
2009  005036  001404                               BEQ     5$                      ;IF OK THEN CONTINUE
2010  005040                                        ERRDF   1,,ROEROR               ;RDV OR WRV STILL SET OR RST H SET
2011  005040  104455                               TRAP    C$ERDF
2012  005042  000001                               .WORD   1
2013  005044  000000                               .WORD   0
2014  005046  003622                               .WORD   ROEROR
2015  005050                              5$:       ENDSEG
2016  005050                              10003$:
2017  005050  104405                               TRAP    C$ESEG
2018
2019                                                ;CLEAR THE SIGNALS MSEL1 H, MSEL2 H, MSAD17 H AND MSAD16 H IN CONTROL
2020                                                ;REGISTER 2 AND CHECK THAT THESE SIGNALS CLEARED BY READING CONTROL
2021                                                ;REGISTER 2 BACK AGAIN.
2022
2023  005052                                        BGNSEG
2024  005052  104404                               TRAP    C$BSEG
2025  005054                                        SETVEC  #4,#6$,#PRI07           ;SETUP VECTOR
2026  005054  012746  000340                       MOV     #PRI07,-(SP)
2027  005060  012746  005176                       MOV     #6$,-(SP)
2028  005064  012746  000004                       MOV     #4,-(SP)
2029  005070  012746  000003                       MOV     #3,-(SP)
2030  005074  104437                               TRAP    C$SVEC
2031  005076  062706  000010                       ADD     #10,SP
2032  005102  005037  002264                       CLR     R2LOAD                  ;SETUP TO CLEAR R/W BITS
2033  005106  012737  177740  002270               MOV     #177740,R2MASK          ;IGNORE READ ONLY AND UNUSED BITS
2034  005114  013737  002264  002266               MOV     R2LOAD,R2GOOD           ;PUT DATA TO BE LOADED INTO EXPECTED
2035  005122  013777  002264  175106               MOV     R2LOAD,@REG2            ;WRITE BITS INTO REGISTER 2
2036  005130  017737  175102  002272               MOV     @REG2,R2READ            ;READ REGISTER 2 BACK
2037  005136  013737  002272  002274               MOV     R2READ,R2BAD            ;COPY DATA READ
2038  005144  043737  002270  002274               BIC     R2MASK,R2BAD            ;CLEAR UNWANTED BITS IN REG 2
2039  005152  023737  002266  002274               CMP     R2GOOD,R2BAD            ;CHECK IF EXP EQUALS ACTUAL
2040  005160  001414                               BEQ     7$                      ;IF LOADED OK THEN CONTINUE
```

```
2041  005162                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL TO 0
2042  005162  104455                          TRAP    C$ERDF
2043  005164  000002                          .WORD   2
2044  005166  000000                          .WORD   0
2045  005170  003710                          .WORD   R2EROR
2046  005172                                  CKLOOP
2047  005172  104406                          TRAP    C$CLP1
2048  005174  000406                          BR      7$                      ;BRANCH AROUND TIME OUT ERROR
2049  005176  005726                  6$:     TST     (SP)+                   ;CLEAN UP STACK
2050  005200  005726                          TST     (SP)+                   ;CLEAN UP STACK
2051  005202                                  ERRDF   2,,R2TM                 ;TIME OUT ERROR REG 2
2052  005202  104455                          TRAP    C$ERDF
2053  005204  000002                          .WORD   2
2054  005206  000000                          .WORD   0
2055  005210  004174                          .WORD   R2TM
2056  005212                          7$:     CLRVEC  #4                      ;CLEAR VECTOR
2057  005212  012700  000004                 MOV     #4,R0
2058  005216  104436                          TRAP    C$CVEC
2059  005220                                  ENDSEG
2060  005220                          10004$:
2061  005220  104405                          TRAP    C$ESEG
2062
2063                                          ;CLEAR MSAD BITS 15:0 IN CONTROL REGISTER 4 AND CHECK THAT THESE BITS
2064                                          ;ARE CLEAR BY READING BACK CONTROL REGISTER 4.
2065
2066  005222                                  BGNSEG
2067  005222  104404                          TRAP    C$BSEG
2068  005224                                  SETVEC  #4,#8$,#PRI07           ;SETUP VECTOR
2069  005224  012746  000340                 MOV     #PRI07,-(SP)
2070  005230  012746  005316                 MOV     #8$,-(SP)
2071  005234  012746  000004                 MOV     #4,-(SP)
2072  005240  012746  000003                 MOV     #3,-(SP)
2073  005244  104437                          TRAP    C$SVEC
2074  005246  062706  000010                 ADD     #10,SP
2075  005252  005037  002276                 CLR     R4LOAD                  ;SETUP TO CLEAR ALL MSAD BITS
2076  005256  013777  002276  174754         MOV     R4LOAD,@REG4            ;WRITE WORD INTO REGISTER 4
2077  005264  017737  174750  002300         MOV     @REG4,R4READ            ;READ WORD BACK FROM REGISTER 4
2078  005272  023737  002276  002300         CMP     R4LOAD,R4READ           ;COMPARE WORD LOADED WITH READ
2079  005300  001414                          BEQ     9$                      ;IF LOADED OK THEN CONTINUE
2080  005302                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL TO ZERO
2081  005302  104455                          TRAP    C$ERDF
2082  005304  000003                          .WORD   3
2083  005306  000000                          .WORD   0
2084  005310  003776                          .WORD   R4EROR
2085  005312                                  CKLOOP
2086  005312  104406                          TRAP    C$CLP1
2087  005314  000406                          BR      9$                      ;BRANCH AROUND TIME OUT ERROR
2088  005316  005726                  8$:     TST     (SP)+                   ;CLEAN UP STACK
2089  005320  005726                          TST     (SP)+                   ;CLEAN UP STACK
2090  005322                                  ERRDF   3,,R4TM                 ;TIME OUT ERROR REG 4
2091  005322  104455                          TRAP    C$ERDF
2092  005324  000003                          .WORD   3
2093  005326  000000                          .WORD   0
2094  005330  004216                          .WORD   R4TM
2095  005332                          9$:     CLRVEC  #4                      ;CLEAR VECTOR
2096  005332  012700  000004                 MOV     #4,R0
```

```
2097  005336  104436                          TRAP    C$CVEC
2098  005340                                  ENDSEG
2099  005340                          10005$:
2100  005340  104405                          TRAP    C$ESEG
2101  005342  000207                          RTS     PC                      ;RETURN BACK TO TEST
2102
2103                                  ;ROUTINE TO LOAD, READ, AND COMPARE CONTENTS OF REGISTER 0
2104                                  ;CONDITION CODES ARE SET ON EXIT AS RESULT OF THE ''CMP'' INSTRUCTION.
2105
2106  005344  013737  002252  002254  LDRDR0::MOV     R0LOAD,R0GOOD           ;PUT DATA LOADED INTO EXPECTED
2107  005352  013777  002252  174654  LDRD0R::MOV     R0LOAD,@REG0            ;WRITE WORD TO REGISTER 0
2108  005360  017737  174650  002260  READR0::MOV     @REG0,R0READ            ;READ REGISTER CONTENTS BACK
2109  005366  013737  002260  002262          MOV     R0READ,R0BAD            ;COPY REG 0 READ TO ALLOW MASKING
2110  005374  043737  002256  002262          BIC     R0MASK,R0BAD            ;CLEAR UNWANTED BITS OF REG 0
2111  005402  023737  002254  002262          CMP     R0GOOD,R0BAD            ;COMPARE EXPECTED WITH THAT READ
2112  005410  000207                          RTS     PC                      ;EXIT WITH CONDITION CODES SET
2113
2114                                  ;ROUTINE TO LOAD, READ, AND COMPARE CONTENTS OF REGISTER 2.
2115                                  ;CONDITION CODES ARE SET ON EXIT AS RESULT OF ''CMP'' INSTRUCTION
2116
2117  005412  013737  002264  002266  LDRDR2::MOV     R2LOAD,R2GOOD           ;PUT DATA TO BE LOADED INTO EXPECTED
2118  005420  013777  002264  174610  LDRD2R::MOV     R2LOAD,@REG2            ;WRITE BITS INTO REGISTER 2
2119  005426  017737  174604  002272  READR2::MOV     @REG2,R2READ            ;READ REGISTER 2 BACK
2120  005434  013737  002272  002274          MOV     R2READ,R2BAD            ;COPY DATA READ
2121  005442  043737  002270  002274          BIC     R2MASK,R2BAD            ;CLEAR UNWANTED BITS IN REG 2
2122  005450  023737  002266  002274          CMP     R2GOOD,R2BAD            ;CHECK IF EXP EQUALS ACTUAL
2123  005456  000207                          RTS     PC                      ;EXIT WITH CONDITION CODES SET
2124
2125                                  ;ROUTINE TO LOAD, READ AND COMPARE CONTENTS OF REGISTER 4.
2126                                  ;CONDITION CODES ARE SET ON EXET AS RESULT OF ''CMP'' INSTRUCTION.
2127
2128  005460  013777  002276  174552  LDRDR4::MOV     R4LOAD,@REG4            ;WRITE WORD INTO REGISTER 4
2129  005466  017737  174546  002300  READR4::MOV     @REG4,R4READ            ;READ WORD BACK FROM REGISTER 4
2130  005474  023737  002276  002300          CMP     R4LOAD,R4READ           ;COMPARE WORD LOADED WITH READ
2131  005502  000207                          RTS     PC                      ;RETURN WITH CONDITION CODES SET
2132
2133                                  ;ROUTINE TO LOAD, READ AND COMPARE CONTENTS OF CONTROL REGISTER 6
2134                                  ;CONDITION CODES ARE SET ON EXIT AS RESULT OF ''CMP'' INSTRUCTION.
2135
2136  005504  013737  002302  002304  LDRDR6::MOV     R6LOAD,R6GOOD           ;COPY DATA TO BE LOADED
2137  005512  013777  002302  174522  LDRD6R::MOV     R6LOAD,@REG6            ;WRITE WORD INTO REGISTER 6
2138  005520  017737  174516  002310  READR6::MOV     @REG6,R6READ            ;READ THE WORD BACK
2139  005526  013737  002310  002312          MOV     R6READ,R6BAD            ;COPY DATA READ
2140  005534  043737  002306  002312          BIC     R6MASK,R6BAD            ;MASK OUT UNWANTED BITS
2141  005542  023737  002304  002312          CMP     R6GOOD,R6BAD            ;COMPARE DATA LOADED WITH DATA READ
2142  005550  000207                          RTS     PC                      ;EXIT WITH CONDTION CODES SET
2143
2144  005552                                  ENDMOD
2145
```

GLOBAL SUBROUTINES SECTION SEQ 0042

```
                                .TITLE MISCELLANEOUS SECTIONS
                                .SBTTL  REPORT CODING SECTION

005552                                  BGNMOD

                                ;++
                                ; THE REPORT CODING SECTION CONTAINS THE
                                ; 'PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;--

005552                                  BGNRPT
005552                          L$RPT::

005552                                  EXIT    RPT
005552  000167                          .WORD   J$JMP
005554  000000                          .WORD   L10013-2-.

                                        .EVEN

005556                                  ENDRPT
005556                          L10013:
005556  104425                          TRAP    C$RPT

                                .SBTTL  PROTECTION TABLE

                                ;++
                                ; THIS TABLE IS USED BY THE RUNTIME SERVICES
                                ; TO PROTECT THE LOAD MEDIA.
                                ;--

005560                                  BGNPROT
005560                          L$PROT::

005560  177777                          -1              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
005562  177777                          -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
005564  177777                          -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

005566                                  ENDPROT
```

```
                        .SBTTL  INITIALIZE SECTION

                        ;++
                        ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                        ; AT THE BEGINNING OF EACH PASS.
                        ;--

                                BGNINIT
                        L$INIT::
                                READEF  #EF.START               ;SEE IF A START COMMAND
005566  012700  000040          MOV     #EF.START,R0
005572  104447                  TRAP    C$REFG
005574                          BCOMPLETE       1$              ;BRANCH IF START COMMAND
005574  103410                  BCS     1$
005576                          READEF  #EF.RESTART             ;SEE IF A RESTART COMMAND
005576  012700  000037          MOV     #EF.RESTART,R0
005602  104447                  TRAP    C$REFG
005604                          BCOMPLETE       1$              ;BRANCH IF RESTART
005604  103404                  BCS     1$
005606                          READEF  #EF.PWR                 ;SEE IF RECOVERING FROM A POWER FAIL
005606  012700  000034          MOV     #EF.PWR,R0
005612  104447                  TRAP    C$REFG
005614                          BNCOMPLETE      2$              ;IF NOT CHECK IN CONTINUE
005614  103001                  BCC     2$
005616                  1$:     BRESET                          ;INITIALIZE THE SYSTEM TO A KNOWN STATE
005616  104433                  TRAP    C$RESET
005620                  2$:     READEF  #EF.NEW                 ;SEE IF A NEW PASS
005620  012700  000035          MOV     #EF.NEW,R0
005624  104447                  TRAP    C$REFG
005626                          BNCOMPLETE      3$              ;IF NOT GO CHECK IF CONTINUE
005626  103003                  BCC     3$
005630  012737  177777  002246  MOV     #-1,UNITNB              ;SETUP TO INIT UNIT NUMBER
005636                  3$:     READEF  #EF.CONTINUE            ;CHECK IF CONTINUE
005636  012700  000036          MOV     #EF.CONTINUE,R0
005642  104447                  TRAP    C$REFG
005644                          BCOMPLETE       6$              ;IF YES THEN EXIT
005644  103431                  BCS     6$
005646  005237  002246  4$:     INC     UNITNB                  ;INC TO NEW UNIT NUMBER
005652                          GPHARD  UNITNB,R5               ;GET DEVICE INFORMATION
005652  013700  002246          MOV     UNITNB,R0
005656  104442                  TRAP    C$GPHRD
005660  010005                  MOV     R0,R5
005662                          BNCOMPLETE      4$              ;GO TRY ANOTHER UNIT
005662  103371                  BCC     4$
005664  012701  002234          MOV     #REG0,R1                ;ADDRESS OF MS DEVICE ADDRESS TABLE
005670  005002                  CLR     R2                      ;CLEAR OFFSET TO ADD TO TABLE ADDRESS
005672  011511          5$:     MOV     (R5),(R1)               ;GET ADDRESS AND SAVE
005674  060221                  ADD     R2,(R1)+                ;ADD OFFSET TO ADDRESS
005676  005202                  INC     R2                      ;UPDATE OFFSET BY 2
005700  005202                  INC     R2                      ;
005702  022702  000010          CMP     #10,R2                  ;CHECK IF DONE LOADING TABLE
005706  001371                  BNE     5$                      ;GO UPDATE NEXT ADDRESS
005710  005725                  TST     (R5)+                   ;UPDATE THE POINTER
005712  005037  002244          CLR     IDDEV                   ;CLEAR OUT DEVICE NUMBER
005716  111537  002245          MOVB    (R5),IDDEV+1            ;GET THE MS DEVICE NUMBER
005722  012737  100400  002250  MOV     #IDH!SIG8H,IDTYPE       ;SETUP MS DEVICE TYPE
```

```
2243  005730                          6$:     SETPRI  #PRI07                  ;RAISE PROCESSOR PRIORITY
2244  005730  012700  000340                MOV     #PRI07,R0
2245  005734  104441                      TRAP    C$SPRI
2246
2247
2248  005736                          EXIT    INIT
2249  005736  104432                  TRAP    C$EXIT
2250  005740  000002                  .WORD   L10015-.
2251
2252
2253                                  .EVEN
2254
2255  005742                          ENDINIT
2256  005742                          L10015:
2257  005742  104411                  TRAP    C$INIT
2258
2259                                  .SBTTL  AUTODROP SECTION
2260
2261                                  ;++
2262                                  ; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
2263                                  ; THE "ADR" FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
2264                                  ; SEE IF THEY WILL RESPOND.  THOSE THAT DON'T ARE IMMEDIATELY
2265                                  ; DROPPED FROM TESTING.
2266                                  ;--
2267
2268  005744                          BGNAUTO
2269  005744                          L$AUTO::
2270
2271
2272  005744                          ENDAUTO
2273  005744                          L10016:
2274  005744  104461                  TRAP    C$AUTO
2275
2276                                  .SBTTL  CLEANUP CODING SECTION
2277
2278                                  ;++
2279                                  ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
2280                                  ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
2281                                  ;--
2282
2283  005746                          BGNCLN
2284  005746                          L$CLEAN::
2285  005746  013777  002244  174260  MOV     IDDEV,@REG0             ;CLEAR CONTROL REGISTER 0 EXCEPT
2286                                                                  ;FOR DEVICE NUMBER
2287  005754  012777  000000  174254  MOV     #0,@REG2                ;CLEAR REGISTER 2
2288  005762  012777  000000  174250  MOV     #0,@REG4                ;CLEAR REGISTER 4
2289
2290
2291  005770                          EXIT    CLN
2292  005770  104432                  TRAP    C$EXIT
2293  005772  000002                  .WORD   L10017-.
2294
2295
2296                                  .EVEN
2297
2298  005774                          ENDCLN
```

CLEANUP CODING SECTION SEQ 0045

```
2299  005774                          L10017:
2300  005774  104412                          TRAP    C$CLEAN
2301
2302                                  .SBTTL  DROP UNIT SECTION
2303
2304                                  ;++
2305                                  ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
2306                                  ; TO NO LONGER BE TESTED.
2307                                  ;--
2308
2309  005776                                  BGNDU
2310  005776                          L$DU::
2311
2312
2313  005776                                  EXIT    DU
2314  005776  000167                          .WORD   J$JMP
2315  006000  000000                          .WORD   L10020-2-.
2316
2317
2318                                          .EVEN
2319
2320  006002                                  ENDDU
2321  006002                          L10020:
2322  006002  104453                          TRAP    C$DU
2323
2324                                  .SBTTL  ADD UNIT SECTION
2325
2326                                  ;++
2327                                  ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
2328                                  ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
2329                                  ;  TO THE TEST CYCLE.
2330                                  ;--
2331
2332  006004                                  BGNAU
2333  006004                          L$AU::
2334
2335
2336  006004                                  EXIT    AU
2337  006004  000167                          .WORD   J$JMP
2338  006006  000000                          .WORD   L10021-2-.
2339
2340
2341                                          .EVEN
2342
2343  006010                                  ENDAU
2344  006010                          L10021:
2345  006010  104452                          TRAP    C$AU
2346
2347  006012                                  ENDMOD
2348
```

```
                                .TITLE HARDWARE TESTS

                                .SBTTL  TEST 1: SELECT AND INITIALIZE MEMORY SIMULATOR

006012                                  BGNMOD

                                ;++
                                ; TEST TO CHECK THAT THE MEMORY SIMULATOR CAN BE SELECTED AND INITIALIZED
                                ; TO A KNOWN STATE.  THIS TEST WILL BE EXECUTED AT THE BEGINNING OF EVERY
                                ; TEST TO PUT THE MODULE IN A KNOWN STATE.  THE TEST WILL LOAD THE DEVICE
                                ; NUMBER INTO REGISTER 0 AND CHECK THAT THE DEVICE NUMBER CAN BE READ BACK
                                ; CORRECTLY.  THE R/W BITS IN THE LOW BYTE ARE CHECKED TO BE CLEARED. THE
                                ; READ ONLY BITS, WRV H AND RDV H, ARE NOT CHECKED. THE TEST WILL LOAD THE
                                ; DEVICE NUMBER AND THE SIGNAL I/D H INTO REGISTER 0 AND CHECK THAT THE
                                ; DEVICE TYPE AND THE R/W BITS IN THE LOW BYTE CAN BE READ BACK CORRECTLY.
                                ; THE TEST WILL THEN LOAD THE DEVICE NUMBER AND THE SIGNAL RST H (CLEAR
                                ; READ AND WRITE VIOLATION FLIP-FLOPS) INTO REGISTER 0 AND CHECK THAT THE
                                ; DEVICE NUMBER AND THE SIGNAL RST H ARE SET IN REGISTER 0.  THE OTHER READ
                                ; WRITE BITS AND THE READ ONLY BITS (WRVH AND RDVH) ARE CHECKED TO BE
                                ; CLEARED.  THE LAST PART OF THIS TEST WILL BE TO CLEAR THE SIGNAL RST H
                                ; IN REGISTER 0 AND CHECK THAT REGISTER 0 IS READ BACK CORRECTLY.
                                ;--


006012                                  BGNTST
006012                          T1::
006012  004737  004546                  JSR     PC,INITMS               ;INITIALIZE THE MEMORY SIMULATOR


006016                                  ENDTST
006016                          L10022:
006016  104401                          TRAP    C$ETST
```

```
                                        .SBTTL TEST 2: CONTROL REG 0 TEST (1'S, 0'S, 1'S + 0'S, 0'S + 1'S)

                                        ;++
                                        ; TEST TO CHECK CONTROL REGISTER 0.  THIS TEST WILL CHECK THAT THE READ/WRITE
                                        ; BITS RST H, CTS H, MP H, 8 BIT H, AND CK H, CAN BE SET AND CLEARED.  THIS
                                        ; IS A BASIC TEST OF THE READ/WRITE BITS.  THE READ ONLY BITS RDV H AND WRV H
                                        ; WILL BE CHECKED TO BE CLEARED WHEN THE SIGNAL RST H IS ASSERTED,OTHERWISE,
                                        ; THOSE BITS WILL BE IGNORED.  THE TEST PATTERNS USED ARE AS FOLLOWS:
                                        ;    1. SET ALL R/W BITS TO A 1 AND THEN A 0
                                        ;    2. SET R/W BITS TO ALTERNATING 1'S + 0'S AND THEN 0'S + 1'S
                                        ;--

006020                                          BGNTST
006020                                  T2::
006020  004737  004546                          JSR     PC,INITMS               ;SELECT AND INITIALIZE MEM SIM

006024                                          BGNSUB
006024                                  T2.1:
006024  104402                                  TRAP    C$BSUB

                                        ;CHECK THE RST H, CTS H, MP H, 8 BIT H AND CK H CAN BE SET TO 1
                                        ;RDV H AND WRV H WILL BE CHECKED TO BE 0

006026                                          BGNSEG
006026  104404                                  TRAP    C$BSEG
006030  052737  000117  002252                  BIS     #RSTH!CTSH!MPH!BIT8H.CKH,ROLOAD ;SET ALL R/W BITS TO 1
006036  042737  000060  002256                  BIC     #RDVH!WRVH,ROMASK       ;CLEAR MASK BITS FOR RDV AND WRV
006044  004737  005344                          JSR     PC,LDRDR0               ;GO LOAD, READ AND COMPARE REG 0
006050  001404                                  BEQ     1$                      ;IF LOAD EQUALS READ - CONTINUE
006052                                          ERRDF   1,,ROEROR               ;R/W BITS NOT ALL SET
006052  104455                                  TRAP    C$ERDF
006054  000001                                  .WORD   1
006056  000000                                  .WORD   0
006060  003622                                  .WORD   ROEROR
006062                                  1$:     ENDSEG
006062                                  10000$:
006062  104405                                  TRAP    C$ESEG

                                        ;CHECK THAT ALL R/W BITS CAN BE CLEARED
                                        ;RDV AND WRV ARE IGNORED DURING THIS SUB TEST

006064                                          BGNSEG
006064  104404                                  TRAP    C$BSEG
006066  105037  002252                          CLRB    ROLOAD                  ;CLEAR ALL R/W BITS TO BE LOADED
006072  052737  000060  002256                  BIS     #RDVH!WRVH,ROMASK       ;SETUP TO IGNORE RDV AND WRV BITS
006100  004737  005344                          JSR     PC,LDRDR0               ;GO LOAD, READ AND COMPARE REG 0
006104  001404                                  BEQ     2$                      ;IF ALL R/W BITS ARE 0 - CONT
006106                                          ERRDF   1,,ROEROR               ;ALL R/W BITS NOT 0
006106  104455                                  TRAP    C$ERDF
006110  000001                                  .WORD   1
006112  000000                                  .WORD   0
006114  003622                                  .WORD   ROEROR
006116                                  2$:     ENDSEG
006116                                  10001$:
006116  104405                                  TRAP    C$ESEG
```

```
2441  006120                                    ENDSUB
2442  006120                              L10024:
2443  006120  104403                             TRAP    C$ESUB
2444
2445  006122                                    BGNSUB
2446  006122                              T2.2:
2447  006122  104402                             TRAP    C$BSUB
2448
2449                                            ;CHECK THAT ALTERNATING BITS IN REG 0 CAN BE SET AND CLEARED
2450                                            ;
24                                              ;CHECK THAT CTS H AND 8 BIT H ARE 0 WHEN RST H, MP H AND CK H ARE 1
245                                             ;RDV H AND WRV H ARE CHECKED TO BE 0
2453
2454  006124                                    BGNSEG
2455  006124  104404                           TRAP    C$BSEG
2456  006126  112737  000105  002252           MOVB    #RSTH!MPH!CKH,ROLOAD    ;SETUP BITS TO LOAD
2457  006134  042737  000060  002256           BIC     #RDVH.WRVH,ROMASK       ;SETUP TO CHECK RDV AND WRV
2458  006142  004737  005344                    JSR     PC,LDRDRO               ;GO LOAD, READ AND COMPARE REG 0
2459  006146  001404                           BEQ     3$                      ;IF OK THEN CONTINUE
2460  006150                                    ERRDF   1,,ROEROR               ;A BIT IS SHORTED TO ANOTHER
2461  006150  104455                           TRAP    C$ERDF
2462  006152  000001                           .WORD   1
2463  006154  000000                           .WORD   0
2464  006156  003622                           .WORD   ROEROR
2465  006160                              3$:   ENDSEG
2466  006160                              10000$:
2467  006160  104405                           TRAP    C$ESEG
2468
2469                                            ;CHECK THAT RST H, MP H, AND CK H ARE 0 WHEN CTS H AND 8 BIT H ARE 1
2470                                            ;RDV H AND WRV H ARE IGNORED
2471
2472  006162                                    BGNSEG
2473  006162  104404                           TRAP    C$BSEG
2474  006164  112737  000012  002252           MOVB    #CTSH!BIT8H,ROLOAD      ;SETUP BITS TO LOAD
2475  006172  052737  000060  002256           BIS     #RDVH.WRVH,ROMASK       ;SETUP TO IGNORE RDV AND WRV
2476  006200  004737  005344                    JSR     PC,LDRDRO               ;GO LOAD, READ AND COMPARE REG 0
2477  006204  001404                           BEQ     4$                      ;IF OK THEN CONTINUE
2478  006206                                    ERRDF   1,,ROEROR               ;A BIT IS SHORTED TO ANOTHE
2479  006206  104455                           TRAP    C$ERDF
2480  006210  000001                           .WORD   1
2481  006212  000000                           .WORD   0
2482  006214  003622                           .WORD   ROEROR
2483  006216                              4$:   ENDSEG
2484  006216                              10001$:
2485  006216  104405                           TRAP    C$ESEG
2486
2487  006220                                    ENDSUB
2488  006220                              L10025:
2489  006220  104403                           TRAP    C$ESUB
2490
2491  006222                                    ENDTST
2492  006222                              L10023:
2493  006222  104401                           TRAP    C$ETST
2494
```

```
                                        .SBTTL TEST 3: CONTROL REG 0 TEST USING A BINARY COUNT

                                        ;++
                                        ; TEST TO CHECK CONTROL REGISTER 0.  THIS TEST WILL CHECK THE SIGNALS RST H,
                                        ; CTS H, MP H, 8 BIT H, AND CK H USING A BINARY COUNT PATTERN.  THE READ ONLY
                                        ; BITS RDV H AND WRV H WILL BE CHECKED FOR A 0 WHEN THE SIGNAL RST H IS SET,
                                        ; OTHERWISE, THE TWO SIGNALS WILL BE IGNORED.
                                        ;--

006224                                          BGNTST
006224                                  T3::
006224  004737  004546                          JSR     PC,INITMS               ;SELECT MS AND INITIALIZE IT
006230  105037  002252                          CLRB    ROLOAD                  ;SET LOW BYTE OF REG 0 TO 0
006234                                  1$:     BGNSEG
006234  104404                                  TRAP    C$BSEG
006236  052737  000060  002256                  BIS     #RDVH.WRVH,ROMASK       ;SETUP TO IGNORE RDV H AND WRV H
006244  032737  000001  002252                  BIT     #RSTH,ROLOAD            ;CHECK IF SIGNAL RST H WILL BE SET
006252  001403                                  BEQ     2$                      ;IF NOT THEN IGNORE RDV H AND WRV H
006254  042737  000060  002256                  BIC     #RDVH!WRVH,ROMASK       ;SETUP TO CHECK RDV H AND WRV H
006262  004737  005344                  2$:     JSR     PC,LDRDRO               ;GO LOAD, READ AND CHECK REGISTER 0
006266  001404                                  BEQ     3$                      ;IF LOADED OK THEN CONTINUE
006270                                          ERRDF   1,,ROEROR               ;REGISTER 0 NOT EQUAL EXPECTED
006270  104455                                  TRAP    C$ERDF
006272  000001                                  .WORD   1
006274  000000                                  .WORD   0
006276  003622                                  .WORD   ROEROR
006300                                  3$:     ENDSEG
006300                                  10000$:
006300  104405                                  TRAP    C$ESEG

006302  005237  002252                          INC     ROLOAD                  ;UPDATE TEST PATTERN BY 1
006306  032737  000020  002252                  BIT     #BIT4,ROLOAD            ;CHECK IF LOW R/W BITS DONE
006314  001747                                  BEQ     1$                      ;IF NOT LOAD NEXT BIT
006316  032737  000100  002252                  BIT     #CKH,ROLOAD             ;CHECK IF 2ND PORTION DONE
006324  001004                                  BNE     4$                      ;IF YES THEN END OF TEST
006326  112737  000100  002252                  MOVB    #CKH,ROLOAD             ;SETUP TO DO 2ND PORTION OF TEST
006334  000737                                  BR      1$                      ;DO BINARY COUNT AGAIN WITH CK H SET
006336                                  4$:     ENDTST
006336                                  L10026:
006336  104401                                  TRAP    C$ETST

```

TEST 4: CONTROL REG 2 TEST (1'S, 0'S, 1'S + 0'S, 0'S + 1'S) SEQ 0050

```
                                .SBTTL TEST 4: CONTROL REG 2 TEST (1'S, 0'S, 1'S + 0'S, 0'S + 1'S)

                                ;++
                                ; TEST TO CHECK CONTROL REGISTER 2 READ/WRITE BITS.  THIS TEST WILL CHECK THAT
                                ; THE SIGNALS MSAD16 H, MSAD17 H, MSEL0 H AND MSEL1 H CAN BE SET AND CLEARED
                                ; IN REGISTER 2.  THE READ ONLY SIGNALS ESR H, WREN H, AND MSBRK H ARE IG-
                                ; NORED DURING THIS TEST ALONG WITH OTHER UNDEFINED BITS.  THIS IS A BASIC
                                ; TEST OF THE READ/WRITE BITS.  THE TEST PATTERNS USED ARE AS FOLLOWS:
                                ;    1. SET ALL R/W BITS TO A 1 AND THEN A 0
                                ;    2. SET R/W BITS TO ALTERNATING 1'S + 0'S AND THEN 0'S + 1'S
                                ;--

006340                                  BGNTST
006340                          T4::
006340  004737  004546                  JSR     PC,INITMS               ;SELECT AND INIT MEM SIM
006344  012737  177740  002270          MOV     #177740,R2MASK          ;SETUP REG 2 MASK WORD

006352                                  BGNSUB
006352                          T4.1:
006352  104402                          TRAP    C$BSUB

                                ;CHECK THAT MSAD16 H, MSAD17 H, MSEL0 H, AND MSEL1 H CAN BE SET TO 1.
                                ;THE REMAINING BITS ARE IGNORED BY THIS TEST

006354                                  BGNSEG
006354  104404                          TRAP    C$BSEG
006356  012737  000017  002264          MOV     #MSAD16!MSAD17!MSEL0!MSEL1, LOAD ;SETUP BITS TO LOAD
006364  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND COMPARE REG 2
006370  001404                          BEQ     1$                      ;IF ALL ONES THEN CONTINUE
006372                                  ERRDF   2,,R2EROR               ;A BIT(S) FAILED TO SET
006372  104455                          TRAP    C$ERDF
006374  000002                          .WORD   2
006376  000000                          .WORD   0
006400  003710                          .WORD   R2EROR
006402                          1$:     ENDSEG
006402                          10000$:
006402  104405                          TRAP    C$ESEG

                                ;CHECK THAT MSAD16 H, MSAD17 H, MSEL0 H, MSEL1 H CAN BE CLEARED. THE
                                ;REMAINING BITS ARE IGNORED DURING THIS TEST.

006404                                  BGNSEG
006404  104404                          TRAP    C$BSEG
006406  005037  002264                  CLR     R2LOAD                  ;SETUP BITS TO LOAD TO 0
006412  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND COMPARE REG 2
006416  001404                          BEQ     2$                      ;IF ALL ZERO THEN CONT
006420                                  ERRDF   2,,R2EROR               ;A BIT(S) FAILED TO ZERO
006420  104455                          TRAP    C$ERDF
006422  000002                          .WORD   2
006424  000000                          .WORD   0
006426  003710                          .WORD   R2EROR
006430                          2$:     ENDSEG
006430                          10001$:
006430  104405                          TRAP    C$ESEG
006432                                  ENDSUB
006432                          L10030:
```

```
2592  006432  104403                          TRAP    C$ESUB
2593
2594  006434                                  BGNSUB
2595  006434                            T4.2:
2596  006434  104402                          TRAP    C$BSUB
2597
2598                                          ;CHECK THAT ALTERNATING BITS IN REG 2 CAN BE SET AND CLEARED.
2599                                          ;
2600                                          ;CHECK THAT MSAD17 H AND MSEL1 H ARE 0 WHEN MSAD16 H AND MSEL0 H
2601                                          ;ARE SET TO A ONE.
2602
2603  006436                                  BGNSEG
2604  006436  104404                          TRAP    C$BSEG
2605  006440  012737  000005  002264          MOV     #MSAD16!MSEL0,R2LOAD    ;SETUP BITS TO LOAD INTO REG 2
2606  006446  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ, AND COMPARE REG 2
2607  006452  001404                          BEQ     1$                      ;IF OK THEN CONTINUE
2608  006454                                  ERRDF   2,,R2EROR               ;ALTERNATING 1'S AND 0'S FAILED
2609  006454  104455                          TRAP    C$ERDF
2610  006456  000002                          .WORD   2
2611  006460  000000                          .WORD   0
2612  006462  003710                          .WORD   R2EROR
2613  006464                            1$:   ENDSEG
2614  006464                            10000$:
2615  006464  104405                          TRAP    C$ESEG
2616
2617                                          ;CHECK THAT MSAD16 H AND MSEL0 H ARE 0 WHEN MSAD17 H AND MSEL1 H
2618                                          ;ARE SET TO A ONE
2619
2620  006466                                  BGNSEG
2621  006466  104404                          TRAP    C$BSEG
2622  006470  012737  000012  002264          MOV     #MSAD17!MSEL1,R2LOAD    ;SETUP BITS TO LOAD
2623  006476  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ, AND COMPARE REG 2
2624  006502  001404                          BEQ     2$                      ;IF EQUAL THEN CONT
2625  006504                                  ERRDF   2,,R2EROR               ;ALTERNATING 0'S AND 1'S
2626  006504  104455                          TRAP    C$ERDF
2627  006506  000002                          .WORD   2
2628  006510  000000                          .WORD   0
2629  006512  003710                          .WORD   R2EROR
2630  006514                            2$:   ENDSEG
2631  006514                            10001$:
2632  006514  104405                          TRAP    C$ESEG
2633  006516                                  ENDSUB
2634  006516                            L10031:
2635  006516  104403                          TRAP    C$ESUB
2636  006520                                  ENDTST
2637  006520                            L10027:
2638  006520  104401                          TRAP    C$ETST
2639
```

```
                                        .SBTTL TEST 5: CONTROL REG 2 TEST USING A BINARY COUNT

                                        ;++
                                        ; TEST TO CHECK CONTROL REGISTER 2 READ/WRITE BITS.  THIS TEST WILL CHECK THE
                                        ; SIGNALS MSAD16 H, MSAD17 H, MSEL0 H, AND MSEL2 H USING A BINARY COUNT PATTERN.
                                        ; THE READ ONLY BITS (ESR H, WREN H, AND MSBRK H) AND THE UNUSED BITS WILL BE
                                        ; IGNORED DURING THIS TEST.
                                        ;--

006522                                          BGNTST
006522                                  T5::
006522  004737  004546                          JSR     PC,INITMS               ;SELECT AND INIT MEMORY SIM
006526  012701  000020                          MOV     #20,R1                  ;SETUP TEST COUNTER
006532  005037  002264                          CLR     R2LOAD                  ;SET PATTERN INITIALLY TO 0
006536  012737  177740  002270                  MOV     #177740,R2MASK          ;SETUP REG 2 MASK WORD
006544                                  1$:     BGNSEG
006544  104404                                  TRAP    C$BSEG
006546  004737  005412                          JSR     PC,LDRDR2               ;GO LOAD, READ AND COMPARE REG 2
006552  001404                                  BEQ     2$                      ;IF COMPARED OK THEN CONT
006554                                          ERRDF   2,,R2EROR               ;DATA LOADED NOT EQL EXPECTED
006554  104455                                  TRAP    C$ERDF
006556  000002                                  .WORD   2
006560  000000                                  .WORD   0
006562  003710                                  .WORD   R2EROR
006564                                  2$:     ENDSEG
006564                                  10000$:
006564  104405                                  TRAP    C$ESEG
006566  005237  002264                          INC     R2LOAD                  ;UPDATE THE TEST PATTERN
006572  005301                                  DEC     R1                      ;DECREMENT THE TEST COUNTER
006574  001363                                  BNE     1$                      ;IF NOT 0 THEN DO NEXT PATTERN
006576                                          ENDTST
006576                                  L10032:
006576  104401                                  TRAP    C$ETST
```

TEST 6: CONTROL REG 4 TEST (1'S AND THEN 0'S) SEQ 0053

```
                                .SBTTL TEST 6: CONTROL REG 4 TEST (1'S AND THEN 0'S)

                                ;++
                                ; TEST TO CHECK CONTROL REGISTER 4 BY LOADING ALL ONES INTO REGISTER 4 AND
                                ; CHECKING THAT ALL ONES WERE LOADED INTO REGISTER 4 BY READING REGISTER 4.
                                ; THE TEST WILL THEN LOAD ALL ZEROES INTO REGISTER 4 AND CHECK THAT ALL
                                ; ZEROES WERE LOADED BY READING THE REGISTER.
                                ;--

006600                                  BGNTST
006600                          T6::
006600  004737  004546                   JSR     PC,INITMS               ;SELECT AND INIT MEM SIMULATOR

                                        ;SET MSAD BITS 15 THROUGH 0 TO A 1

006604                                  BGNSEG
006604  104404                          TRAP    C$BSEG
006606  012737  177777  002276          MOV     #-1,R4LOAD              ;SETUP TO LOAD ALL ONES INTO REG 4
006614  004737  005460                   JSR     PC,LDRDR4               ;LOAD, READ AND COMPARE REG 4
006620  001404                          BEQ     1$                      ;IF ALL ONES THEN CONT
006622                                  ERRDF   3,,R4EROR               ;FAILED TO LOAD ALL ONES
006622  104455                          TRAP    C$ERDF
006624  000003                          .WORD   3
006626  000000                          .WORD   0
006630  003776                          .WORD   R4EROR
006632                          1$:     ENDSEG
006632                          10000$:
006632  104405                          TRAP    C$ESEG

                                        ;SET MSAD BITS 15 THROUGH 0 TO A 0

006634                                  BGNSEG
006634  104404                          TRAP    C$BSEG
006636  005037  002276                  CLR     R4LOAD                  ;SETUP TO LOAD ALL ZEROES INTO REG 4
006642  004737  005460                   JSR     PC,LDRDR4               ;LOAD, READ AND COMPARE REG 4
006646  001404                          BEQ     2$                      ;IF ALL ZEORES THEN CONT
006650                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT ALL 0'S
006650  104455                          TRAP    C$ERDF
006652  000003                          .WORD   3
006654  000000                          .WORD   0
006656  003776                          .WORD   R4EROR
006660                          2$:     ENDSEG
006660                          10001$:
006660  104405                          TRAP    C$ESEG
006662                                  ENDTST
006662                          L10033:
006662  104401                          TRAP    C$ETST
```

TEST 7: CONTROL REG 4 TEST (1'S + 0'S, AND 0'S + 1'S) SEQ 0054

```
                                .SBTTL TEST 7: CONTROL REG 4 TEST (1'S + 0'S, AND 0'S + 1'S)

                                ;++
                                ; TEST TO CHECK CONTROL REGISTER 4 BY LOADING ALTERNATING ONES AND ZEROES AND
                                ; THEN ZEROES AND ONES.  CHECKS THAT ADJACENT BITS ARE NOT SHORED TOGETHER.
                                ;--

006664                                  BGNTST
006664                          T7::
006664  004737  004546                  JSR     PC,INITMS               ;SELECT AND INIT MS

                                        ;LOAD REGISTER 4 WITH 125252 PATTERN

006670                                  BGNSEG
006670  104404                          TRAP    C$BSEG
006672  012737  125252  002276          MOV     #125252,R4LOAD          ;SETUP PATTERN TO LOAD
006700  004737  005460                  JSR     PC,LDRDR4               ;LOAD, READ AND COMPARE REGISTER 4
006704  001404                          BEQ     1$                      ;IF PATTERN OK THEN CONTINUE
006706                                  ERRDF   3,,R4EROR               ;PATTERN READ NEQ 125252
006706  104455                          TRAP    C$ERDF
006710  000003                          .WORD   3
006712  000000                          .WORD   0
006714  003776                          .WORD   R4EROR
006716                          1$:     ENDSEG
006716                          10000$:
006716  104405                          TRAP    C$ESEG

                                        ;LOAD REGISTER 4 WITH 052525 PATTERN

006720                                  BGNSEG
006720  104404                          TRAP    C$BSEG
006722  012737  052525  002276          MOV     #052525,R4LOAD          ;SETUP PATTERN TO LOAD
006730  004737  005460                  JSR     PC,LDRDR4               ;LOAD, READ AND COMPARE REGISTER 4
006734  001404                          BEQ     2$                      ;IF PATTERN OK THEN CONT
006736                                  ERRDF   3,,R4EROR               ;PATTERN READ NEQ 052525
006736  104455                          TRAP    C$ERDF
006740  000003                          .WORD   3
006742  000000                          .WORD   0
006744  003776                          .WORD   R4EROR
006746                          2$:     ENDSEG
006746                          10001$:
006746  104405                          TRAP    C$ESEG
006750                                  ENDTST
006750                          L10034:
006750  104401                          TRAP    C$ETST
```

TEST 8: CONTROL REG 4 TEST ON LOW BYTE USING BINARY COUNT SEQ 0055

```
                        .SBTTL TEST 8: CONTROL REG 4 TEST ON LOW BYTE USING BINARY COUNT

                        ;++
                        ; TEST TO CHECK LOW BYTE OF CONTROL REGISTER 4 USING A BINARY COUNT PATTERN
                        ;--

                                BGNTST
                        T8::
006752  004737  004546          JSR     PC,INITMS               ;SELECT AND INIT MEM SIM
006756  005037  002276          CLR     R4LOAD                  ;SET PATTERN INITIALLY TO 0
006762                  1$:     BGNSEG
006762  104404                  TRAP    C$BSEG
006764  004737  005460          JSR     PC,LDRDR4               ;LOAD, READ AND COMPARE PATTERN
006770  001404                  BEQ     2$                      ;IF PATTERN OK THEN CONT
006772                          ERRDF   3,,R4EROR               ;PATTERN LOADED NEQ PATTERN READ
006772  104455                  TRAP    C$ERDF
006774  000003                  .WORD   3
006776  000000                  .WORD   0
007000  003776                  .WORD   R4EROR
007002                  2$:     ENDSEG
007002                  10000$:
007002  104405                  TRAP    C$ESEG
007004  005237  002276          INC     R4LOAD                  ;UPDATE THE PATTERN BY 1
007010  032737  000400  002276  BIT     #BIT8,R4LOAD            ;CHECK IF DONE
007016  001761                  BEQ     1$                      ;IF NOT THEN LOAD NEXT PATTERN
007020                          ENDTST
007020                  L10035:
007020  104401                  TRAP    C$ETST
```

```
                                        .SBTTL TEST 9: CONTROL REG 4 TEST HIGH BYTE USING BINARY COUNT

                                        ;++
                                        ; TEST TO CHECK HIGH BYTE OF CONTROL REGISTER 4 USING A BINARY COUNT PATTERN
                                        ;--

007022                                          BGNTST
007022                                  T9::
007022  004737  004546                          JSR     PC,INITMS               ;SELECT AND INIT MEM SIM
007026  005037  002276                          CLR     R4LOAD                  ;SET PATTERN INITIALLY TO 0
007032                                  1$:     BGNSEG
007032  104404                                  TRAP    C$BSEG
007034  004737  005460                          JSR     PC,LDRDR4               ;LOAD, READ AND COMPARE PATTERN
007040  001404                                  BEQ     2$                      ;IF PATTERN OK THEN CONT
007042                                          ERRDF   3,,R4EROR               ;PATTERN LOADED NEQ PATTERN READ
007042  104455                                  TRAP    C$ERDF
007044  000003                                  .WORD   3
007046  000000                                  .WORD   0
007050  003776                                  .WORD   R4EROR
007052                                  2$:     ENDSEG
007052                                  10000$:
007052  104405                                  TRAP    C$ESEG
007054  062737  000400  002276                  ADD     #BIT8,R4LOAD            ;UPDATE THE DATA PATTERN
007062  001363                                  BNE     1$                      ;IF NOT 0 THEN GO LOAD DATA PATTERN
007064                                          ENDTST
007064                                  L10036:
007064  104401                                  TRAP    C$ETST
```

```
                                .SBTTL TEST 10: CHECK SIGNAL MSBRK H TO BE 0 IN CONTROL REG 2.

                                ;++
                                ; TEST TO CHECK THAT THE SIGNAL MSBRK H IN CONTROL REGISTER 2 CAN BE SET
                                ; TO A 0 WHEN THE SIGNALS RST H AND MP H ARE ASSERTED IN CONTROL REGISTER 0.
                                ; SETTING THE SIGNAL RST H WILL PRESET THE RDV AND WRV FLIP-FLOP'S SUCH
                                ; THAT THE SIGNAL BRK L WILL BE HIGH.  THE SIGNAL MP H WILL ALLOW THE
                                ; SIGNAL BRK L AS A HIGH TO BE INVERTED, GENERATING THE SIGNAL MSBRK H AS
                                ; A LOW OR ZERO.  THE TEST WILL THEN LOAD AND READ CONTROL REGISTER 2 WITH
                                ; ZEROES CHECKING THE SIGNAL MSBRK H TO BE A 0.
                                ;--

007066                                  BGNTST
007066                          T10::
007066  004737  004546                  JSR     PC,INITMS               ;SELECT AND INIT THE MEMORY SIMULATOR
007072                                  BGNSEG
007072  104404                          TRAP    C$BSEG

                                ;SET THE SIGNAL RST H AND MP H TO A 1 IN CONTROL REGISTER 0.  CHECK
                                ;THE SIGNALS RDV AND WRV TO BE A 0 AS THE RESULT OF RST H BEING SET.

007074  052737  000005  002252          BIS     #RSTH.MPH,ROLOAD        ;SETUP TO SET RST H AND MP H TO 1
007102  042737  000060  002256          BIC     #RDVH!WRVH,ROMASK       ;SETUP TO CHECK RDV AND WRV BITS
007110  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REG 0
007114  001405                          BEQ     1$                      ;IF OK THEN CONTINUE
007116                                  ERRDF   1,,R0EROR               ;REGISTER 0 NOT EQULA EXPECTED
007116  104455                          TRAP    C$ERDF
007120  000001                          .WORD   1
007122  000000                          .WORD   0
007124  003622                          .WORD   R0EROR
007126                                  CKLOOP
007126  104406                          TRAP    C$CLP1

                                ;LOAD ALL ZEROES INTO CONTROL REGISTER 2 AND CHECK THAT ALL ZEROES
                                ;WERE LOADED AND THAT THE SIGNAL MSBRK H IS A 0.

007130  005037  002264          1$:     CLR     R2LOAD                  ;SETUP TO LOAD ALL ZEROES
007134  012737  177540  002270          MOV     #177540,R2MASK          ;SETUP REG 2 MASK WORD
007142  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REG 2
007146  001404                          BEQ     2$                      ;IF REG 2 OK THEN CONT
007150                                  ERRDF   2,,R2EROR               ;REG 2 NOT EQUAL EXPECTED
007150  104455                          TRAP    C$ERDF
007152  000002                          .WORD   2
007154  000000                          .WORD   0
007156  003710                          .WORD   R2EROR
007160                          2$:     ENDSEG
007160                          10000$:
007160  104405                          TRAP    C$ESEG
007162                                  ENDTST
007162                          L10037:
007162  104401                          TRAP    C$ETST
```

```
                                .SBTTL TEST 11: CHECK 1K BY 4 MAP PROTECT RAM WITH 1'S AND THEN 0'S

                                ;++
                                ; TEST TO CHECK THE 1K BY 4 MAP PROTECT RAM WITH A PATTERN OF ALL ONES AND
                                ; THEN ALL ZEROES.  THE TEST WILL SELECT AND INITIALIZE THE MEMORY SIMULATOR.
                                ; THE TEST WILL SET THE SIGNAL MSEL1 H IN CONTROL REGISTER 2 WHICH WILL
                                ; SET THE SIGNAL SMPM L WHEN A WRITE TO REGISTER 6 IS ISSUED.  THE TEST WILL
                                ; THEN SELECT THE MEMORY SIMULATOR ADDRESS BY LOADING THE ADDRESS INTO
                                ; CONTROL REGISTER 4 BITS 15-8 AND CONTROL REGISTER 2 BITS 1-0.  THE TEST
                                ; WILL THEN WRITE ALL ONES INTO THE MAP PROTECTION RAM VIA REGISTER 6 AND
                                ; THEN READ THE RAM LOCATION BACK VIA CONTROL REGISTER 6.  THE RAM BITS
                                ; MUTB H, MPIN H, WRE H, AND RDE H WILL BE CHECKED FOR ALL ONES. THE TEST
                                ; WILL THEN WRITE, READ AND CHECK THE LOCATION FOR ALL ZEROES. THE TEST
                                ; WILL THEN SEQUENCE TO THE NEXT ADDRESS AND REPEAT THE SAME TEST PATTERNS
                                ; UNTIL ALL ADDRESSES HAVE BEEN CHECKED.
                                ;--

007164                                  BGNTST
007164                          T11::
007164  004737  004546                  JSR     PC,INITMS               ;SELECT AND INIT MEMORY SIMULATOR
007170  005001                          CLR     R1                      ;CLEAR BITS 15-8 MSAD ADDRESS COUNTER
007172  005002                          CLR     R2                      ;CLEAR BITS 17-16 MSAD ADDRESS COUNTER

007174                          1$:     BGNSEG
007174  104404                          TRAP    C$BSEG

                                ;SET SIGNAL MSEL1 H TO A 1 AND MSAD BITS 17 OR 16 TO A 1 OR 0 IN
                                ;CONTROL REGISTER 2

007176  012737  000010  002264          MOV     #MSEL1,R2LOAD           ;SETUP TO SET MSEL1 BIT TO A 1
007204  050237  002264                  BIS     R2,R2LOAD               ;SETUP MSAD BITS 17 AND 16
007210  012737  177740  002270          MOV     #177740,R2MASK          ;SETUP TO IGNORE REG 2 BITS 15-5
007216  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND COMPARE REG 2
007222  001405                          BEQ     2$                      ;IF LOADED OK THEN CONTINUE
007224                                  ERRDF   2,,R2EROR               ;REGISTER 2 FAILED TO LOAD CORRECTLY
007224  104455                          TRAP    C$ERDF
007226  000002                          .WORD   2
007230  000000                          .WORD   0
007232  003710                          .WORD   R2EROR
007234                                  CKLOOP
007234  104406                          TRAP    C$CLP1

                                ;LOAD MSAD BITS 15-8 INTO CONTROL REGISTER 4

007236  010137  002276          2$:     MOV     R1,R4LOAD               ;SETUP BITS TO LOAD
007242  004737  005460                  JSR     PC,LDRDR4               ;GO LOAD, READ AND COMPARE REG 4
007246  001405                          BEQ     3$                      ;IF REGISTER 4 CORRECT THEN CONT
007250                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
007250  104455                          TRAP    C$ERDF
007252  000003                          .WORD   3
007254  000000                          .WORD   0
007256  003776                          .WORD   R4EROR
007260                                  CKLOOP
007260  104406                          TRAP    C$CLP1

                                ;SET BITS MUTB H, MPINH, WRE H, RDE H TO A 1 IN LOCATION ADDRESSED
```

```
                                                ;BY CONTROL REGISTER 2 AND 4. THE SIGNAL SMPM L WILL BE ASSERTED
                                                ;ON THE WRITE/READ TO REGISTER 6

007262  012737  177760  002306  3$:   MOV     #177760,R6MASK          ;SETUP REG 6 MASK WORD
007270  012737  000017  002302        MOV     #MUTBH!MPINH!WREH!RDEH,R6LOAD
007276  004737  005504                JSR     PC,LDRDR6               ;GO LOAD, READ AND COMPARE REGISTER 6
007302  001405                        BEQ     4$                      ;IF DATA OK THEN CONTINUE
007304                                ERRDF   4,MSGMP,ALINFO          ;MAP PROTECTION RAM DATA ERROR
007304  104455                        TRAP    C$ERDF
007306  000004                        .WORD   4
007310  002354                        .WORD   MSGMP
007312  004050                        .WORD   ALINFO
007314                                CKLOOP
007314  104406                        TRAP    C$CLP1

                                                ;WRITE THE SAME LOCATION WITH ALL THE BITS SET TO A 0

007316  005037  002302          4$:   CLR     R6LOAD                  ;SETUP ALL BITS TO BE LOADED AS A 0
007322  004737  005504                JSR     PC,LDRDR6               ;GO LOAD,READ AND COMPARE REG 6
007326  001404                        BEQ     5$                      ;IF OK THEN CONTINUE
007330                                ERRDF   4,MSGMP,ALINFO          ;DATA ERROR MAP PROTECTION RAM
007330  104455                        TRAP    C$ERDF
007332  000004                        .WORD   4
007334  002354                        .WORD   MSGMP
007336  004050                        .WORD   ALINFO
007340                          5$:   ENDSEG
007340                          '0000$:
007340  104405                        TRAP    C$ESEG

007342  062701  000400                ADD     #MSAD8,R1               ;UPDATE MSAD BITS 15-8 BY 1
007346  001312                        BNE     1$                      ;IF NOT 0 THEN DO NEXT ADDRESS
007350  062702  000001                ADD     #MSAD16,R2              ;UPDATE MSAD BITS 16 AND 17
007354  032702  000004                BIT     #MSEL0,R2               ;CHECK IF DONE
007360  001705                        BEQ     1$                      ;IF NOT DONE DO RANGE OF ADDRESSES
007362                                ENDTST
007362                          L10040:
007362  104401                        TRAP    C$ETST
```

```
                        .SBTTL TEST 12: CHECK 1K BY 4 MAP PROTECT RAM (1'S + 0'S, AND 0'S + 1'S).

                        ;++
                        ; TEST TO CHECK THE 1K BY 4 MAP PROTECT RAM USING AN ALTERNATING ONES AND
                        ; ZEROES PATTERN AND AN ALTERNATING ZEROES AND ONES PATTERN.  THE TEST WILL
                        ; ALSO CHECK THAT THE MAP PROTECT BITS MPIN H AND WRE H CAN BE READ BACK
                        ; INTO CONTROL REGISTER 2 AS SIGNALS ESR H AND WREN H RESPECTIVELY.  THE
                        ; SIGNAL MSBRK H IN CONTROL REGISTER 2 WILL BE CHECKED FOR A 0.
                        ;
                        ; THE TEST WILL SELECT AND INITIALIZE THE MEMORY SIMULATOR.  THE FOLLOWING
                        ; SECTION WILL BE REPEATED FOR EACH ADDRESS OF THE 1K BY 4 MAP PROTECT RAM.
                        ;
                        ; THE TEST WILL SET THE SIGNALS RST H AND MP H IN CONTROL REGISTER 0.  THE
                        ; THE TEST WILL THEN READ REGISTER 0 AND CHECK THAT SIGNALS CTS H, 8 BIT H,
                        ; RDV H AND WRV H ARE 0 AND THAT RST H AND MP H ARE A ONE.  THE TEST WILL
                        ; THEN CLEAR THE SIGNAL RST H AND CHECK THAT RST H WAS THE ONLY BIT THAT
                        ; CHANGED IN CONTROL REGISTER 0.
                        ;
                        ; THE TEST WILL THEN SET THE SIGNAL MSEL1 H TO A ONE IN CONTROL REGISTER 2
                        ; ALONG WITH THE BITS MSAD17 H AND MSAD 16 H. THE TWO ADDRESS BITS MSAD17
                        ; AND MSAD16 WILL BE SET TO A ONE OR 0 DEPENDING UPON THE ADDRESS TO BE
                        ; TESTED.  THE TEST WILL THEN READ CONTROL REGISTER 2 AND CHECK THAT THE
                        ; BITS WERE LOADED CORRECTLY. THE SIGNAL MSBRK H WILL BE CHECKED FOR A 0.
                        ; THE SIGNAL MSBRK H IS READ BACK AS A RESULT OF MP H BEING SET AND THE
                        ; SIGNAL RST H CLEARING THE RDV AND WRV FLIP-FLOPS.  THE SIGNAL MSEL1 H
                        ; BEING ASSERTED ON A READ OR WRITE TO CONTROL REGISTER 6 WILL CAUSE THE
                        ; SIGNAL SMPM L TO BE ASSERTED WHICH ENABLES THE MAP PROTECT RAM TO BE
                        ; WRITTEN OR READ.
                        ;
                        ; THE TEST WILL NOW SELECT THE REMAINING PART OF THE 18 BIT MEMORY
                        ; SIMULATOR ADDRESS BY LOADING THE ADDRESS INTO CONTROL REGISTER 4 BITS
                        ; 15-8, AND CHECKING THAT THE ADDRESSS LOADED CORRECTLY BY READING BACK
                        ; REGISTER 2.
                        ;
                        ; THE TEST WILL THEN WRITE THE LOCATION SELECTED WITH ONES IN BITS MPIN H
                        ; AND RDE H AND ZEROES IN BITS WRE H AND MUTB H.  THIS IS DONE VIA WRITING
                        ; TO CONTROL REGISTER 6 WHICH WILL ASSERT THE SIGNAL SMPM L.  THE SIGNAL
                        ; SMPM L ENABLES THE MAP PROTECT RAM TO BE WRITTEN OR READ.  THE PROGRAM
                        ; WILL READ BACK THE BITS LOADED BY READING BACK CONTROL REGISTER 6. THE
                        ; LOCATION WILL BE COMPARED FOR CORRECT DATA.
                        ;
                        ; THE TEST WILL THEN READ CONTROL REGISTER 2 AND CHECK THAT BIT ESR H
                        ; IS SET TO A ONE AND THAT BIT WREN H IS SET TO A 0. THESE BITS ARE
                        ; ENABLED BY THE SIGNAL MP H.
                        ;
                        ; THE TEST WILL THEN WRITE ONES IN BITS WRE H AND MUTB H, AND ZEROES INTO
                        ; BITS MPIN H AND RDE H.  THE TEST WILL THEN READ AND COMPARE THE LOCATION
                        ; FOR THE CORRECT CONTENTS.
                        ;
                        ; THE TEST WILL THEN READ CONTROL REGISTER 2 AND CHECK THAT THE BIT WREN H
                        ; IS SET TO A ONE AND BIT ESR H IS SET TO A 0.
                        ;
                        ; THE TEST WILL THEN BE REPEATED UNTIL ALL ADDRESSES HAVE BEEN VERIFIED.
                        ;--

007364                          BGNTST
```

```
3027  007364                          T12::
3028  007364  004737  004546                  JSR     PC,INITMS               ;SELECT AND INIT MEMORY SIMULATOR
3029  007370  005001                          CLR     R1                      ;CLEAR BITS 15-8 MSAD ADDRESS COUNTER
3030  007372  005002                          CLR     R2                      ;CLEAR BITS 17-16 MSAD ADDRESS COUNTER
3031
3032  007374                          1$:     BGNSEG
3033  007374  104404                          TRAP    C$BSEG
3034
3035                                          ;SET SIGNALS RST H AND MP H IN CONTROL REGISTER 0.  THE SIGNAL
3036                                          ;RST H WILL PRESET THE RDV AND WRV FLIP-FLOPS.  THE SIGNAL MP H
3037                                          ;WILL ALLOW THE MAP PROTECTION RAM BITS MPIN H AND WRE H TO BE
3038                                          ;READ INTO CONTROL REGISTER 2 AS ESR H AND WREN H RESPECTIVELY.
3039                                          ;THESE BITS WILL NOT BE CHECKED UNTIL THE RAM LOCATION HAS BEEN
3040                                          ;WRITTEN AND TESTED.
3041
3042  007376  052737  000005  002252          BIS     #RSTH!MPH,ROLOAD        ;SETUP TO SET RST H AND MP H
3043  007404  042737  000060  002256          BIC     #RDVH.WRVH,ROMASK       ;SETUP TO CHECK RDV AND WRV FOR A 0
3044  007412  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REG 0
3045  007416  001405                          BEQ     2$                      ;IF OK THEN GO CLEAR SIGNAL RST H
3046  007420                                  ERRDF   1,,ROEROR               ;REGISTER 0 NOT EQULA EXPECTED
3047  007420  104455                          TRAP    C$ERDF
3048  007422  000001                          .WORD   1
3049  007424  000000                          .WORD   0
3050  007426  003622                          .WORD   ROEROR
3051  007430                                  CKLOOP
3052  007430  104406                          TRAP    C$CLP1
3053
3054                                          ;CLEAR BIT RST H IN CONTORL REGISTER 0. RDV AND WRV SHOULD NOT
3055                                          ;CHANGE DURING THIS TEST.
3056
3057  007432  042737  000001  002252  2$:     BIC     #RSTH,ROLOAD            ;SETUP TO CLEAR BIT RST H
3058  007440  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REG 0
3059  007444  001405                          BEQ     3$                      ;IF OK THEN CONTINUE
3060  007446                                  ERRDF   1,,ROEROR               ;REGISTER 0 NOT EQUAL EXPECTED
3061  007446  104455                          TRAP    C$ERDF
3062  007450  000001                          .WORD   1
3063  007452  000000                          .WORD   0
3064  007454  003622                          .WORD   ROEROR
3065  007456                                  CKLOOP
3066  007456  104406                          TRAP    C$CLP1
3067
3068                                          ;SET SIGNAL MSEL1 H TO A 1 AND MSAD BITS 17 OR 16 TO A 1 OR 0 IN
3069                                          ;CONTROL REGISTER 2
3070
3071  007460  012737  000010  002264  3$:     MOV     #MSEL1,R2LOAD           ;SETUP TO SET MSEL1 BIT TO A 1
3072  007466  050237  002264                  BIS     R2,R2LOAD               ;SETUP MSAD BITS 17 AND 16
3073  007472  012737  177540  002270          MOV     #177540,R2MASK          ;SETUP TO IGNORE REG 2 BITS 15-8 6-5
3074  007500  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND COMPARE REG 2
3075  007504  001405                          BEQ     4$                      ;IF LOADED OK THEN CONTINUE
3076  007506                                  ERRDF   2,,R2EROR               ;REGISTER 2 FAILED TO LOAD CORRECTLY
3077  007506  104455                          TRAP    C$ERDF
3078  007510  000002                          .WORD   2
3079  007512  000000                          .WORD   0
3080  007514  003710                          .WORD   R2EROR
3081  007516                                  CKLOOP
3082  007516  104406                          TRAP    C$CLP1
```

```
3083
3084                                        ;LOAD MSAD BITS 15-8 INTO CONTROL REGISTER 4
3085
3086  007520  010137  002276        4$:     MOV     R1,R4LOAD               ;SETUP BITS TO LOAD
3087  007524  004737  005460                JSR     PC,LDRDR4               ;GO LOAD, READ AND COMPARE REG 4
3088  007530  001405                        BEQ     5$                      ;IF REGISTER 4 CORRECT THEN CONT
3089  007532                                ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
3090  007532  104455                        TRAP    C$ERDF
3091  007534  000003                        .WORD   3
3092  007536  000000                        .WORD   0
3093  007540  003776                        .WORD   R4EROR
3094  007542                                CKLOOP
3095  007542  104406                        TRAP    C$CLP1
3096
3097                                        ;SET BITS MPIN H AND RDE H TO A 1 - SET BITS WRE H AND MUTB H TO A 0.
3098                                        ;THESE BITS ARE SET IN LOCATION ADDRESSED BY CONTROL REGISTER 2 AND 4.
3099                                        ;THE SIGNAL SMPM L WILL BE ASSERTED ON A WRITE OR READ OF CONTROL
3100                                        ;REGISTER 6.
3101
3102  007544  012737  177760  002306  5$:     MOV     #177760,R6MASK          ;SETUP REG 6 MASK WORD
3103  007552  012737  000005  002302          MOV     #MPINH!RDEH,R6LOAD
3104  007560  004737  005504                JSR     PC,LDRDR6               ;GO LOAD, READ AND COMPARE REGISTER 6
3105  007564  001405                        BEQ     6$                      ;IF DATA OK THEN CONTINUE
3106  007566                                ERRDF   4,MSGMP,ALINFO          ;MAP PROTECTION RAM DATA ERROR
3107  007566  104455                        TRAP    C$ERDF
3108  007570  000004                        .WORD   4
3109  007572  002354                        .WORD   MSGMP
3110  007574  004050                        .WORD   ALINFO
3111  007576                                CKLOOP
3112  007576  104406                        TRAP    C$CLP1
3113
3114                                        ;SETUP TO READ REGISTER 2 WITH REGISTER 0 BIT MP H SET TO A 1 TO
3115                                        ;ALLOW MPIN H AND WRE H TO BE READ AS BITS ESR H AND WREN H
3116
3117  007600  042737  000140  002270  6$:     BIC     #ESRH!WRENH,R2MASK      ;SETUP TO CHECK ESR H AND WREN H
3118  007606  052737  000040  002266          BIS     #ESRH,R2GOOD            ;SETUP EXPECTED BIT TO BE SET (MPIN H)
3119  007614  004737  005426                JSR     PC,READR2               ;GO READ AND CHECK REGISTER 2
3120  007620  001405                        BEQ     7$                      ;IF BIT ESR H SET THEN CONTINUE
3121  007622                                ERRDF   2,MSGMPL,ALR2IN         ;REG 2 NOT EQUAL EXPECTED
3122  007622  104455                        TRAP    C$ERDF
3123  007624  000002                        .WORD   2
3124  007626  002461                        .WORD   MSGMPL
3125  007630  004124                        .WORD   ALR2IN
3126  007632                                CKLOOP
3127  007632  104406                        TRAP    C$CLP1
3128
3129                                        ;WRITE THE SAME LOCATION WITH WRE H AND MUTB H SET TO A 1 AND
3130                                        ;MPIN H AND RED H SSET TO A 0
3131
3132  007634  012737  000012  002302  7$:     MOV     #WREH!MUTBH,R6LOAD      ;SET WRE H AND MUTB H TO A 1
3133  007642  004737  005504                JSR     PC,LDRDR6               ;GO LOAD,READ AND COMPARE REG 6
3134  007646  001405                        BEQ     8$                      ;IF OK THEN CONTINUE
3135  007650                                ERRDF   4,MSGMP,ALINFO          ;DATA ERROR MAP PROTECTION RAM
3136  007650  104455                        TRAP    C$ERDF
3137  007652  000004                        .WORD   4
3138  007654  002354                        .WORD   MSGMP
```

```
3139  007656  004050                          .WORD   ALINFO
3140  007660                                  CKLOOP
3141  007660  104406                          TRAP    C$CLP1
3142
3143                                          ;SETUP TO CHECK THAT MAP PROTECTION BIT WRE H WILL SET THE SIGNAL
3144                                          ;WREN H IN CONTROL REGISTER 2 WHEN MP H BIT IS SET.
3145
3146  007662  042737  000140  002266  8$:     BIC     #ESRH!WRENH,R2GOOD      ;CLEAR BOTH EXPECTED BITS
3147  007670  052737  000100  002266          BIS     #WRENH,R2GOOD           ;SETUP TO EXPECT WREN H BIT TO BE SET
3148  007676  004737  005426                  JSR     PC,READR2               ;GO READ AND CHECK REGISTER 2
3149  007702  001404                          BEQ     9$                      ;IF OK THEN CONTINUE
3150  007704                                  ERRDF   2,MSGMPL,ALR2IN         ;REGISTER 2 NOT EQUAL EXPECTED
3151  007704  104455                          TRAP    C$ERDF
3152  007706  000002                          .WORD   2
3153  007710  002461                          .WORD   MSGMPL
3154  007712  004124                          .WORD   ALR2IN
3155  007714                          9$:     ENDSEG
3156  007714                          10000$:
3157  007714  104405                          TRAP    C$ESEG
3158
3159  007716  062701  000400                  ADD     #MSAD8,R1               ;UPDATE MSAD BITS 15-8 BY 1
3160  007722  001224                          BNE     1$                      ;IF NOT 0 THEN DO NEXT ADDRESS
3161  007724  062702  000001                  ADD     #MSAD16,R2              ;UPDATE MSAD BITS 16 AND 17
3162  007730  032702  000004                  BIT     #MSEL0,R2               ;CHECK IF DONE
3163  007734  001617                          BEQ     1$                      ;IF NOT DO NEXT RANGE OF ADDRESSES
3164  007736                                  ENDTST
3165  007736                          L10041:
3166  007736  104401                          TRAP    C$ETST
3167
```

TEST 13: CHECK 1K BY 4 MAP PROTECT RAM FOR ADDRESS SHORTS. SEQ 0064

```
                        .SBTTL TEST 13: CHECK 1K BY 4 MAP PROTECT RAM FOR ADDRESS SHORTS.

                        ;++
                        ; TEST TO CHECK THE 1K BY 4 MAP PROTECTION RAM FOR ADDRESS SHORTS.  THIS TEST
                        ; WILL CHECK THAT WRITING A LOCATION DOES NOT EFFECT THE CONTENTS OF ANOTHER
                        ; LOCATION IN THE MAP PROTECTION RAM.  THE TEST WILL FILL THE MAP PROTECTION
                        ; RAM WITH ALL ZEROES.  THE TEST WILL THEN RESET THE ADDRESS POINTER TO THE
                        ; FIRST ADDRESS, CHECK THE LOCATION TO BE ZERO, WRITE ONES INTO THE LOCATION,
                        ; AND THEN READ AND CHECK THE LOCATION FOR ONES.  THIS TEST IS REPEATED FOR EACH
                        ; ADDRESS OF THE MAP PROTECTION RAM.
                        ;--

                                BGNTST
                        T13::
004737  004546                  JSR     PC,INITMS               ;SELECT AND INIT THE MEMORY SIMULATOR
                                BGNSUB
                        T13.1:
104402                          TRAP    C$BSUB
005001                          CLR     R1                      ;CLEAR MSAD ADDRESS BITS 15-0
005002                          CLR     R2                      ;CLEAR MSAD ADDRESS BITS 17-16
                        1$:     BGNSEG
104404                          TRAP    C$BSEG
                                ;SET SIGNAL MSEL1 H TO A 1 AND MSAD BITS 17 AND/OR 16 TO A 1 OR 0 IN
                                ;CONTROL REGISTER 2

012737  000010  002264          MOV     #MSEL1,R2LOAD           ;SETUP TO SET MSEL1 BIT TO A 1
050237  002264                  BIS     R2,R2LOAD               ;ADD STATE OF MSAD BITS 17 AND 16
012737  177740  002270          MOV     #177740,R2MASK          ;SETUP TO IGNORE REG 2 BITS 15-5
004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND COMPARE REG 2
001405                          BEQ     2$                      ;IF OK THEN CONT
                                ERRDF   2,,R2EROR               ;REG 2 FAILED TO LOAD CORRECTLY
104455                          TRAP    C$ERDF
000002                          .WORD   2
000000                          .WORD   0
003710                          .WORD   R2EROR
                                CKLOOP
104406                          TRAP    C$CLP1

                                ;LOAD MSAD BITS 15-8 INTO CONTROL REGISTER 2

010137  002276          2$:     MOV     R1,R4LOAD               ;SETUP ADDRESS BITS TO LOAD
004737  005460                  JSR     PC,LDRDR4               ;GO LOAD, READ AND COMPARE REG 4
001405                          BEQ     3$                      ;IF LOADED OK THEN CONT
                                ERRDF   3,,R4EROR               ;REGISTER 4 FAILED TO LOAD CORRECTLY
104455                          TRAP    C$ERDF
000003                          .WORD   3
000000                          .WORD   0
003776                          .WORD   R4EROR
                                CKLOOP
104406                          TRAP    C$CLP1

                                ;SET BITS MUTB H, MPIN H, WRE H, AND RDE H TO A 0 IN LOCATION
                                ;ADDRESSED BY CONTROL REGISTER 2 AND 4.  THE SIGNAL SMPM L WILL BE
                                ;ASSERTED ON A WRITE AND READ TO CONTROL REGISTER 6.

012737  177760  002306  3$:     MOV     #177760,R6MASK          ;SETUP REGISTER 6 MASK WORD
```

```
3224  010046  005037  002302                    CLR     R6LOAD                  ;SET WORD TO LOAD TO 0
3225  010052  004737  005504                    JSR     PC,LDRDR6               ;GO LOAD, READ AND COMPARE REG 6
3226  010056  001404                            BEQ     4$                      ;IF ALL 0'S THEN CONTINUE
3227  010060                                     ERRDF   4,MSGMP,ALINFO          ;MAP PROTECTION RAM DATA ERROR
3228  010060  104455                            TRAP    C$ERDF
3229  010062  000004                            .WORD   4
3230  010064  002354                            .WORD   MSGMP
3231  010066  004050                            .WORD   ALINFO
3232  010070                              4$:    ENDSEG
3233  010070                              10000$:
3234  010070  104405                            TRAP    C$ESEG
3235  010072  062701  000400                    ADD     #MSAD8,R1               ;UPDATE MSAD BITS 15-8 TO NEXT ADDRESS
3236  010076  001325                            BNE     1$                      ;IF NOT 0 THEN CONTINUE
3237  010100  062702  000001                    ADD     #MSAD16,R2              ;UPDATE ADDRESS BITS 16 AND 17
3238  010104  032702  000004                    BIT     #MSEL0,R2               ;CHECK IF ALL ADDRESSES DONE
3239  010110  001720                            BEQ     1$                      ;IF NOT DO NEXT ADDRESS RANGE
3240  010112                                     ENDSUB
3241  010112                              L10043:
3242  010112  104403                            TRAP    C$ESUB
3243  010114                                     BGNSUB
3244  010114                              T13.2:
3245  010114  104402                            TRAP    C$BSUB
3246  010116  005001                            CLR     R1                      ;RESET THE ADDRESS POINTERS
3247  010120  005002                            CLR     R2                      ;
3248                                             ;THE FOLLOWING PORTION OF THE TEST WILL CHECK FOR ADDRESS SHORTS
3249                                             ;IN THE MAP PROTECTION RAM BY READING THE LOCATION FOR 0'S, WHICH
3250                                             ;WERE PREVIOUSLY WRITTEN IN ABOVE SUBTST, AND THEN WRITING AND
3251                                             ;CHECKING THE LOCATION FOR ONES.
3252
3253  010122                              1$:    BGNSEG
3254  010122  104404                            TRAP    C$BSEG
3255                                             ;SET SIGNAL MSEL1 H TO A ONE AND MSAD BITS 17 OR 16 TO A 1 OR 0 IN
3256                                             ;CONTROL REGISTER 2
3257
3258  010124  012737  000010  002264             MOV     #MSEL1,R2LOAD           ;SETUP TO SET MSEL1 H TO A 1
3259  010132  050237  002264                     BIS     R2,R2LOAD               ;SETUP BITS MSAD 17-16
3260  010136  012737  177740  002270             MOV     #177740,R2MASK          ;SETUP REGISTER 2 MASK WORD
3261  010144  004737  005412                     JSR     PC,LDRDR2               ;GO LOAD, READ AND COMPARE REG 2
3262  010150  001405                            BEQ     2$                      ;IF LOADED OK THEN CONTINUE
3263  010152                                     ERRDF   2,,R2EROR               ;REGISTER 2 FAILED TO LOAD CORRECTLY
3264  010152  104455                            TRAP    C$ERDF
3265  010154  000002                            .WORD   2
3266  010156  000000                            .WORD   0
3267  010160  003710                            .WORD   R2EROR
3268  010162                                     CKLOOP
3269  010162  104406                            TRAP    C$CLP1
3270
3271                                             ;LOAD MSAD BITS 15-8 INTO CONTROL REGISTER 4
3272
3273  010164  010137  002276              2$:    MOV     R1,R4LOAD               ;SETUP ADDRESS BITS 15-8
3274  010170  004737  005460                     JSR     PC,LDRDR4               ;GO LOAD, READ AND COMPARE REGISTER 4
3275  010174  001405                            BEQ     3$                      ;IF OK THEN CONTINUE
3276  010176                                     ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
3277  010176  104455                            TRAP    C$ERDF
3278  010200  000003                            .WORD   3
3279  010202  000000                            .WORD   0
```

```
3280  010204  003776                          .WORD   R4EROR
3281  010206                                  CKLOOP
3282  010206  104406                          TRAP    C$CLP1
3283
3284                                          ;SETUP TO READ MAP PROTECTION RAM FOR ZEROES WHICH WERE WRITTEN IN
3285                                          ;THE PREVIOUS SUBTST.  IF AN ERROR OCCURS, THEN WRITING A PREVIOUS
3286                                          ;LOCATION CHANGED THE LOCATION BEING READ. THIS ERROR IS PROBABLY
3287                                          ;INTERNAL TO RAM CHIP.  THE SIGNAL SMPM L WILL BE ASSERTED DURING
3288                                          ;A READ OF REGISTER 6
3289
3290  010210  012737  177760  002306  3$:     MOV     #177760,R6MASK          ;SETUP REG 6 MASK WORD
3291  010216  005037  002302                  CLR     R6LOAD                  ;SETUP DATA PATTERN THAT WAS LOADED
3292  010222  005037  002304                  CLR     R6GOOD                  ;SETUP EXPECTED DATA PATTERN
3293  010226  004737  005520                  JSR     PC,READR6               ;GO READ REG 6 AND CHECK DATA FOR 0'S
3294  010232  001404                          BEQ     4$                      ;IF ALL ZEROES THEN CONTINUE
3295  010234                                  ERRDF   4,MSGMPS,ALINFO         ;ADDRESS SHORT IN MAP PROTECTION RAM
3296  010234  104455                          TRAP    C$ERDF
3297  010236  000004                          .WORD   4
3298  010240  002415                          .WORD   MSGMPS
3299  010242  004050                          .WORD   ALINFO
3300  010244                          4$:     ENDSEG
3301  010244                          10000$:
3302  010244  104405                          TRAP    C$ESEG
3303
3304  010246                                  BGNSEG
3305  010246  104404                          TRAP    C$BSEG
3306                                          ;THE FOLLWOING SECTION OF CODE WILL WRITE THE LOCATION JUST READ WITH
3307                                          ;ONES AND THEN READ AND CHECK THE LOCATION FOR ONES.  IF THERE IS
3308                                          ;AN INTERNAL ADDRESS SHORT IN THE RAM THIS WILL CAUSE ANOTHER LOCATION
3309                                          ;TO BE WRITTEN AT THE SAME TIME.
3310
3311  010250  012737  000017  002302          MOV     #MUTBH!MPINH!WREH!RDEH,R6LOAD ;SET ALL BITS TO ONES
3312  010256  012737  177760  002306          MOV     #177760,R6MASK          ;SETUP MASK WORD
3313  010264  004737  00>504                  JSR     PC,LDRDR6               ;GO LOAD, READ AND COMPARE REG 6
3314  010270  001404                          BEQ     5$                      ;IF ALL ONES THEN CONT
3315  010272                                  ERRDF   4,MSGMP,ALINFO          ;DATA ERROR IN MAP PROTECTION RAM
3316  010272  104455                          TRAP    C$ERDF
3317  010274  000004                          .WORD   4
3318  010276  002354                          .WORD   MSGMP
3319  010300  004050                          .WORD   ALINFO
3320  010302                          5$:     ENDSEG
3321  010302                          10001$:
3322  010302  104405                          TRAP    C$ESEG
3323  010304  062701  000400                  ADD     #MSAD8,R1               ;UPDATE MSAD ADDRESS 15-8
3324  010310  001304                          BNE     1$                      ;IF NOT 0 THEN DO NEXT ADDRESS
3325  010312  062702  000001                  ADD     #MSAD16,R2              ;UPDATE ADDRESS BITS 17-16
3326  010316  032702  000004                  BIT     #MSEL0,R2               ;CHECK IF DONE
3327  010322  001677                          BEQ     1$                      ;IF DOT DO NEXT HIGHER RANGE
3328  010324                                  ENDSUB
3329  010324                          L10044:
3330  010324  104403                          TRAP    C$ESUB
3331  010326                                  ENDTST
3332  010326                          L10042:
3333  010326  104401                          TRAP    C$ETST
```

TEST 14: CHECK WRV F/F TO SET (0) + CLEAR (1) VIA WRE H,CK H + RST H SEQ 0067

```
                                .SBTTL TEST 14: CHECK WRV F/F TO SET (0) + CLEAR (1) VIA WRE H,CK H + RST H

                                ;++
                                ; TEST TO CHECK THAT THE WRV FLIP-FLOP CAN BE SET VIA THE SIGNALS WRE H AND CK H.
                                ; THE TEST WILL CHECK THAT THE WRV FLIP-FLOP, ONCE SET, CAN NOT BE CLOCKED TO A
                                ; ZERO BY CHANGING THE STATE OF WRE H AND CLOCKING THE SIGNAL CK H AGAIN.  THE
                                ; TEST WILL CHECK THAT THE WRV FLIP-FLOP CAN BE CLEARED WHEN THE SIGNAL RST H
                                ; IS PULSED.  THE TEST WILL ALSO CHECK THAT THE SIGNAL MSBRK H CAN BE SET TO A
                                ; ONE AND ZERO AS A RESULT OF THE WRV FLIP-FLOP BEING SET AND CLEARED.
                                ;--

010330                                  BOT ST
010330                          T14::
010330  004737  004546                  JSR     PC,INITMS               ;SELECT AND INIT THE MEMORY SIMULATOR
010334                                  BGNSEG
010334  104404                          TRAP    C$BSEG
                                        ;SET SIGNALS RST H AND MP H IN CONTROL REGISTER 0.  THE SIGNAL RST H
                                        ;WILL PRESET THE RDV AND WRV FLIP-FLOP'S TO A 1.  THE SIGNAL MP H
                                        ;WILL ALLOW THE MAP PROTECTION RAM BITS MPIN H AND WRE H TO BE READ
                                        ;INTO CONTROL REGISTER 2 ALONG WITH THE SIGNAL BRK L.  THE SIGNALS
                                        ;MPIN H, WRE H, AND BRK L WILL BE READ AS ESR H, WREN H AND MSBRK H
                                        ;IN CONTROL REGISTER 2.

010336  105037  002252                  CLRB    R0LOAD                  ;CLEAR LOWER BYTE OF REG 0 FOR LOOPING
010342  052737  000005  002252          BIS     #RSTH!MPH,R0LOAD        ;SETUP TO SET RST H AND MP H
010350  042737  000060  002256          BIC     #WRVH!RDVH,R0MASK       ;SETUP TO CHECK WRV AND RDV F/F
010356  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REG 0
010362  001405                          BEQ     1$                      ;IF OK THEN CONTINUE
010364                                  ERRDF   1,,R0EROR               ;REGISTER 0 NOT EQUAL EXPECTED
010364  104455                          TRAP    C$ERDF
010366  000001                          .WORD   1
010370  000000                          .WORD   0
010372  003622                          .WORD   R0EROR
010374                                  CKLOOP
010374  104406                          TRAP    C$CLP1

                                        ;CLEAR SIGNAL RST H IN CONTROL REGISTER 0, RDV AND WRV BITS SHOULD NOT
                                        ;CHANGE STATE IN CONTROL REGISTER 0.

010376  042737  000001  002252  1$:     BIC     #RSTH,R0LOAD            ;SETUP TO CLEAR RST H IN REG 0
010404  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REG 0
010410  001405                          BEQ     2$                      ;IF EQUAL THEN CONTINUE
010412                                  ERRDF   1,,R0EROR               ;REGISTER NOT EQUAL EXPECTED
010412  104455                          TRAP    C$ERDF
010414  000001                          .WORD   1
010416  000000                          .WORD   0
010420  003622                          .WORD   R0EROR
010422                                  CKLOOP
010422  104406                          TRAP    C$CLP1

                                        ;SET SIGNAL MSEL1 H TO A ONE AND MSAD BITS 17 AND 16 TO A 0. THE
                                        ;SIGNAL MSBRK H SHOULD BE A 0 DURING THIS TEST

010424  012737  000010  002264  2$:     MOV     #MSEL1,R2LOAD           ;SETUP TO SET MSEL1 H TO A 1
010432  012737  177540  002270          MOV     #177540,R2MASK          ;SETUP TO READ MSBRK H AND LOWER 5 BITS
010440  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REG 2
```

```
3390  010444  001405                        BEQ     3$                      ;IF OK THEN CONTINUE
3391  010446                                ERRDF   2,,R2EROR               ;REG 2 NOT EQUAL EXPECTED
3392  010446  104455                        TRAP    C$ERDF
3393  010450  000002                        .WORD   2
3394  010452  000000                        .WORD   0
3395  010454  003710                        .WORD   R2EROR
3396  010456                                CKLOOP
3397  010456  104406                        TRAP    C$CLP1
3398
3399                                        ;LOAD MSAD BIT 15 THROUGH 8 TO A 0 TO SELECT ADDRESS 0
3400
3401  010460  005037  002276        3$:     CLR     R4LOAD                  ;SETUP TO SET ALL BITS TO 0
3402  010464  004737  005460                JSR     PC,LDRDR4               ;GO LOAD,READ AND CHECK REG 4
3403  010470  001405                        BEQ     4$                      ;IF NO ERRORS THEN CONTINUE
3404  010472                                ERRDF   3,,R4EROR               ;REGISTER 4 NOT ALL ZEROES
3405  010472  104455                        TRAP    C$ERDF
3406  010474  000003                        .WORD   3
3407  010476  000000                        .WORD   0
3408  010500  003776                        .WORD   R4EROR
3409  010502                                CKLOOP
3410  010502  104406                        TRAP    C$CLP1
3411
3412                                        ;SET BITS MPIN H, RDE H AND MUTB H TO A ONE AND BIT WRE H TO A 0.
3413                                        ;WRE H ON A 0 WILL SET THE WRV FLIP-FLOP WHEN BIT CK H IS TOGGLED
3414                                        ;IN CONTROL REGISTER 0.
3415
3416  010504  012737  000015  002302  4$:     MOV     #MPINH!RDEH!MUTBH,R6LOAD ;SETUP TO LOAD THE MAP PROTECT RAM
3417  010512  012737  177760  002306          MOV     #177760,R6MASK          ;SETUP REGISTER 6 MASK WORD
3418  010520  004737  005504                JSR     PC,LDRDR6               ;GO LOAD,READ AND CHECK REGISTER 6
3419  010524  001405                        BEQ     5$                      ;OF COMPARED OK THEN CONTINUE
3420  010526                                ERRDF   4,MSGMP,ALINFO          ;MAP PROTECT RAM DATA ERROR
3421  010526  104455                        TRAP    C$ERDF
3422  010530  000004                        .WORD   4
3423  010532  002354                        .WORD   MSGMP
3424  010534  004050                        .WORD   ALINFO
3425  010536                                CKLOOP
3426  010536  104406                        TRAP    C$CLP1
3427
3428                                        ;CHECK CONTROL REGISTER 0 TO MAKE SURE THAT NO CHANGES OCCURED.
3429
3430  010540  004737  005360          5$:     JSR     PC,READR0               ;READ AND CHECK REGISTER 0
3431  010544  001405                        BEQ     6$                      ;IF OK THEN CONTINUE
3432  010546                                ERRDF   1,MSGMPL,ALROIN         ;REGISTER 0 CHANGED STATES
3433  010546  104455                        TRAP    C$ERDF
3434  010550  000001                        .WORD   1
3435  010552  002461                        .WORD   MSGMPL
3436  010554  004076                        .WORD   ALROIN
3437  010556                                CKLOOP
3438  010556  104406                        TRAP    C$CLP1
3439
3440                                        ;SETUP TO READ CONTROL REGISTER 2 WITH MP H SET TO A 1 IN CONTROL
3441                                        ;REGISTER 0 TO ALLOW MPIN H, WRE H AND BRK L TO BE READ INTO
3442                                        ;CONTROL REGISTER 2 AS BITS ESR H, WREN H, AND MSBRK H.
3443
3444  010560  042737  000140  002270  6$:     BIC     #ESRH!WRENH,R2MASK      ;CLEAR BITS TO BE CHECKED
3445  010566  052737  000040  002266          BIS     #ESRH,R2GOOD            ;SETUP EXPECTED MAP PROTECT BIT
```

```
3446  010574  004737  005426                    JSR     PC,READR2               ;GO READ REGISTER 2 AND CHFCK IT
3447  010600  001405                            BEQ     7$                      ;IF OK THEN CONTINUE
3448  010602                                    ERRDF   2,MSGMPL,ALR2IN         ;REGISTER 2 NOT EQUAL EXPECTED
3449  010602  104455                            TRAP    C$ERDF
3450  010604  000002                            .WORD   2
3451  010606  002461                            .WORD   MSGMPL
3452  010610  004124                            .WORD   ALR2IN
3453  010612                                    CKLOOP
3454  010612  104406                            TRAP    C$CLP1
3455
3456                                            ;SET SIGNAL CK H IN CONTROL REGISTER 0 TO CLOCK THE WRV AND RDV
3457                                            ;F/F'S.  THE WRV F/F SHOULD BE SET (0) AND THE RDV F/F SHOULD BE CLEARED (1).
3458
3459  010614  052737  000140  002254  7$:       BIS     #CKH!WRVH,ROGOOD        ;SETUP EXPECTED DATA
3460  010622  052737  000100  002252            BIS     #CKH,ROLOAD             ;SETUP BIT TO BE LOADED
3461  010630  004737  005352                    JSR     PC,LDRDOR               ;GO LOAD, READ AND CHECK REG 0
3462  010634  001405                            BEQ     8$                      ;IF OK THEN CONTINUE
3463  010636                                    ERRDF   1,MSGMPL,ALR0IN         ;WRV F/F PROBABLY NOT SET (0)
3464  010636  104455                            TRAP    C$ERDF
3465  010640  000001                            .WORD   1
3466  010642  002461                            .WORD   MSGMPL
3467  010644  004076                            .WORD   ALR0IN
3468  010646                                    CKLOOP
3469  010646  104406                            TRAP    C$CLP1
3470
3471                                            ;CLEAR THE SIGNAL CK H IN CONTROL REGISTER 0. NO OTHER BITS SHOULD CHANGE.
3472
3473  010650  042737  000100  002254  8$:       BIC     #CKH,ROGOOD             ;CLEAR CKH IN EXPECTED BITS
3474  010656  042737  000100  002252            BIC     #CKH,ROLOAD             ;SETUP TO CLEAR BIT CK H IN REG 0
3475  010664  004737  005352                    JSR     PC,LDRDOR               ;GO CLEAR, LOAD AND CHECK REG 0
3476  010670  001405                            BEQ     9$                      ;IF OK THEN CONTINUE
3477  010672                                    ERRDF   1,MSGMPL,ALR0IN         ;REG 0 NOT EQUAL EXPECTED
3478  010672  104455                            TRAP    C$ERDF
3479  010674  000001                            .WORD   1
3480  010676  002461                            .WORD   MSGMPL
3481  010700  004076                            .WORD   ALR0IN
3482  010702                                    CKLOOP
3483  010702  104406                            TRAP    C$CLP1
3484
3485                                            ;CHECK THAT SIGNAL MSBRK H IS SET TO A 1 IN CONTROL REGISTER 2
3486
3487  010704  052737  000240  002266  9$:       BIS     #MSBRKH!ESRH,R2GOOD     ;SETUP EXPECTED BITS IN REGISTER 2
3488  010712  004737  005426                    JSR     PC,READR2               ;GO READ REGISTER 2 AND CHECK IT
3489  010716  001405                            BEQ     10$                     ;IF OK THEN CONTINUE
3490  010720                                    ERRDF   2,MSGMPL,ALR2IN         ;MSBRK H PROBABLY NOT SET VIA BRK L BEING LOW
3491  010720  104455                            TRAP    C$ERDF
3492  010722  000002                            .WORD   2
3493  010724  002461                            .WORD   MSGMPL
3494  010726  004124                            .WORD   ALR2IN
3495  010730                                    CKLOOP
3496  010730  104406                            TRAP    C$CLP1
3497
3498                                            ;THE FOLLOWING SECTION WILL CHECK THAT THE WRV FLIP-FLOP CAN NOT
3499                                            ;BE CLEARED (1) BY TRYING TO CLOCK A ONE INTO IT.  ONCE THE WRV FLIP-
3500                                            ;FLOP IS SET (0), IT IS LATCHED TO THAT STATE UNTIL A PULSE IS ISSUED
3501                                            ;ON THE SIGNALS RST H OR INIT H.
```

```
3502
3503  010732  012737  000017  002302  10$:    MOV     #MPINH!WREH.RDEH!MUTBH,R6LOAD ;SETUP TO SET ALL RAM BITS TO 1
3504  010740  004737  005504                  JSR     PC,LDRDR6                     ;GO LOAD, READ AND CHECK RAM
3505  010744  001405                          BEQ     11$                           ;IF ALL ONES THEN CONTINUE
3506  010746                                  ERRDF   4,MSGMP,ALINFO                ;MAP PROTECT RAM DATA ERROR
3507  010746  104455                          TRAP    C$ERDF
3508  010750  000004                          .WORD   4
3509  010752  002354                          .WORD   MSGMP
3510  010754  004050                          .WORD   ALINFO
3511  010756                                  CKLOOP
3512  010756  104406                          TRAP    C$CLP1
3513
3514                                          ;CHECK THAT CONTROL REGISTER 0 DID NOT CHANGE STATE.
3515
3516  010760  004737  005360          11$:    JSR     PC,READR0                     ;GO READ AND CHECK REGISTER 0
3517  010764  001405                          BEQ     12$                           ;IF NO CHANGES THEN CONTINUE
3518  010766                                  ERRDF   1,MSGMPL,ALR0IN               ;REGISTER 0 CHANGED STATE
3519  010766  104455                          TRAP    C$ERDF
3520  010770  000001                          .WORD   1
3521  010772  002461                          .WORD   MSGMPL
3522  010774  004076                          .WORD   ALR0IN
3523  010776                                  CKLOOP
3524  010776  104406                          TRAP    C$CLP1
3525
3526                                          ;CHECK CONTROL REGISTER 2 TO HAVE ESR H, WREN H AND MSBRK H SET TO 1
3527
3528  011000  052737  000340  002266  12$:    BIS     #ESRH!WRENH!MSBRKH,R2GOOD ;ADD WREN H TO EXPECTED DATA
3529  011006  004737  005426                  JSR     PC,READR2                     ;GO READ AND CHECK REGISTER 2
3530  011012  001405                          BEQ     13$                           ;IF NO ERRORS THEN CONTINUE
3531  011014                                  ERRDF   2,MSGMPL,A_R2IN               ;REGISTER 2 NOT EQUAL EXPECTED
3532  011014  104455                          TRAP    C$ERDF
3533  011016  000002                          .WORD   2
3534  011020  002461                          .WORD   MSGMPL
3535  011022  004124                          .WORD   ALR2IN
3536  011024                                  CKLOOP
3537  011024  104406                          TRAP    C$CLP1
3538
3539                                          ;SET THE SIGNAL CK H TO THE HIGH STATE TO CLOCK THE WRV AND RDV FLIP-
3540                                          ;FLOPS.  THE WRV FLIP-FLOP SHOULD REMAIN SET (0) AS A RESULT OF IT BEING
3541                                          ;SET ALREADY.  THE WRV FLIP-FLOP, ONCE SET, CAN NOT BE CLOCKED AGAIN
3542                                          ;UNLESS IT HAS BEEN PRESET BY A PULSE ON THE SIGNALS RST H OR INIT H.
3543
3544  011026  052737  000100  002252  13$:    BIS     #CKH,R0LOAD                   ;SETUP TO SET SIGNAL CK H TO HIGH STATE
3545  011034  052737  000100  002254          BIS     #CKH,R0GOOD                   ;EXPECT CK H TO BE SET IN CONTROL REG 0
3546  011042  004737  005352                  JSR     PC,LDRDR0                     ;GO LOAD, READ AND CHECK CONTROL REG 0
3547  011046  001405                          BEQ     14$                           ;IF OK THEN CONTINUE
3548  011050                                  ERRDF   1,MSGMPL,ALR0IN               ;WRV F/F CLEARED (1) AFTER BEING LATCHED
3549  011050  104455                          TRAP    C$ERDF
3550  011052  000001                          .WORD   1
3551  011054  002461                          .WORD   MSGMPL
3552  011056  004076                          .WORD   ALR0IN
3553  011060                                  CKLOOP
3554  011060  104406                          TRAP    C$CLP1
3555
3556                                          ;READ CONTROL REG 2 TO CHECK THAT ESR H, WREN H AND MSBRK H ARE STILL SET (1).
3557
```

```
3558  011062  004737  005426          14$:    JSR     PC,READR2               ;READ AND CHECK CONTROL REGISTER 2
3559  011066  001405                          BEQ     15$                     ;IF OK THEN CONTINUE
3560  011070                                  ERRDF   2,MSGMPL,ALR2IN         ;REGISTER 2 NOT EQUAL EXPECTED
3561  011070  104455                          TRAP    C$ERDF
3562  011072  000002                          .WORD   2
3563  011074  002461                          .WORD   MSGMPL
3564  011076  004124                          .WORD   ALR2IN
3565  011100                                  CKLOOP
3566  011100  104406                          TRAP    C$CLP1
3567
3568                                          ;CLEAR THE SIGNAL CK H AND SET THE SIGNAL RST H IN CONTROL REGISTER 0.
3569                                          ;SETTING THE SIGNAL RST H TO A ONE WILL CAUSE THE WRV FLIP-FLOP TO BE
3570                                          ;SET IN ITS PRESET STATE (1).
3571
3572  011102  112737  000005  002252  15$:    MOVB    #RSTH!MPH,ROLOAD        ;SET RST H AND CLEAR CK H
3573  011110  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK CONTROL REG 0
3574  011114  001405                          BEQ     16$                     ;IF LOADED OK THEN CONTINUE
3575  011116                                  ERRDF   1,MSGMPL,ALR0IN         ;WRV F/F PROBABLY NOT CLEARED BY RST H
3576  011116  104455                          TRAP    C$ERDF
3577  011120  000001                          .WORD   1
3578  011122  002461                          .WORD   MSGMPL
3579  011124  004076                          .WORD   ALR0IN
3580  011126                                  CKLOOP
3581  011126  104406                          TRAP    C$CLP1
3582
3583                                          ;CLEAR THE SIGNAL RST H IN CONTROL REGISTER 0
3584
3585  011130  042737  000001  002252  16$:    BIC     #RSTH,ROLOAD            ;SETUP TO CLEAR RST H
3586  011136  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK CONTROL REG 0
3587  011142  001405                          BEQ     17$                     ;IF LOADED OK THEN CONTINUE
3588  011144                                  ERRDF   1,MSGMPL,ALR0IN         ;CONTROL REG 0 NOT EQUAL EXPECTED
3589  011144  104455                          TRAP    C$ERDF
3590  011146  000001                          .WORD   1
3591  011150  002461                          .WORD   MSGMPL
3592  011152  004076                          .WORD   ALR0IN
3593  011154                                  CKLOOP
3594  011154  104406                          TRAP    C$CLP1
3595
3596                                          ;CHECK CONTROL REGISTER 2 TO MAKE SURE THAT THE SIGNAL MSBRK H WENT TO
3597                                          ;A ZERO AS A RESULT OF THE WRV AND RDV FLIP-FLOPS BEING PRESET BY RST H.
3598
3599  011156  042737  000200  002266  17$:    BIC     #MSBRKH,R2GOOD          ;CLEAR MSBRK H IN EXPECTED DATA
3600  011164  004737  005426                  JSR     PC,READR2               ;READ AND CHECK CONTROL REGISTER 2
3601  011170  001404                          BEQ     18$                     ;IF MSBRK H EQUALS A 0 THEN CONTINUE
3602  011172                                  ERRDF   2,MSGMPL,ALR2IN         ;CONTROL REGISTER 2 NOT EQUAL EXPECTED
3603  011172  104455                          TRAP    C$ERDF
3604  011174  000002                          .WORD   2
3605  011176  002461                          .WORD   MSGMPL
3606  011200  004124                          .WORD   ALR2IN
3607  011202                          18$:    ENDSEG
3608  011202                          10000$:
3609  011202  104405                          TRAP    C$ESEG
3610  011204                                  ENDTST
3611  011204                          L10045:
3612  011204  104401                          TRAP    C$ETST
3613
```

```
                                        .SBTTL  TEST 15: CHECK RDV F/F TO SET (0) + CLEAR (1) VIA RDE H, CK H + RST H

                                        ;++
                                        ; TEST TO CHECK THAT THE RDV FLIP-FLOP CAN BE SET VIA RDE H AND CK H.  THE TEST
                                        ; WILL CHECK THAT THE RDV FLIP-FLOP, ONCE SET, CAN NOT BE CLOCKED TO A ZERO BY
                                        ; CHANGING THE STATE OF RDE H AND CLOCKING THE SIGNAL CK H AGAIN.  THE TEST WILL
                                        ; CHECK THAT THE RDV FLIP-FLOP CAN BE CLEARED WHEN THE SIGNAL RST H IS PULSED.
                                        ; THE TEST WILL ALSO CHECK THAT THE SIGNAL MSBRK H CAN BE SET TO A ONE AND ZERO
                                        ; AS A RESULT OF THE RDV FLIP-FLOP BEING SET AND CLEARED.
                                        ;--

011206                                          BGNTST
011206                                  T15:
011206  004737  004546                          JSR     PC,INITMS               ;SELECT AND INIT THE MEMORY SIMULATOR
011212                                          BGNSEG
011212  104404                                  TRAP    C$BSEG
                                                ;SET SIGNALS RST H AND MP H IN CONTROL REGISTER 0.  THE SIGNAL RST H
                                                ;WILL PRESET THE RDV AND WRV FLIP-FLOP'S TO A 1.  THE SIGNAL MP H
                                                ;WILL ALLOW THE MAP PROTECTION RAM BITS MPIN H AND WRE H TO BE READ
                                                ;INTO CONTROL REGISTER 2 ALONG WITH THE SIGNAL BRK L.  THE SIGNALS
                                                ;MPIN H, WRE H, AND BRK L WILL BE READ AS ESR H, WREN H AND MSBRK H
                                                ;IN CONTROL REGISTER 2.

011214  105037  002252                          CLRB    ROLOAD                  ;CLEAR LOWER BYTE OF REG 0 FOR LOOPING
011220  052737  000005  002252                  BIS     #RSTH!MPH,ROLOAD        ;SETUP TO SET RST H AND MP H
011226  042737  000060  002256                  BIC     #WRVH!RDVH,ROMASK       ;SETUP TO CHECK WRV AND RDV F/F
011234  004737  005344                          JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REG 0
011240  001405                                  BEQ     1$                      ;IF OK THEN CONTINUE
011242                                          ERRDF   1,,ROERROR              ;REGISTER 0 NOT EQUAL EXPECTED
011242  104455                                  TRAP    C$ERDF
011244  000001                                  .WORD   1
011246  000000                                  .WORD   0
011250  003622                                  .WORD   ROERROR
011252                                          CKLOOP
011252  104406                                  TRAP    C$CLP1

                                                ;CLEAR SIGNAL RST H IN CONTROL REGISTER 0, RDV AND WRV BITS SHOULD NOT
                                                ;CHANGE STATE IN CONTROL REGISTER 0.

011254  042737  000001  002252  1$:             BIC     #RSTH,ROLOAD            ;SETUP TO CLEAR RST H IN REG 0
011262  004737  005344                          JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REG 0
011266  001405                                  BEQ     2$                      ;IF EQUAL THEN CONTINUE
011270                                          ERRDF   1,,ROERROR              ;REGISTER NOT EQUAL EXPECTED
011270  104455                                  TRAP    C$ERDF
011272  000001                                  .WORD   1
011274  000000                                  .WORD   0
011276  003622                                  .WORD   ROERROR
011300                                          CKLOOP
011300  104406                                  TRAP    C$CLP1

                                                ;SET SIGNAL MSEL1 H TO A ONE AND MSAD BITS 17 AND 16 TO A 0. THE
                                                ;SIGNAL MSBRK H SHOULD BE A 0 DURING THIS TEST

011302  012737  000010  002264  2$:             MOV     #MSEL1,R2LOAD           ;SETUP TO SET MSEL1 H TO A 1
011310  012737  177540  002270                  MOV     #177540,R2MASK          ;SETUP TO READ MSBRK H AND LOWER 5 BITS
011316  004737  005412                          JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REG 2
```

```
3670  011322  001405                          BEQ     3$                      ;IF OK THEN CONTINUE
3671  011324                                  ERRDF   2,,R2EROR               ;REG 2 NOT EQUAL EXPECTED
3672  011324  104455                          TRAP    C$ERDF
3673  011326  000002                          .WORD   2
3674  011330  000000                          .WORD   0
3675  011332  003710                          .WORD   R2EROR
3676  011334                                  CKLOOP
3677  011334  104406                          TRAP    C$CLP1
3678
3679                                          ;LOAD MSAD BIT 15 THROUGH 8 TO A 0 TO SELECT ADDRESS 0
3680
3681  011336  005037  002276          3$:     CLR     R4LOAD                  ;SETUP TO SET ALL BITS TO 0
3682  011342  004737  005460                  JSR     PC,LDRDR4               ;GO LOAD,READ AND CHECK REG 4
3683  011346  001405                          BEQ     4$                      ;IF NO ERRORS THEN CONTINUE
3684  011350                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT ALL ZEROES
3685  011350  104455                          TRAP    C$ERDF
3686  011352  000003                          .WORD   3
3687  011354  000000                          .WORD   0
3688  011356  003776                          .WORD   R4EROR
3689  011360                                  CKLOOP
3690  011360  104406                          TRAP    C$CLP1
3691
3692                                          ;SET BITS MPIN H, WRE H AND MUTB H TO A ONE AND BIT RDE H TO A 0.
3693                                          ;RDE H ON A 0 WILL SET THE RDV FLIP-FLOP WHEN BIT CK H IS TOGGLED
3694                                          ;IN CONTROL REGISTER 0.
3695
3696  011362  012737  000013  002302  4$:     MOV     #MPINH!WREH!MUTBH,R6LOAD ;SETUP TO LOAD THE MAP PROTECT RAM
3697  011370  012737  177760  002306          MOV     #177760,R6MASK          ;SETUP REGISTER 6 MASK WORD
3698  011376  004737  005504                  JSR     PC,LDRDR6               ;GO LOAD,READ AND CHECK REGISTER 6
3699  011402  001405                          BEQ     5$                      ;OF COMPARED OK THEN CONTINUE
3700  011404                                  ERRDF   4,MSGMP,ALINFO          ;MAP PROTECT RAM DATA ERROR
3701  011404  104455                          TRAP    C$ERDF
3702  011406  000004                          .WORD   4
3703  011410  002354                          .WORD   MSGMP
3704  011412  004050                          .WORD   ALINFO
3705  011414                                  CKLOOP
3706  011414  104406                          TRAP    C$CLP1
3707
3708                                          ;CHECK CONTROL REGISTER 0 TO MAKE SURE THAT NO CHANGES OCCURED.
3709
3710  011416  004737  005360          5$:     JSR     PC,READR0               ;READ AND CHECK REGISTER 0
3711  011422  001405                          BEQ     6$                      ;IF OK THEN CONTINUE
3712  011424                                  ERRDF   1,MSGMPL,ALR0IN         ;REGISTER 0 CHANGED STATES
3713  011424  104455                          TRAP    C$ERDF
3714  011426  000001                          .WORD   1
3715  011430  002461                          .WORD   MSGMPL
3716  011432  004076                          .WORD   ALR0IN
3717  011434                                  CKLOOP
3718  011434  104406                          TRAP    C$CLP1
3719
3720                                          ;SETUP TO READ CONTROL REGISTER 2 WITH MP H SET TO A 1 IN CONTROL
3721                                          ;REGISTER 0 TO ALLOW MPIN H, WRE H AND BRK L TO BE READ INTO
3722                                          ;CONTROL REGISTER 2 AS BITS ESR H, WREN H, AND MSBRK H.
3723
3724  011436  042737  000140  002270  6$:     BIC     #ESRH!WRENH,R2MASK      ;CLEAR BITS TO BE CHECKED
3725  011444  052737  000140  002266          BIS     #ESRH.WRENH,R2GOOD      ;SETUP EXPECTED MAP PROTECT BIT
```

```
3726  011452  004737  005426                  JSR     PC,READR2               ;GO READ REGISTER 2 AND CHECK IT
3727  011456  001405                          BEQ     7$                      ;IF OK THEN CONTINUE
3728  011460                                  ERRDF   2,MSGMPL,ALR2IN         ;REGISTER 2 NOT EQUAL EXPECTED
3729  011460  104455                          TRAP    C$ERDF
3730  011462  000002                          .WORD   2
3731  011464  002461                          .WORD   MSGMPL
3732  011466  004124                          .WORD   ALR2IN
3733  011470                                  CKLOOP
3734  011470  104406                          TRAP    C$CLP1
3735
3736                                          ;SET SIGNAL CK H IN CONTROL REGISTER 0 TO CLOCK THE WRV AND RDV
3737                                          ;F/F'S.  THE RDV F/F SHOULD BE SET (0) AND THE WRV F/F SHOULD BE CLEARED (1).
3738
3739  011472  052737  000120  002254  7$:     BIS     #CKH!RDVH,ROGOOD        ;SETUP EXPECTED DATA
3740  011500  052737  000100  002252          BIS     #CKH,ROLOAD             ;SETUP BIT TO BE LOADED
3741  011506  004737  005352                  JSR     PC,LDRDOR               ;GO LOAD, READ AND CHECK REG 0
3742  011512  001405                          BEQ     8$                      ;IF OK THEN CONTINUE
3743  011514                                  ERRDF   1,MSGMPL,ALR0IN         ;RDV F/F PROBABLY NOT SET (0)
3744  011514  104455                          TRAP    C$ERDF
3745  011516  000001                          .WORD   1
3746  011520  002461                          .WORD   MSGMPL
3747  011522  004076                          .WORD   ALR0IN
3748  011524                                  CKLOOP
3749  011524  104406                          TRAP    C$CLP1
3750
3751                                          ;CLEAR THE SIGNAL CK H IN CONTROL REG 0. NO OTHER BITS SHOULD CHANGE STATE.
3752
3753  011526  042737  000100  002254  8$:     BIC     #CKH,ROGOOD             ;CLEAR CKH IN EXPECTED BITS
3754  011534  042737  000100  002252          BIC     #CKH,ROLOAD             ;SETUP TO CLEAR BIT CK H IN REG 0
3755  011542  004737  005352                  JSR     PC,LDRDOR               ;GO CLEAR, LOAD AND CHECK REG 0
3756  011546  001405                          BEQ     9$                      ;IF OK THEN CONTINUE
3757  011550                                  ERRDF   1,MSGMPL,ALR0IN         ;REG 0 NOT EQUAL EXPECTED
3758  011550  104455                          TRAP    C$ERDF
3759  011552  000001                          .WORD   1
3760  011554  002461                          .WORD   MSGMPL
3761  011556  004076                          .WORD   ALR0IN
3762  011560                                  CKLOOP
3763  011560  104406                          TRAP    C$CLP1
3764
3765                                          ;CHECK THAT SIGNAL MSBRK H IS SET TO A 1 IN CONTROL REGISTER 2
3766
3767  011562  052737  000340  002266  9$:     BIS     #MSBRKH.WRENH!ESRH,R2GOOD ;SETUP EXPECTED BITS IN REGISTER 2
3768  011570  004737  005426                  JSR     PC,READR2               ;GO READ REGISTER 2 AND CHECK IT
3769  011574  001405                          BEQ     10$                     ;IF OK THEN CONTINUE
3770  011576                                  ERRDF   2,MSGMPL,ALR2IN         ;MSBRK H PROBABLY NOT SET VIA BRK L BEING LOW
3771  011576  104455                          TRAP    C$ERDF
3772  011600  000002                          .WORD   2
3773  011602  002461                          .WORD   MSGMPL
3774  011604  004124                          .WORD   ALR2IN
3775  011606                                  CKLOOP
3776  011606  104406                          TRAP    C$CLP1
3777
3778                                          ;THE FOLLOWING SECTION WILL CHECK THAT THE RDV FLIP-FLOP CAN NOT BE
3779                                          ;CLEARED (1) BY TRYING TO CLOCK A ONE INTO IT.  ONCE THE RDV FLIP-FLOP
3780                                          ;IS SET (0), IT IS LATCHED TO THAT STATE UNTIL A PULSE IS ISSUED ON THE
3781                                          ;SIGNAL RST H OR INIT H.
```

```
3782
3783  011610  012737  000017  002302  10$:    MOV     #MPINH!WREH!RDEH!MUTBH,R6LOAD ;SETUP TO SET ALL RAM BITS TO 1
3784  011616  004737  005504                  JSR     PC,LDRDR6                     ;GO LOAD, READ AND CHECK RAM
3785  011622  001405                          BEQ     11$                           ;IF ALL ONES THEN CONTINUE
3786  011624                                  ERRDF   4,MSGMP,ALINFO                ;MAP PROTECT RAM DATA ERROR
3787  011624  104455                          TRAP    C$ERDF
3788  011626  000004                          .WORD   4
3789  011630  002354                          .WORD   MSGMP
3790  011632  004050                          .WORD   ALINFO
3791  011634                                  CKLOOP
3792  011634  104406                          TRAP    C$CLP1
3793
3794                                          ;CHECK THAT CONTROL REGISTER 0 DID NOT CHANGE STATE.
3795
3796  011636  004737  005360          11$:    JSR     PC,READR0                     ;GO READ AND CHECK REGISTER 0
3797  011642  001405                          BEQ     12$                           ;IF NO CHANGES THEN CONTINUE
3798  011644                                  ERRDF   1,MSGMPL,ALR0IN               ;REGISTER 0 CHANGED STATE
3799  011644  104455                          TRAP    C$ERDF
3800  011646  000001                          .WORD   1
3801  011650  002461                          .WORD   MSGMPL
3802  011652  004076                          .WORD   ALR0IN
3803  011654                                  CKLOOP
3804  011654  104406                          TRAP    C$CLP1
3805
3806                                          ;CHECK CONTROL REGISTER 2 TO HAVE ESR H, WREN H AND MSBRK H SET TO 1
3807
3808  011656  052737  000340  002266  12$:    BIS     #ESRH!WRENH!MSBRKH,R2GOOD ;SETUP EXPECTED DATA
3809  011664  004737  005426                  JSR     PC,READR2                     ;GO READ AND CHECK REGISTER 2
3810  011670  001405                          BEQ     13$                           ;IF NO ERRORS THEN CONTINUE
3811  011672                                  ERRDF   2,MSGMPL,ALR2IN               ;REGISTER 2 NOT EQUAL EXPECTED
3812  011672  104455                          TRAP    C$ERDF
3813  011674  000002                          .WORD   2
3814  011676  002461                          .WORD   MSGMPL
3815  011700  004124                          .WORD   ALR2IN
3816  011702                                  CKLOOP
3817  011702  104406                          TRAP    C$CLP1
3818
3819                                          ;SET THE SIGNAL CK H TO THE HIGH STATE TO CLOCK THE WRV AND RDV FLIP-
3820                                          ;FLOPS.  THE RDV FLIP-FLOP SHOULD REMAIN SET (0) AS A RESULT OF IT BEING
3821                                          ;SET ALREADY.  THE RDV FLIP-FLOP, ONCE SET, CAN NOT BE CLOCKED AGAIN
3822                                          ;UNLESS IT HAS BEEN PRESET BY A PULSE ON THE SIGNALS RST H OR INIT H.
3823
3824  011704  052737  000100  002252  13$:    BIS     #CKH,R0LOAD                   ;SETUP TO SET SIGNAL CK H TO HIGH STATE
3825  011712  052737  000100  002254          BIS     #CKH,R0GOOD                   ;EXPECT CK H TO BE SET IN CONTROL REG 0
3826  011720  004737  005352                  JSR     PC,LDRDR0                     ;GO LOAD, READ AND CHECK CONTROL REG 0
3827  011724  001405                          BEQ     14$                           ;IF OK THEN CONTINUE
3828  011726                                  ERRDF   1,MSGMPL,ALR0IN               ;RDV F/F CLEARED (1) AFTER BEING LATCHED
3829  011726  104455                          TRAP    C$ERDF
3830  011730  000001                          .WORD   1
3831  011732  002461                          .WORD   MSGMPL
3832  011734  004076                          .WORD   ALR0IN
3833  011736                                  CKLOOP
3834  011736  104406                          TRAP    C$CLP1
3835
3836                                          ;READ CONTROL REG 2 TO CHECK THAT ESR H, WREN H AND MSBRK H ARE STILL SET TO 1'S
3837
```

```
3838  011740  004737  005426          14$:    JSR     PC,READR2               ;READ AND CHECK CONTROL REGISTER 2
3839  011744  001405                          BEQ     15$                     ;IF OK THEN CONTINUE
3840  011746                                  ERRDF   2,MSGMPL,ALR2IN         ;REGISTER 2 NOT EQUAL EXPECTED
3841  011746  104455                          TRAP    C$ERDF
3842  011750  000002                          .WORD   2
3843  011752  002461                          .WORD   MSGMPL
3844  011754  004124                          .WORD   ALR2IN
3845  011756                                  CKLOOP
3846  011756  104406                          TRAP    C$CLP1
3847
3848                                          ;CLEAR THE SIGNAL CK H AND SET THE SIGNAL RST H IN CONTROL REGISTER 0.
3849                                          ;SETTING THE SIGNAL RST H TO A ONE WILL CAUSE THE RDV FLIP-FLOP TO BE
3850                                          ;SET IN ITS PRESET STATE (1).
3851
3852  011760  112737  000005  002252  15$:    MOVB    #RSTH!MPH,ROLOAD        ;SET RST H AND CLEAR CK H
3853  011766  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK CONTROL REG 0
3854  011772  001405                          BEQ     16$                     ;IF LOADED OK THEN CONTINUE
3855  011774                                  ERRDF   1,MSGMPL,ALR0IN         ;RDV F/F PROBABLY NOT CLEARED BY RST H
3856  011774  104455                          TRAP    C$ERDF
3857  011776  000001                          .WORD   1
3858  012000  002461                          .WORD   MSGMPL
3859  012002  004076                          .WORD   ALR0IN
3860  012004                                  CKLOOP
3861  012004  104406                          TRAP    C$CLP1
3862
3863                                          ;CLEAR THE SIGNAL RST H IN CONTROL REGISTER 0
3864
3865  012006  042737  000001  002252  16$:    BIC     #RSTH,ROLOAD            ;SETUP TO CLEAR RST H
3866  012014  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK CONTROL REG 0
3867  012020  001405                          BEQ     17$                     ;IF LOADED OK THEN CONTINUE
3868  012022                                  ERRDF   1,MSGMPL,ALR0IN         ;CONTROL REG 0 NOT EQUAL EXPECTED
3869  012022  104455                          TRAP    C$ERDF
3870  012024  000001                          .WORD   1
3871  012026  002461                          .WORD   MSGMPL
3872  012030  004076                          .WORD   ALR0IN
3873  012032                                  CKLOOP
3874  012032  104406                          TRAP    C$CLP1
3875
3876                                          ;CHECK CONTROL REGISTER 2 TO MAKE SURE THAT THE SIGNAL MSBRK H WENT TO
3877                                          ;A ZERO AS A RESULT OF THE WRV AND RDV FLIP-FLOPS BEING PRESET BY RST H.
3878
3879  012034  042737  000200  002266  17$:    BIC     #MSBRKH,R2GOOD          ;CLEAR MSBRK H IN EXPECTED DATA
3880  012042  004737  005426                  JSR     PC,READR2               ;READ AND CHECK CONTROL REGISTER 2
3881  012046  001404                          BEQ     18$                     ;IF MSERK H EQUALS A 0 THEN CONTINUE
3882  012050                                  ERRDF   2,MSGMPL,ALR2IN         ;CONTROL REGISTER 2 NOT EQUAL EXPECTED
3883  012050  104455                          TRAP    C$ERDF
3884  012052  000002                          .WORD   2
3885  012054  002461                          .WORD   MSGMPL
3886  012056  004124                          .WORD   ALR2IN
3887  012060                          18$:    ENDSEG
3888  012060                          10000$:
 889  012060  104405                          TRAP    C$ESEG
3890  012062                                  ENDTST
3891  012062                          L10046:
3892  012062  104401                          TRAP    C$ETST
3893
```

```
                                .SBTTL TEST 16: CHECK THAT SIGNAL RST H WILL CLEAR RDV + WRV F/F'S

                                ;++
                                ; TEST TO CHECK THAT THE RDV AND WRV FLIP-FLOPS CAN BE SET VIA THE SIGNALS
                                ; WRE H, RDE H, AND CK H.  THE TEST WILL CHECK THAT THE SIGNAL RST H WILL
                                ; CLEAR THE RDV AND WRV FLIP-FLOPS.  THE SIGNAL MSBRK WILL BE CHECKED TO BE
                                ; A ONE WHEN THE RDV AND WRV FLIP-FLOPS ARE SET AND CHECKED FOR A 0 WHEN THE
                                ; FLIP-FLOPS ARE CLEARED.
                                ;--

                                        BGNTST
                                T16::
012064  004737  004546                  JSR     PC,INITMS               ;SELECT AND INIT THE MEMORY SIMULATOR
                                        BGNSEG
012070  104404                          TRAP    C$BSEG

                                        ;SET SIGNALS RST H AND MP H IN CONTROL REGISTER 0.  THE SIGNAL RST H
                                        ;WILL PRESET THE RDV AND WRV FLIP-FLOP'S TO A 0.  THE SIGNAL MP H
                                        ;WILL ALLOW THE MAP PROTECTION RAM BITS MPIN H AND WRE H TO BE READ
                                        ;INTO CONTROL REGISTER 2 ALONG WITH THE SIGNAL BRK L.  THE SIGNALS
                                        ;MPIN H, WRE H, AND BRK L WILL BE READ AS ESR H, WREN H AND MSBRK H
                                        ;IN CONTROL REGISTER 2.

012072  105037  002252                  CLRB    ROLOAD                  ;CLEAR LOWER BYTE OF REG 0 FOR LOOPING
012076  052737  000005  002252          BIS     #RSTH!MPH,ROLOAD        ;SETUP TO SET RST H AND MP H
012104  042737  000060  002256          BIC     #WRVH!RDVH,ROMASK       ;SETUP TO CHECK WRV AND RDV F/F
012112  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REG 0
012116  001405                          BEQ     1$                      ;IF OK THEN CONTINUE
012120                                  ERRDF   1,,ROEROR               ;REGISTER 0 NOT EQUAL EXPECTED
012120  104455                          TRAP    C$ERDF
012122  000001                          .WORD   1
012124  000000                          .WORD   0
012126  003622                          .WORD   ROEROR
012130                                  CKLOOP
012130  104406                          TRAP    C$CLP1

                                        ;CLEAR SIGNAL RST H IN CONTROL REGISTER 0, RDV AND WRV BITS SHOULD NOT
                                        ;CHANGE STATE IN CONTROL REGISTER 0.

012132  042737  000001  002252  1$:     BIC     #RSTH,ROLOAD            ;SETUP TO CLEAR RST H IN REG 0
012140  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REG 0
012144  001405                          BEQ     2$                      ;IF EQUAL THEN CONTINUE
012146                                  ERRDF   1,,ROEROR               ;REGISTER NOT EQUAL EXPECTED
012146  104455                          TRAP    C$ERDF
012150  000001                          .WORD   1
012152  000000                          .WORD   0
012154  003622                          .WORD   ROEROR
012156                                  CKLOOP
012156  104406                          TRAP    C$CLP1

                                        ;SET SIGNAL MSEL1 H TO A ONE AND MSAD BITS 17 AND 16 TO A 0. THE
                                        ;SIGNAL MSBRK H SHOULD BE A 0 DURING THIS TEST

012160  012737  000010  002264  2$:     MOV     #MSEL1,R2LOAD           ;SETUP TO SET MSEL1 H TO A 1
012166  012737  177540  002270          MOV     #177540,R2MASK          ;SETUP TO READ MSBRK H AND LOWER 5 BITS
012174  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REG 2
```

```
3950  012200  001405                        BEQ     3$                      ;IF OK THEN CONTINUE
3951  012202                                ERRDF   2,,R2EROR               ;REG 2 NOT EQUAL EXPECTED
3952  012202  104455                        TRAP    C$ERDF
3953  012204  000002                        .WORD   2
3954  012206  000000                        .WORD   0
3955  012210  003710                        .WORD   R2EROR
3956  012212                                CKLOOP
3957  012212  104406                        TRAP    C$CLP1
3958
3959                                        ;LOAD MSAD BIT 15 THROUGH 8 TO A 0 TO SELECT ADDRESS 0
3960
3961  012214  005037  002276        3$:     CLR     R4LOAD                  ;SETUP TO SET ALL BITS TO 0
3962  012220  004737  005460                JSR     PC,LDRDR4               ;GO LOAD,READ AND CHECK REG 4
3963  012224  001405                        BEQ     4$                      ;IF NO ERRORS THEN CONTINUE
3964  012226                                ERRDF   3,,R4EROR               ;REGISTER 4 NOT ALL ZEROES
3965  012226  104455                        TRAP    C$ERDF
3966  012230  000003                        .WORD   3
3967  012232  000000                        .WORD   0
3968  012234  003776                        .WORD   R4EROR
3969  012236                                CKLOOP
3970  012236  104406                        TRAP    C$CLP1
3971
3972                                        ;SET BITS MPIN H AND MUTB H TO A ONE AND BITS WRE H AND RDE H TO
3973                                        ;A 0.  RDE H AND WRE H ON A 0 WILL SET THE RDV AND WRV FLIP-FLOPS WHEN
3974                                        ;THE SIGNAL CK H IS TOGGLED IN CONTROL REGISTER 0.
3975
3976  012240  012737  000011  002302  4$:     MOV     #MPINH!MUTBH,R6LOAD     ;SETUP TO LOAD THE MAP PROTECT RAM
3977  012246  012737  177760  002306          MOV     #177760,R6MASK          ;SETUP REGISTER 6 MASK WORD
3978  012254  004737  005504                JSR     PC,LDRDR6               ;GO LOAD,READ AND CHECK REGISTER 6
3979  012260  001405                        BEQ     5$                      ;OF COMPARED OK THEN CONTINUE
3980  012262                                ERRDF   4,MSGMP,ALINFO          ;MAP PROTECT RAM DATA ERROR
3981  012262  104455                        TRAP    C$ERDF
3982  012264  000004                        .WORD   4
3983  012266  002354                        .WORD   MSGMP
3984  012270  004050                        .WORD   ALINFO
3985  012272                                CKLOOP
3986  012272  104406                        TRAP    C$CLP1
3987
3988                                        ;CHECK CONTROL REGISTER 0 TO MAKE SURE THAT NO CHANGES OCCURED.
3989
3990  012274  004737  005360        5$:     JSR     PC,READR0               ;READ AND CHECK REGISTER 0
3991  012300  001405                        BEQ     6$                      ;IG OK THEN CONTINUE
3992  012302                                ERRDF   1,MSGMPL,ALROIN         ;REGISTER 0 CHANGED STATES
3993  012302  104455                        TRAP    C$ERDF
3994  012304  000001                        .WORD   1
3995  012306  002461                        .WORD   MSGMPL
3996  012310  004076                        .WORD   ALROIN
3997  012312                                CKLOOP
3998  012312  104406                        TRAP    C$CLP1
3999
4000                                        ;SETUP TO READ CONTROL REGISTER 2 WITH MP H SET TO A 1 IN CONTROL
4001                                        ;REGISTER 0 TO ALLOW MPIN H, WRE H AND BRK L TO BE READ INTO
4002                                        ;CONTROL REGISTER 2 AS BITS ESR H, WREN H, AND MSBRK H.
4003
4004  012314  042737  000140  002270  6$:     BIC     #ESRH!WRENH,R2MASK      ;CLEAR BITS TO BE CHECKED
4005  012322  052737  000040  002266          BIS     #ESRH,R2GOOD            ;SETUP EXPECTED MAP PROTECT BIT
```

```
                                        JSR     PC,READR2               ;GO READ REGISTER 2 AND CHECK IT
                                        BEQ     7$                      ;IF OK THEN CONTINUE
                                        ERRDF   2,MSGMPL,ALR2IN         ;REGISTER 2 NOT EQUAL EXPECTED
                                        TRAP    C$ERDF
                                        .WORD   2
                                        .WORD   MSGMPL
                                        .WORD   ALR2IN
                                        CKLOOP
                                        TRAP    C$CLP1

                                        ;SET SIGNAL CK H IN CONTRO_ REGISTER 0 TO CLOCK THE WRV AND RDV
                                        ;FLIP-FLOPS.  THE RDV AND WRV FLIP-FLOPS SHOLUD BE SET TO A 1.

                                7$:     BIS     #CKH!RDVH!WRVH,RO.OOD   ;SETUP EXPECTED DATA
                                        BIS     #CKH,ROLOAD             ;SETUP BIT TO BE LOADED
                                        JSR     PC,LDRDOR               ;GO LOAD, READ AND CHECK REG 0
                                        BEQ     8$                      ;IF OK THEN CONTINUE
                                        ERRDF   1,MSGMPL,ALROIN         ;REGISTER 0 NOT EQUAL EXPECTED
                                        TRAP    C$ERDF
                                        .WORD   1
                                        .WORD   MSGMPL
                                        .WORD   ALROIN
                                        CKLOOP
                                        TRAP    C$CLP1

                                        ;CLEAR THE SIGNAL CK H IN CONTROL REGISTER 0. NO OTHER BITS SHOULD
                                        ;CHANGE STATE.

                                8$:     BIC     #CKH,ROGOOD             ;CLEAR CKH IN EXPECTED BITS
                                        BIC     #CKH,ROLOAD             ;SETUP TO CLEAR BIT CK H IN REG 0
                                        JSR     PC,LDRDOR               ;GO CLEAR, LOAD AND CHECK REG 0
                                        BEQ     9$                      ;IF OK THEN CONTINUE
                                        ERRDF   1,MSGMPL,ALROIN         ;REG 0 NOT EQUAL EXPECTED
                                        TRAP    C$ERDF
                                        .WORD   1
                                        .WORD   MSGMPL
                                        .WORD   ALROIN
                                        CKLOOP
                                        TRAP    C$CLP1

                                        ;CHECK THAT SIGNAL MSBRK H IS SET TO A 1 IN CONTROL REGISTER 2

                                9$:     BIS     #MSBRKH!ESRH,R2GOOD     ;SETUP EXPECTED BIT IN REGISTER 2
                                        JSR     PC,READR2               ;GO READ REGISTER 2 AND CHECK IT
                                        BEQ     10$                     ;IF OK THEN CONTINUE
                                        ERRDF   2,MSGMPL,ALR2IN         ;REGISTER 2 NOT EQUAL EXPECTED
                                        TRAP    C$ERDF
                                        .WORD   2
                                        .WORD   MSGMPL
                                        .WORD   ALR2IN
                                        CKLOOP
                                        TRAP    C$CLP1

                                        ;THE FOLLOWING SECTION WILL CHECK THAT THE SIGANL RST H WILL CLEAR
                                        ;THE RDV AND WRV FLIP-FLOPS.
```

```
4062  012466  052737  000005  002252  10$:    BIS     #RSTH!MPH,ROLOAD        ;SETUP TO SET THE SIGNAL RST H
4063  012474  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REG 0
4064  012500  001405                          BEQ     11$                     ;IF REG 0 OK THEN CONTINUE
4065  012502                                   ERRDF   1,MSGMPL,ALROIN         ;RDV AND WRV PROBABLY NOT CLEARED
4066  012502  104455                          TRAP    C$ERDF
4067  012504  000001                          .WORD   1
4068  012506  002461                          .WORD   MSGMPL
4069  012510  004076                          .WORD   ALROIN
4070  012512                                   CKLOOP
4071  012512  104406                          TRAP    C$CLP1
4072
4073                                           ;CLEAR THE SIGNAL RST H IN CONTROL REGISTER 0
4074
4075  012514  042737  000001  002252  11$:    BIC     #RSTH,ROLOAD            ;SETUP TO CLEAR SIGNAL RST H
4076  012522  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD,READ AND CHECK REG 0
4077  012526  001405                          BEQ     12$                     ;IF OK THEN CONTINUE
4078  012530                                   ERRDF   1,MSGMPL,ALROIN         ;REGISTER 0 NET EQUAL EXPECTED
4079  012530  104455                          TRAP    C$ERDF
4080  012532  000001                          .WORD   1
4081  012534  002461                          .WORD   MSGMPL
4082  012536  004076                          .WORD   ALROIN
4083  012540                                   CKLOOP
4084  012540  104406                          TRAP    C$CLP1
4085
4086                                           ;READ REGISTER 2 AND CHECK THAT THE SIGNAL MSBRK H WENT TO A 0
4087
4088  012542  042737  000200  002266  12$:    BIC     #MSBRKH,R2GOOD          ;CLEAR EXPECTED BIT IN REG 2
4089  012550  004737  005426                  JSR     PC,READR2               ;GO READ AND CHECK REGISTER 2
4090  012554  001404                          BEQ     13$                     ;IF OK THEN CONTINUE
4091  012556                                   ERRDF   2,MSGMPL,A_R2IN         ;REGISTER 2 NOT EQUAL EXPECTED
4092  012556  104455                          TRAP    C$ERDF
4093  012560  000002                          .WORD   2
4094  012562  002461                          .WORD   MSGMPL
4095  012564  004124                          .WORD   ALR2IN
4096  012566                           13$:    ENDSEG
4097  012566                           10000$:
4098  012566  104405                          TRAP    C$ESEG
4099  012570                                   ENDTST
4100  012570                           L10047:
4101  012570  104401                          TRAP    C$ETST
4102
```

```
                                .SBTTL TEST 17: CHECK MODULE SELECT RAM 0 USING A BINARY COUNT

                                ;++
                                ; TEST TO CHECK MODULE SELECT RAM 0. A BINARY COUNT PATTERN WILL BE LOADED
                                ; INTO EACH ADDRESS OF THE MODULE SELECT RAM 0.  FOR EACH PATTERN LOADED THE
                                ; TEST WILL CHECK FOR THE APPROPRIATE ENABLES ON THE OUTPUT OF THE RAM. IF
                                ; MORE THEN ONE BIT IS SET ON THE INPUT TO THE RAM, ONLY ONE BIT WILL BE SET ON
                                ; THE OUTPUT OF THE RAM. A ONE IN THE LEAST SIGNIFICANT BIT OF THE RAM WILL
                                ; FORCE THE REMAINING MORE SIGNIFICANT BITS TO A ZERO. MODULE SELECT RAM 0
                                ; IS SELECTED BY SETTING THE SIGNAL MSEL0 H TO 1 AND MSEL1 H TO A 0 IN CONTROL
                                ; REGISTER 2 AND THEN DOING A WRITE OR READ TO CONTROL REGISTER 6 WHICH WILL
                                ; ASSERT THE SIGNAL SMDS0 L. THE SIGNAL SMDS0 L WILL SELECT MODULE SELECT RAM 0.
                                ;--

012572                                  BGNTST
012572                          T17::
012572  004737  004546                  JSR     PC,INITMS               ;SELECT AND INIT THE MEMORY SIMULATOR
012576  005001                          CLR     R1                      ;CLEAR MSAD BITS 15-13
012600  005002                          CLR     R2                      ;SET DATA PATTERN INITIALLY TO 0

012602                          1$:     BGNSEG
012602  104404                          TRAP    C$BSEG

                                        ;SET SIGNAL MSEL0 H TO A ONE AND SIGNALS MSEL1 H, MSAD16 H AND MSAD17 H
                                        ;TO A 0 IN CONTROL REGISTER 2. THE SIGNAL MSEL0 H ON A 1 WILL CAUSE
                                        ;THE SIGNAL SMDS0 L TO BE ASSERTED ON A WRITE OR READ TO REGISTER 6.
                                        ;THE SIGNAL SMDS0 L WILL SELECT MODULE SELECT RAM 0.

012604  012737  000004  002264          MOV     #MSEL0,R2LOAD           ;SETUP TO SET MSEL0 H TO A 1
012612  012737  177740  002270          MOV     #177740,R2MASK          ;SETUP MASK WORD TO IGNORE BITS 15-5
012620  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REG 2
012624  001405                          BEQ     2$                      ;IF LOADED OK THEN CONTINUE
012626                                  ERRDF   2,,R2EROR               ;REG 2 NOT EQUAL EXPECTED DATA
012626  104455                          TRAP    C$ERDF
012630  000002                          .WORD   2
012632  000000                          .WORD   0
012634  003710                          .WORD   R2EROR
012636                                  CKLOOP
012636  104406                          TRAP    C$CLP1

                                        ;SET MSAD BITS 15-13 IN CONTROL REGISTER 4 TO THE ADDRESS TO BE TESTED.
                                        ;MSAD BITS 15-13 SELECT MODULE SELECT RAM 0 ADDRESSES.

012640  010137  002276          2$:     MOV     R1,R4LOAD               ;SETUP BITS TO BE LOADED
012644  004737  005460                  JSR     PC,LDRDR4               ;GO LOAD, READ AND CHECK REGISTER 4
012650  001405                          BEQ     3$                      ;IF LOADED OK THEN CONTINUE
012652                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED DATA
012652  104455                          TRAP    C$ERDF
012654  000003                          .WORD   3
012656  000000                          .WORD   0
012660  003776                          .WORD   R4EROR
012662                                  CKLOOP
012662  104406                          TRAP    C$CLP1

                                        ;THE FOLLOWING TEST WILL WRITE THE 4 BIT DATA PATTERN INTO THE RAM
                                        ;LOCATION ADDRESSED BY MSAD BITS 15-13, AND CHECK THAT THE CORRECT
```

```
4159                                        ;PATTERN IS READ BACK.  ON A WRITE OR READ COMMAND TO CONTROL
4160                                        ;REGISTER 6 WITH THE SIGNAL MSEL0 H A 1 AND SIGNAL MSEL1 H A 0, THE
4161                                        ;SIGNAL SMDS0 L WILL BE ASSERTED WHICH WILL SELECT MODULE SELECT RAM 0.
4162
4163  012664  012737  177760  002306  3$:   MOV     #177760,R6MASK          ;SETUP REGISTER 6 MASK WORD
4164  012672  010237  002302                MOV     R2,R6LOAD               ;SETUP DATA PATTERN TO BE LOADED
4165  012676  010203                       MOV     R2,R3                   ;COPY DATA TO BE LOADED
4166  012700  032703  000001               BIT     #BIT0,R3                ;CHECK IF BIT 0 IS SET
4167  012704  001403                       BEQ     4$                      ;IF NOT GO CHECK BIT 1
4168  012706  042703  000016               BIC     #BIT3.BIT2.BIT1,R3      ;CLEAR MOST SIGNIFICANT BITS
4169  012712  000413                       BR      6$                      ;GO DO THE TEST
4170  012714  032703  000002          4$:  BIT     #BIT1,R3                ;CHECK IF BIT 1 IS SET
4171  012720  001403                       BEQ     5$                      ;IF NOT THEN GO CHECK CHECK BIT 2
4172  012722  042703  000014               BIC     #BIT3 BIT2,R3           ;CLEAR MORE SIGNIFICANT BITS
4173  012726  000405                       BR      6$                      ;GO START THE TEST
4174  012730  032703  000004          5$:  BIT     #BIT2,R3                ;CHECK IF BIT 2 IS SET
4175  012734  001402                       BEQ     6$                      ;IF NOT THE GO DO THE TEST
4176  012736  042703  000010               BIC     #BIT3,R3                ;CLEAR MOST SIGNIFICANT BIT
4177  012742  010337  002304          6$:  MOV     R3,R6GOOD               ;SAVE EXPECTED PATTERN
4178  012746  004737  005512               JSR     PC,LDRD6R               ;GO LOAD, READ, AND CHECK REG 6
4179  012752  001404                       BEQ     7$                      ;IF DATA OK THEN CONTINUE
4180  012754                               ERRDF   4,MSGMD0,ALINF0         ;DATA ERROR IN MODULE SELECT RAM 0
4181  012754  104455                       TRAP    C$ERDF
4182  012756  000004                       .WORD   4
4183  012760  002511                       .WORD   MSGMD0
4184  012762  004050                       .WORD   ALINF0
4185
4186  012764                          7$:  ENDSEG
4187  012764                          10000$:
4188  012764  104405                       TRAP    C$ESEG
4189  012766  005202                       INC     R2                      ;UPDATE DATA PATTERN BY 1
4190  012770  022702  000020               CMP     #20,R2                  ;CHECK IF DONE
4191  012774  001302                       BNE     1$                      ;IF NOT THEN DO NEXT PATTERN
4192  012776  005002                       CLR     R2                      ;RESET PATTERN TO 0
4193
4194  013000  062701  020000               ADD     #MSAD13,R1              ;UPDATE MODULE SELECT RAM ADDRESS BY 1
4195  013004  001276                       BNE     1$                      ;IF NOT DONE REPEAT THE TEST
4196  013006                               ENDTST
4197  013006                          L10050:
4198  013006  104401                       TRAP    C$ETST
4199
```

```
                                .SBTTL TEST 18: CHECK MODULE SELECT RAM 1 USING A BINARY COUNT

                                ;++
                                ; TEST TO CHECK MODULE SELECT RAM 1. A BINARY COUNT PATTERN WILL BE LOADED INTO
                                ; EACH ADDRESS OF THE MODULE SELECT RAM 1.  FOR EACH PATTERN LOADED, THE TEST
                                ; WILL CHECK FOR THE APPROPRIATE ENABLES ON THE OUPUT OF THE RAM.  IF MORE THEN
                                ; ONE BIT IS SET ON THE INPUT TO THE RAM, ONLY ONE BIT WILL BE SET ON THE
                                ; OUTPUT OF THE RAM.  A ONE IN THE LEAST SIGNIFICANT BITS OF THE RAM WILL
                                ; FORCE THE MORE SIGNIFICANT BITS TO ZEROES.  MODULE SELECT RAM 1 IS SELECTED
                                ; BY SETTING THE SIGNALS MSEL0 H AND MSEL1 H TO A ONE IN CONTROL REGISTER 2
                                ; AND THEN DOING A WRITE OR READ TO CONTROL REGISTER 6 WHICH WILL ASSERT THE
                                ; SIGNAL SMDS1 L. THE SIGNAL SMDS1 L WILL ENABLE MODULE SELECT RAM 1 TO BE
                                ; WRITTEN OR READ.  MSAD BITS 17 AND 16 WILL BE USED TO ADDRESS THE MODULE
                                ; SELECT RAM 1.
                                ;--

013010                                  BGNTST
013010                          T18::
013010  004737  004546                  JSR     PC,INITMS               ;SELECT AND INIT THE MEMORY SIMULATOR
013014  012701  000014                  MOV     #MSEL0.MSEL1,R1         ;SETUP BITS FOR CONTROL REGISTER 2
013020  005002                          CLR     R2                      ;SET DATA PATTERN INITIALLY TO 0

013022                          1$:     BGNSEG
013022  104404                          TRAP    C$BSEG

                                        ;SET SIGNAL MSEL0 H AND MSEL1 H TO A ONE AND MSAD BITS 17 AND 16
                                        ;TO THE ADDRESS TO BE TESTED.  THE SIGNALS MSEL0 H AND MSEL1 H WILL
                                        ;CAUSE THE SIGNAL SMDS1 L TO BE ASSERTED ON A WRITE OR READ TO
                                        ;CONTROL REGISTER 6.  THE SIGNAL SMDS1 L WILL ENABLE MODULE SELECT
                                        ;RAM 1.

013024  010137  002264                  MOV     R1,R2LOAD               ;SETUP ADDRESS + SET MSEL0 AND MSEL1    1
013030  012737  177740  002270          MOV     #177740,R2MASK          ;SETUP REGISTER 2 MASK WORD
013036  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REGISTER 2
013042  001405                          BEQ     2$                      ;IF LOADED OK THEN CONTINUE
013044                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
013044  104455                          TRAP    C$ERDF
013046  000002                          .WORD   2
013050  000000                          .WORD   0
013052  003710                          .WORD   R2EROR
013054                                  CKLOOP
013054  104406                          TRAP    C$CLP1

                                        ;THE FOLLOWING TEST WILL WRITE THE 4 BIT DATA PATTERN INTO THE RAM
                                        ;LOCATION ADDRESSED BY MSAD BITS 17 AND 16, AND CHECK THAT THE
                                        ;CORRECT PATTERN IS READ BACK.  ON A WRITE OR READ COMMAND TO
                                        ;CONTROL REGISTER 6 WITH THE SIGNALS MSEL0 H AND MSEL1 H SET TO A 1,
                                        ;THE SIGNAL SMDS1 L WILL BE ASSERTED TO ENABLE MODULE SELECT RAM 1.

013056  012737  177760  002306  2$:     MOV     #177760,R6MASK          ;SETUP REGISTER 6 MASK WORD
013064  010237  002302                  MOV     R2,R6LOAD               ;SETUP PATTERN TO BE LOADED
013070  010203                          MOV     R2,R3                   ;COPY DATA PATTERN TO BE LOADED
013072  032703  000001                  BIT     #BIT0,R3                ;CHECK IF BIT 0 IS SET
013076  001403                          BEQ     3$                      ;IF NOT GO CHECK BIT 1
013100  042703  000016                  BIC     #BIT3.BIT2.BIT1,R3      ;IF YES - 0 MOST SIGNIFICANT BITS
013104  000413                          BR      5$                      ;GO LOAD THE DATA PATTERN
```

```
4256  013106  032703  000002          3$:     BIT     #BIT1,R3                ;CHECK IF BIT 1 IS SET
4257  013112  001403                          BEQ     4$                      ;IF NOT THEN CHECK BIT 2
4258  013114  042703  000014                  BIC     #BIT3.BIT2,R3           ;IF YES - 0 MOST SIGNIFICANT BITS
4259  013120  000405                          BR      5$                      ;GO LOAD THE DATA PATTERN
4260  013122  032703  000004          4$:     BIT     #BIT2,R3                ;CHECK IF BIT 2 IS SET
4261  013126  001402                          BEQ     5$                      ;IF NOT GO LOAD THE DATA PATTERN
4262  013130  042703  000010                  BIC     #BIT3,R3                ;CLEAR MOST SIGNIFICANT BITS
4263  013134  010337  002304          5$:     MOV     R3,R6GOOD               ;SAVE EXPECTED DATA PATTERN
4264  013140  004737  005512                  JSR     PC,LDRD6R               ;GO LOAD, READ AND CHECK DATA
4265  013144  001404                          BEQ     6$                      ;IF DATA OK THEN CONTINUE
4266  013146                                  ERRDF   4,MSGMD1,ALINFO         ;DATA ERROR IN MODULE SELECT RAM 1
4267  013146  104455                          TRAP    C$ERDF
4268  013150  000004                          .WORD   4
4269  013152  002553                          .WORD   MSGMD1
4270  013154  004050                          .WORD   ALINFO
4271  013156                          6$:     ENDSEG
4272  013156                          10000$:
4273  013156  104405                          TRAP    C$ESEG
4274  013160  005202                          INC     R2                      ;UPDATE THE DATA PATTERN
4275  013162  022702  000020                  CMP     #20,R2                  ;CHECK IF BINARY COUNT DONE
4276  013166  001315                          BNE     1$                      ;IF NOT LOAD NEXT PATTERN
4277  013170  005002                          CLR     R2                      ;OTHERWISE CLEAR DATA PATTERN
4278  013172  005201                          INC     R1                      ;UPDATE MSAD BITS 17 AND 16
4279  013174  032701  000020                  BIT     #BIT4,R1                ;CHECK IF DONE
4280  013200  001710                          BEQ     1$                      ;IF NOT THEN CHECK NEXT ADDRESS
4281  013202                                  ENDTST
4282  013202                          L10051:
4283  013202  104401                                  TRAP    C$ETST
4284
```

```
                                .SBTTL TEST 19: CHECK MODULE SELECT RAM 0 + 1 CHIP SELECT LOGIC

                                ;++
                                ; TEST TO CHECK THAT MODULE SELECT RAM 0 AND 1 ARE ACTUALLY SELECTED WHEN
                                ; THE SIGNALS MSEL0 H AND MSEL1 H ARE SET TO SELECT THEM. THE TEST WILL
                                ; SELECT MODULE SELECT RAM 0, ADDRESS 0, AND WRITE A DATA PATTERN OF 1 INTO
                                ; IT. THE TEST WILL THEN SELECT MODULE SELECT RAM 1, ADDRESS 0, AND WRITE A DATA
                                ; PATTERN OF 10 INTO IT. THE TEST WILL THEN RESELECT MODULE SELECT RAM 0 AND
                                ; CHECK THE PATTERN TO BE 1 AND THEN SELECT MODULE SELECT RAM 1 AND CHECK THE
                                ; PATTERN TO BE 10.
                                ;--

013204                                  BGNTST
013204                          T19::
013204  004737  004546                   JSR     PC,INITMS               ;SELECT AND INIT MEMORY SIMULATOR

013210                                  BGNSEG
013210  104404                          TRAP    C$BSEG

                                ;SET SIGNAL MSEL0 H TO A ONE AND SIGNALS MSEL1 H, MSAD17 H AND
                                ;MSAD16 H TO A ZERO IN CONTROL REGISTER 2.  THE SIGNAL MSEL0 H ON A
                                ;ONE WILL CAUSE THE SIGNAL SMDS0 L TO BE ASSERTED ON A WRITE OR READ
                                ;TO CONTROL REGISTER 6.  THE SIGNAL SMDS0 L WILL SELECT MODULE SELECT
                                ;RAM 0.

013212  012737  000004  002264          MOV     #MSEL0,R2LOAD           ;SETUP TO SET MSEL0 H TO A 1
013220  012737  177740  002270          MOV     #177740,R2MASK          ;SETUP REGISTER 2 MASK WORD
013226  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REGISTER 2
013232  001405                          BEQ     1$                      ;IF LOADED OK THEN CONTINUE
013234                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED DATA
013234  104455                          TRAP    C$ERDF
013236  000002                          .WORD   2
013240  000000                          .WORD   0
013242  003710                          .WORD   R2EROR
013244                                  CKLOOP
013244  104406                          TRAP    C$CLP1

                                ;CLEAR CONTROL REGISTER 4 TO SET MODULE SELECT ADDRESSES TO 0. MSAD
                                ;BITS 15-13 SELECT MODULE SELECT RAM 0 ADDRESSES.

013246  005037  002276          1$:     CLR     R4LOAD                  ;SETUP TO CLEAR MSAD BITS 15-0
013252  004737  005460                  JSR     PC,LDRDR4               ;GO LOAD, READN AND CHECK REGISTER 4
013256  001405                          BEQ     2$                      ;IF LOADED OK THEN CONTINUE
013260                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED DATA
013260  104455                          TRAP    C$ERDF
013262  000003                          .WORD   3
013264  000000                          .WORD   0
013266  003776                          .WORD   R4EROR
013270                                  CKLOOP
013270  104406                          TRAP    C$CLP1

                                ;WRITE DATA PATTERN OF 1 INTO MODULE SELECT RAM 0. ON A WRITE OR READ
                                ;TO CONTROL REGISTER 6 WITH THE SIGNALS MSEL0 H A 1 AND MSEL1 H A 0,
                                ;THE SIGNAL SMDS0 L SHOULD BE ASSERTED TO SELECT MODULE SELECT RAM 0.

013272  012737  177760  002306  2$:     MOV     #177760,R6MASK          ;SETUP REGISTER 6 MASK WORD
```

```
4341  013300  012737  000002  002302          MOV     #BIT1,R6LOAD            ;SETUP DATA PATTERN OF 1
4342  013306  004737  005504                  JSR     PC,LDRDR6               ;GO LOAD, READ AND CHECK RAM
4343  013312  001405                          BEQ     3$                      ;IF DATA OK THEN CONTINUE
4344  013314                                  ERRDF   4,MSGMD0,ALINFO         ;DATA ERROR IN MODULE SELECT RAM 0
4345  013314  104455                          TRAP    C$ERDF
4346  013316  000004                          .WORD   4
4347  013320  002511                          .WORD   MSGMD0
4348  013322  004050                          .WORD   ALINFO
4349  013324                                  CKLOOP
4350  013324  104406                          TRAP    C$CLP1
4351
4352                                          ;SET SIGNALS MSEL0 H AND MSEL1 H TO A ONE AND MSAD BITS 17 AND 16
4353                                          ;TO A 0 TO SELECT ADDRESS 0. THE SIGNALS MSEL0 H AND MSEL1 H BEING
4354                                          ;SET WILL CAUSE THE SIGNAL SMDS1 L TO BE ASSERTED ON A WRITE OR
4355                                          ;READ TO CONTROL REGISTER 6. THE SIGNAL SMDS1 L WILL ENABLE MODULE
4356                                          ;SELECT RAM 1.
4357
4358  013326  012737  000014  002264  3$:     MOV     #MSEL0!MSEL1,R2LOAD     ;SETUP MODULE SELECT BITS AND ADDRESS
4359  013334  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REG 2
4360  013340  001405                          BEQ     4$                      ;IF LOADED OK THEN CONT
4361  013342                                  ERRDF   2,,R2EROR               ;REG 2 NOT EQUAL EXPECTED DATA
4362  013342  104455                          TRAP    C$ERDF
4363  013344  000002                          .WORD   2
4364  013346  000000                          .WORD   0
4365  013350  003710                          .WORD   R2EROR
4366  013352                                  CKLOOP
4367  013352  104406                          TRAP    C$CLP1
4368
4369                                          ;WRITE DATA PATTERN OF 10 INTO MODULE SELECT RAM 1. ON A WRITE OR
4370                                          ;READ TO CONTROL REGISTER 6 WITH THE SIGNALS MSEL0 H AND MSEL1 H
4371                                          ;SET, THE SIGNAL SMDS1 L SHOULD BE ASSERTED TO SELECT MODE SELECT
4372                                          ;RAM 1.
4373
4374  013354  012737  000010  002302  4$:     MOV     #BIT3,R6LOAD            ;SET DATA PATTERN TO 10
4375  013362  004737  005504                  JSR     PC,LDRDR6               ;GO LOAD, READ AND CHECK DATA
4376  013366  001405                          BEQ     5$                      ;IF DATA OK THEN CONTINUE
4377  013370                                  ERRDF   4,MSGMD1,ALINFO         ;DATA ERROR IN MODULE SELECT RAM 1
4378  013370  104455                          TRAP    C$ERDF
4379  013372  000004                          .WORD   4
4380  013374  002553                          .WORD   MSGMD1
4381  013376  004050                          .WORD   ALINFO
4382  013400                                  CKLOOP
4383  013400  104406                          TRAP    C$CLP1
4384
4385                                          ;RESELECT MODULE SELECT RAM 0 BY SETTING MSEL0 H TO A ONE AND
4386                                          ;MSEL1 H TO A 0 IN CONTROL REGISTER 2.
4387
4388  013402  012737  000004  002264  5$:     MOV     #MSEL0,R2LOAD           ;SETUP TO CLEAR BIT MSEL1 H
4389  013410  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REG 2
4390  013414  001405                          BEQ     6$                      ;IF LOADED OK THEN CONTINUE
4391  013416                                  ERRDF   2,,R2EROR               ;REG 2 NOT EQUAL EXPECTED DATA
4392  013416  104455                          TRAP    C$ERDF
4393  013420  000002                          .WORD   2
4394  013422  000000                          .WORD   0
4395  013424  003710                          .WORD   R2EROR
4396  013426                                  CKLOOP
```

```
4397  013426  104406                          TRAP    C$CLP1
4398
4399                                          ;READ MODULE SELECT RAM 0 BY READING CONTROL REGISTER 6. ISSUING
4400                                          ;A READ COMMAND WHEN MSEL0 H IS SET AND MSEL1 H IS CLEARED WILL
4401                                          ;ASSERT THE SIGNAL SMDS0 L. THE SIGNAL SMDS0 L BEING ASSERTED WILL
4402                                          ;SELECT MODULE SELECT RAM 0.
4403
4404  013430  012737  000002  002302  6$:     MOV     #BIT1,R6LOAD            ;SETUP PREVIOUSLY LOADED DATA
4405  013436  013737  002302  002304          MOV     R6LOAD,R6GOOD           ;SETUP EXPECTED DATA PATTERN
4406  013444  004737  005520                  JSR     PC,READR6               ;GO READ AND CHECK RAM LOCATION
4407  013450  001405                          BEQ     7$                      ;IF DATA OK THEN CONT
4408  013452                                  ERRDF   4,MSGMDC,ALINFO         ;PROBABLY CHIP ENABLE ERROR - MODULE
4409  013452  104455                          TRAP    C$ERDF
4410  013454  000004                          .WORD   4
4411  013456  002615                          .WORD   MSGMDC
4412  013460  004050                          .WORD   ALINFO
4413  013462                                  CKLOOP
4414  013462  104406                          TRAP    C$CLP1
4415                                                                  ;SELECT RAM
4416
4417                                          ;RESELECT MODULE SELECT RAM 1 BY SETTING SIGNALS MSEL0 H AND MSEL1 H
4418                                          ;TO A ONE  IN COTROL REGISTER 2.
4419
4420  013464  012737  000014  002264  7$:     MOV     #MSEL0!MSEL1,R2LOAD     ;SETUP BITS TO BE LOADED
4421  013472  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REG 2
4422  013476  001405                          BEQ     8$                      ;IF LOADED OK THEN CONT
4423  013500                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
4424  013500  104455                          TRAP    C$ERDF
4425  013502  000002                          .WORD   2
4426  013504  000000                          .WORD   0
4427  013506  003710                          .WORD   R2EROR
4428  013510                                  CKLOOP
4429  013510  104406                          TRAP    C$CLP1
4430
4431                                          ;READ MODULE SELECT RAM 1 BY READING CONTROL REGISTER 6. ISSUING
4432                                          ;A READ COMMAND TO CONTROL REGISTER 6 WITH MSEL0 H AND MSEL1 H SET
4433                                          ;WILL ASSERT THE SIGNAL SMDS1 L. THIS SIGNAL BEING ASSERTED WILL
4434                                          ;SELECT MODULE SELECT RAM 1.
4435
4436  013512  012737  000010  002302  8$:     MOV     #BIT3,R6LOAD            ;SETUP PREVIOUSLY LOADED DATA
4437  013520  013737  002302  002304          MOV     R6LOAD,R6GOOD           ;SETUP EXPECTED DATA
4438  013526  004737  005520                  JSR     PC,READR6               ;GO READ AND CHECK THE DATA
4439  013532  001404                          BEQ     9$                      ;IF DATA OK THEN CONT
4440  013534                                  ERRDF   4,MSGMDC,ALINFO         ;PROBABLY CHIP ENABLE ERROR
4441  013534  104455                          TRAP    C$ERDF
4442  013536  000004                          .WORD   4
4443  013540  002615                          .WORD   MSGMDC
4444  013542  004050                          .WORD   ALINFO
4445  013544                          9$:     ENDSEG
4446  013544                          10000$:
4447  013544  104405                          TRAP    C$ESEG
4448  013546                                  ENDTST
4449  013546                          L10052:
4450  013546  104401                          TRAP    C$ETST
4451
```

```
                                       .SBTTL TEST 20: CHECK EACH ADDRESS IN MOD SEL RAM 0 TO BE ADDRESSED

                                       ;++
                                       ; TEST TO CHECK THAT EACH ADDRESS IN MODULE SELECT RAM 0 CAN BE ADDRESSED
                                       ; CORRECTLY. THIS IS DONE BY WRITING A SPECIFIC PATTERN INTO THE RAM AND THEN
                                       ; READING THE RAM CHECKING THAT NO LOCATIONS CHANGED.  THE PATTERNS LOADED
                                       ; INTO THE RAM, STARTING AT THE LOWEST ADDRESS, ARE AS FOLLOWS: 10, 0., 02, 01,
                                       ; 00, 01, 10, 04.
                                       ;--

013550                                         BGNTST
013550                                 T20::
013550  004737  004546                         JSR     PC,INITMS               ;SELECT AND INIT THE MEMORY SIMULATOR
013554  005001                                 CLR     R1                      ;CLEAR WORKING MSAD BITS
013556  005002                                 CLR     R2                      ;CLEAR OFFSET TO DATA TABLE

013560                                 1$:     BGNSEG
013560  104404                                 TRAP    C$BSEG

                                               ;SET SIGNAL MSEL0 H TO A ONE, AND SIGNALS MSEL1 H, MSAD17 H AND MSAD16 H
                                               ;TO A ZERO IN CONTROL REGISTER 2.

013562  012737  000004  002264                 MOV     #MSEL0,R2LOAD           ;SETUP TO SET MSEL0 H TO A 1
013570  012737  177740  ^^2270                 MOV     #177740,R2MASK          ;SETUP REGISTER 2 MASK WORD
013576   04737  0054'                          JSR     PC,LDRDR2               ;GO LOAD,READ AND CHECK REG 2
013602  001405                                 BEQ     2$                      ;IF LOADED OK THEN CONTINUE
013604                                         ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
013604  104455                                 TRAP    C$ERDF
013606  000002                                 .WORD   2
013610  000000                                 .WORD   0
013612  003710                                 .WORD   R2EROR
013614                                         CKLOOP
013614  104406                                 TRAP    C$CLP1

                                               ;SET MSAD BITS 15-13 IN CONTROL REGISTER 4 TO THE ADDRESS TO BE
                                               ;TESTED. MSAD BITS 15-13 ARE USED TO SELECT MODULE SELECT RAM 0
                                               ;ADDRESSES.

013616  010137  002276                 2$:     MOV     R1,R4LOAD               ;SETUP BITS TO BE LOADED
013622  004737  005460                         JSR     PC,LDRDR4               ;GO LOAD, READ AND CHECK REGISTER 4
013626  001405                                 BEQ     3$                      ;IF LOADED OK THEN CONTINUE
013630                                         ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
013630  104455                                 TRAP    C$ERDF
013632  000003                                 .WORD   3
013634  000000                                 .WORD   0
013636  003776                                 .WORD   R4EROR
013640                                         CKLOOP
013640  104406                                 TRAP    C$CLP1

                                               ;THE FOLLOWING SECTION OF CODE WILL WRITE THE DATA PATTERN INTO RAM
                                               ;AND CHECK THAT THE CORRECT PATTERN WAS WRITTEN.  ON A WRITE OR READ
                                               ;COMMAND TO CONTROL REGISTER 6 WITH SIGNAL MSEL0 H SET AND MSEL1 H
                                               ;CLEARED, THE SIGNAL SMDS0 L WILL BE ASSERTED TO SELECT MODULE SELECT
                                               ;RAM 0.

013642  016237  014036  002302         3$:     MOV     MSRODT(R2),R6LOAD       ;GET DATA PATTERN FROM TABLE
```

```
4508  013650  012737  177760  002306          MOV     #177760,R6MASK          ;SETUP REGISTER 6 MASK WORD
4509  013656  004737  005504                  JSR     PC,LDRDR6               ;GO LOAD,READ AND CHECK RAM LOCATION
4510  013662  001404                          BEQ     4$                      ;IF OK THEN CONTINUE
4511  013664                                  ERRDF   4,MSGMD0,ALINFO         ;DATA ERROR IN MODULE SELECT RAM 0
4512  013664  104455                          TRAP    C$ERDF
4513  013666  000004                          .WORD   4
4514  013670  002511                          .WORD   MSGMD0
4515  013672  004050                          .WORD   ALINFO
4516  013674                          4$:     ENDSEG
4517  013674                          10000$:
4518  013674  104405                          TRAP    C$ESEG
4519  013676  005722                          TST     (R2)+                   ;UPDATE OFFSET POINTER TO DATA TABLE
4520  013700  062701  020000                  ADD     #MSAD13,R1              ;UPDATE MODULE SELECT RAM 0 ADDRESS
4521  013704  001325                          BNE     1$                      ;IF NOT 0 THEN DO NEXT ADDRESS
4522
4523  013706  005001                          CLR     R1                      ;RESET WORKING MSAD ADDRESSES TO 0
4524  013710  005002                          CLR     R2                      ;CLEAR OFFSET POINTER TO DATA TABLE
4525
4526  013712                          5$:     BGNSEG
4527  013712  104404                          TRAP    C$BSEG
4528
4529                                          ;SET SIGNAL MSEL0 H TO A ONE AND CLEAR SIGNALS MSEL1 H, MSAD17 H AND
4530                                          ;MSAD16 H. THESE BITS ARE IN CONTROL REGISTER 2.
4531
4532  C13714  012737  000004  002264          MOV     #MSEL0,R2LOAD           ;SETUP TO SET MSEL0 H TO A 1
4533  013722  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REG 2
4534  013726  001405                          BEQ     6$                      ;IF LOADED OK THEN CONTINUE
4535  013730                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
4536  013730  104455                          TRAP    C$ERDF
4537  013732  000002                          .WORD   2
4538  013734  000000                          .WORD   0
4539  013736  003710                          .WORD   R2EROR
4540  013740                                  CKLOOP
4541  013740  104406                          TRAP    C$CLP1
4542
4543                                          ;SET MSAD BITS 15-13 IN CONTROL REGISTER 4 TO THE ADDRESS TO BE CHECKED.
4544
4545  013742  010137  002276          6$:     MOV     R1,R4LOAD               ;SETUP BITS TO BE LOADED
4546  013746  004737  005460                  JSR     PC,LDRDR4               ;GO LOAD, READ AND CHECK REGISTER 4
4547  013752  001405                          BEQ     7$                      ;IF LOADED OK THEN CONTINUE
4548  013754                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
4549  013754  104455                          TRAP    C$ERDF
4550  013756  000003                          .WORD   3
4551  013760  000000                          .WORD   0
4552  013762  003776                          .WORD   R4EROR
4553  013764                                  CKLOOP
4554  013764  104406                          TRAP    C$CLP1
4555
4556                                          ;THE FOLLOWING SECTION WILL VALIDATE THAT THE DATA WRITTEN IN THE
4557                                          ;BEGINNING OF THIS TEST REMAINS UNCHANGED. IF AN ERROR OCCURS, THE
4558                                          ;PROBLEM IS PROBABLY INTERNAL TO THE RAM CHIP.  ON A READ COMMAND
4559                                          ;TO CONTROL REGISTER 6 WITH THE SIGNAL MSEL0 H SET AND MSEL1 H CLEARED,
4560                                          ;THE SIGNAL SMDS0 L WILL BE ASSERTED TO SELECT MODULE SELECT RAM 0.
4561
4562  013766  016237  014036  002302  7$:     MOV     MSRODT(R2),R6LOAD       ;SETUP PREVIOUSLY LOADED DATA
4563  013774  013737  002302  002304          MOV     R6LOAD,R6GOOD           ;SETUP EXPECTED DATA
```

```
 4564  014002  004737  005520                  JSR     PC,READR6               ;GO READ AND CHECK RAM LOCATION
 4565  014006  001404                          BEQ     8$                      ;IF DATA OK THEN CONTINUE
 4566  014010                                  ERRDF   4,MSGMDA,ALINFO         ;MODULE SELECT ADDRESSING ERROR
 4567  014010  104455                          TRAP    C$ERDF
 4568  014012  000004                          .WORD   4
 4569  014014  002665                          .WORD   MSGMDA
 4570  014016  004050                          .WORD   ALINFO
 4571  014020                          8$:     ENDSEG
 4572  014020                          10001$:
 4573  014020  104405                          TRAP    C$ESEG
 4574
 4575  014022  005722                          TST     (R2)+                   ;UPDATE OFFSET POINTER TO DATA TABLE
 4576  014024  062701  020000                  ADD     #MSAD13,R1              ;UPDATE MODULE SELECT ADDRESS
 4577  014030  001330                          BNE     5$                      ;IF NOT DONE CHECK NEXT ADDRESS
 4578  014032                                  EXIT    TST
 4579  014032  104432                          TRAP    C$EXIT
 4580  014034  000022                          .WORD   L10053-.
 4581
 4582  014036  000010                  MSRODT: .WORD   10
 4583  014040  000004                          .WORD   4
 4584  014042  000002                          .WORD   2
 4585  014044  000001                          .WORD   1
 4586  014046  000000                          .WORD   0
 4587  014050  000001                          .WORD   1
 4588  014052  000010                          .WORD   10
 4589  014054  000004                          .WORD   4
 4590
 4591  014056                                  ENDTST
 4592  014056                          L10053:
 4593  014056  104401                          TRAP    C$ETST
 4594
```

TEST 21: CHECK EACH ADDRESS IN MOD SEL RAM 1 TO BE ADDRESSED SEQ 0091

```
                                .SBTTL TEST 21: CHECK EACH ADDRESS IN MOD SEL RAM 1 TO BE ADDRESSED

                                ;++
                                ; TEST TO CHECK THAT EACH ADDRESS IN MODULE SELECT RAM 1 CAN BE ADDRESSED
                                ; CORRECTLY.  THIS IS DONE BY WRITING A SPECIFIC PATTERN INTO THE RAM AND THEN
                                ; READING THE RAM CHECKING THAT NO CHANGES OCCURED. THE PATTERN LOADED INTO
                                ; THE RAM, STARTING AT THE LOWEST ADDRESS, ARE AS FOLLOWS: 01, 02, 04, 10.
                                ;--

014060                                  BGNTST
014060                          T21::
014060  004737  004546                  JSR     PC,INITMS               ;SELECT AND INIT THE MEMORY SIMULATOR
014064  012701  000014                  MOV     #MSEL0!MSEL1,R1         ;SETUP BITS FOR CONTROL REGISTER 2
014070  005002                          CLR     R2                      ;CLEAR OFFSET POINTER TO DATA TABLE

014072                          1$:     BGNSEG
014072  104404                          TRAP    C$BSEG

                                ;SET SIGNALS MSEL0 H AND MSEL1 H TO A ONE, SET SIGNALS MSAD 17 AND 16
                                ;TO THE ADDRESS TO BE TESTED.

014074  010137  002264                  MOV     R1,R2LOAD               ;SETUP BITS TO BE LOADED
014100  012737  177740  002270          MOV     #177740,R2MASK          ;SETUP REGISTER 2 MASK WORD
014106  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REGISTER 2
014112  001405                          BEQ     2$                      ;IF LOADED OK THEN CONTINUE
014114                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED DATA
014114  104455                          TRAP    C$ERDF
014116  000002                          .WORD   2
014120  000000                          .WORD   0
014122  003710                          .WORD   R2EROR
014124                                  CKLOOP
014124  104406                          TRAP    C$CLP1

                                ;THE FOLLOWING SECTION WILL WRITE THE DATA PATTERN INTO THE RAM AND
                                ;CHECK THAT THE CORRECT PATTERN WAS WRITTEN. ON A WRITE OR READ COMMAND
                                ;TO CONTROL REGISTER 6 WITH SIGNALS MSEL0 H AND MSEL1 H SET TO A ONE,
                                ;THE SIGNAL SMDS1 L WILL BE ASSERTED TO SELECT MODULE SELECT RAM 1.

014126  016237  014302  002302    2$:   MOV     MSR1DT(R2),R6LOAD       ;GET DATA PATTERN TO BE WRITTEN
014134  012737  177760  002306          MOV     #177760,R6MASK          ;SETUP MASK WORD FOR REGISTER 6
014142  004737  005504                  JSR     PC,LDRDR6               ;GO LOAD, READ AND CHECK RAM LOCATION
014146  001404                          BEQ     3$                      ;IF DATA OK THEN GO DO NEXT LOCATION
014150                                  ERRDF   4,MSGMD1,ALINFO         ;DATA ERROR IN MODULE SELECT RAM 1
014150  104455                          TRAP    C$ERDF
014152  000004                          .WORD   4
014154  002553                          .WORD   MSGMD1
014156  004050                          .WORD   ALINFO
014160                          3$:     ENDSEG
014160                          10000$:
014160  104405                          TRAP    C$ESEG

014162  005722                          TST     (R2)+                   ;UPDATE THE OFFSET POINTER TO DATA TABLE
014164  005201                          INC     R1                      ;UPDATE MSAD BITS
014166  032701  000020                  BIT     #BIT4,R1                ;CHECK IF RAM COMPLETELY WRITTEN
014172  001737                          BEQ     1$                      ;IF NOT THEN DO NEXT LOCATION
```

```
                                        MOV     #MSEL0!MSEL1,R1         ;RESET BITS FOR CONTROL REGISTER 2
                                        CLR     R2                      ;CLEAR OFFSET POINTER TO DATA TABLE

                                4$:     BGNSEG
                                        TRAP    C$BSEG

                                        ;SET SIGNALS MSEL0 H AND MSEL1 H TO A ONE. SET MSAD BITS 17 AND 16
                                        ;TO THE ADDRESS TO BE TESTED.

                                        MOV     R1,R2LOAD               ;SETUP BITS TO BE LOADED
                                        JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REGISTER 2
                                        BEQ     5$                      ;IF LOADED OK THEN CONTINUE
                                        ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
                                        TRAP    C$ERDF
                                        .WORD   2
                                        .WORD   0
                                        .WORD   R2EROR
                                        CKLOOP
                                        TRAP    C$CLP1

                                        ;THE FOLLOWING SECTION WILL VALIDATE THAT THE DATA WRITTEN IN THE
                                        ;BEGINNING OF THIS TEST REMAINS UNCHANGED. IF AN ERROR OCCURS, THE
                                        ;PROBLEM IS PROBABLY INTERNAL TO THE RAM CHIP.  ON A READ COMMAND
                                        ;TO CONTROL REGISTER 6 WITH THE SIGNALS MSEL0 H AND MSEL1 H SET TO
                                        ;A 1, THE SIGNAL SMDS1 L WILL BE ASSERTED TO SELECT MODULE SELECT RAM 1.

014230  016237  014302  002302  5$:     MOV     MSR1DT(R2),R6LOAD       ;SETUP DATA PREVIOUSLY WRITTEN
014236  013737  002302  002304          MOV     R6LOAD,R6GOOD           ;SETUP EXPECTED DATA PATTERN
014244  004737  005520                  JSR     PC,READR6               ;GO READ THE RAM LOCATION
014250  001404                          BEQ     6$                      ;IF DATA OK THEN CONTINUE
014252                                  ERRDF   4,MSGMD1,ALINFO         ;MODULE SELECT RAM 1 ADDRESSING ERROR
014252  104455                          TRAP    C$ERDF
014254  000004                          .WORD   4
014256  002553                          .WORD   MSGMD1
014260  004050                          .WORD   ALINFO
014262                          6$:     ENDSEG
014262                          10001$:
014262  104405                          TRAP    C$ESEG

014264  005722                          TST     (R2)+                   ;INCREMENT OFFSET POINTER TO DATA TABLE
014266  005201                          INC     R1                      ;UPDATE MSAD BITS 17 AND 16
014270  032701  000020                  BIT     #BIT4,R1                ;CHECK IF DONE
014274  001742                          BEQ     4$                      ;IF NOT CHECK NEXT ADDRESS
014276                                  EXIT    TST
014276  104432                          TRAP    C$EXIT
014300  000012                          .WORD   L10054-.
014302  000001                  MSR1DT: .WORD   1
014304  000002                          .WORD   2
014306  000004                          .WORD   4
014310  000010                          .WORD   10
014312                                  ENDTST
014312                          L10054:
014312  104401                          TRAP    C$ETST
```

```
4705                                    .SBTTL TEST 22: CHECK MEM SIM RAM'S WITH 1'S + 0'S, AND 0'S + 1'S.
4706
4707                                    ;++
4708                                    ; TEST TO CHECK THAT ALL MEMORY SIMULATOR RAM'S CAN BE WRITTEN AND READ WITH
4709                                    ; DATA PATTERN'S 125252 AND 052525. THIS TEST CHECKS THAT ALL BITS CAN BE
4710                                    ; WRITTEN TO A ONE AND ZERO AND THAT NO ADJACENT BITS ARE SHORTED TO EACH OTHER.
4711                                    ; THIS TEST IS EXECUTED IN 16 BIT MODE.  THIS TEST, HOWEVER, DOES NOT CHECK FOR
4712                                    ; INTERNAL ADDRESS SHORTS, ADDRESS BIT DROP OUT, OR THAT THE CORRECT RAM IS
4713                                    ; SELECTED. THE TEST THAT FOLLOW THIS TEST WILL PERFORM THE ABOVE CHECKS.
4714                                    ;
4715                                    ; NOTE:
4716                                    ;       WHEN AN ERROR OCCURS, THE INFORMATION PROVIDED IN CONTROL REGISTER 4
4717                                    ;       INDICATES THE ADDRESS BEING TESTED AND THE RAM BEING SELECTED.  THE
4718                                    ;       TABLE BELOW INDICATES THE 4K RAM SELECTED FOR THE ADDRESSES IN ERROR.
4719                                    ;           1ST 4K OF RAM - ADDRESSES 000000 - 017777
4720                                    ;           2ND 4K OF RAM - ADDRESSES 020000 - 037777
4721                                    ;           3RD 4K OF RAM - ADDRESSES 040000 - 057777
4722                                    ;           4TH 4K OF RAM - ADDRESSES 060000 - 077777
4723                                    ;
4724                                    ;--
4725
4726  014314                                    BGNTST
4727  014314                            T22::
4728  014314  004737  004546                   JSR     PC,INITMS               ;SELECT AND INIT MEMORY SIMULATOR
4729  014320  005037  002276                   CLR     R4LOAD                  ;SET 1ST ADDRESS TO BE TESTED TO 0
4730  014324  012701  000001                   MOV     #BIT0,R1                ;SETUP DATA FOR MODULE SELECT RAM 0
4731
4732  014330                            1$:     BGNSEG
4733  014330  104404                           TRAP    C$BSEG
4734
4735                                    ;SET SIGNAL MSEL0 H TO A ONE AND CLEAR SIGNALS MSEL1 H, MSAD17 H AND
4736                                    ;MSAD16 H IN CONTROL REGISTER 2. MSEL0 H ON A ONE AND MSEL1 H ON
4737                                    ;A ZERO WILL CAUSE THE SIGNAL SMDS0 L TO BE ASSERTED ON A WRITE OR
4738                                    ;READ COMMAND TO CONTROL REGISTER 6. SMDS0 L WILL SELECT MODULE SELECT
4739                                    ;RAM 0.
4740
4741  014332  012737  000004  002264            MOV     #MSEL0,R2LOAD           ;SETUP BITS TO BE LOADED
4742  014340  012737  177740  002270            MOV     #177740,R2MASK          ;SETUP REGISTER 2 MASK WORD
4743  014346  004737  005412                    JSR     PC,LDRDR2               ;GO LOAD,READ AND CHECK REG 2
4744  014352  001405                            BEQ     2$                      ;IF LOADED OK THEN CONT
4745  014354                                    ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
4746  014354  104455                           TRAP    C$ERDF
4747  014356  000002                           .WORD   2
4748  014360  000000                           .WORD   0
4749  014362  003710                           .WORD   R2EROR
4750  014364                                    CKLOOP
4751  014364  104406                           TRAP    C$CLP1
4752
4753                                    ;LOAD THE ADDRESS TO BE TESTED INTO CONTROL REGISTER 4. THE ADDRESS
4754                                    ;WILL APPEAR ON MSAD BITS 15-0 WHICH IS THE OUTPUT OF CONTROL REGISTER 4
4755
4756  014366  004737  005460          2$:     JSR     PC,LDRDR4               ;GO LOAD,READ AND CHECK REGISTER 4
4757  014372  001405                            BEQ     3$                      ;IF LOADED OK THEN CONTINUE
4758  014374                                    ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
4759  014374  104455                           TRAP    C$ERDF
4760  014376  000003                           .WORD   3
```

```
4761  014400  000000                          .WORD   0
4762  014402  003776                          .WORD   R4EROR
4763  014404                                  CKLOOP
4764  014404  104406                          TRAP    C$CLP1
4765
4766                                          ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 0 AND CHECK THAT IS WAS
4767                                          ;WRITTEN CORRECTLY. THE DATA PATTERNS THAT WILL BE WRITTEN ARE 1, 2, 4,
4768                                          ;10.  THESE PATTERNS WILL ENABLE THE SIGNALS EN0 H, EN1 H, EN2 H, AND
4769                                          ;EN3 H RESPECTIVELY. THESE SIGNALS ARE USED TO SELECT MEMORY SIMULATOR
4770                                          ;MEMORY WHEN THE SIGNAL SSM L IS ASSERTED LATER ON IN THE TEST.
4771
4772  014406  010137  002302          3$:     MOV     R1,R6LOAD               ;GET THE PATTERN TO BE LOADED
4773  014412  012737  177760  002306          MOV     #177760,R6MASK          ;SETUP REGISTER 6 MASK WORD
4774  014420  004737  005504                  JSR     PC,LDRDR6               ;GO LOAD, READ AND CHECK RAM 0
4775  014424  001405                          BEQ     4$                      ;IF OK THEN CONTINUE
4776  014426                                  ERRDF   4,MSGMD0,ALINFO         ;DATA ERROR IN MODULE SELECT RAM 0
4777  014426  104455                          TRAP    C$ERDF
4778  014430  000004                          .WORD   4
4779  014432  002511                          .WORD   MSGMD0
4780  014434  004050                          .WORD   ALINFO
4781  014436                                  CKLOOP
4782  014436  104406                          TRAP    C$CLP1
4783
4784                                          ;SET SIGNALS MSEL0 H AND MSEL1 H TO A ONE AND SIGNALS MSAD17 H AND MSAD16 H
4785                                          ;TO A ZERO IN CONTROL REGISTER 2. MSEL0 H AND MSEL1 H ON A ONE, WILL
4786                                          ;CAUSE THE SIGNAL SMDS1 L TO BE ASSERTED ON A WRITE OR READ TO CONTROL
4787                                          ;REGISTER 6. SMDS1 L WILL SELECT MODULE SELECT RAM 1. MSAD BITS 17 AND 16
4788                                          ;ON A ZERO WILL SELECT ADDRESS 0 OF MODULE SELECT RAM 1.
4789
4790  014440  012737  000014  002264  4$:     MOV     #MSEL0!MSEL1,R2LOAD     ;SETUP BITS TO BE LOADED
4791  014446  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REGISTER 2
4792  014452  001405                          BEQ     5$                      ;IF LOADED OK THEN CONTINUE
4793  014454                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
4794  014454  104455                          TRAP    C$ERDF
4795  014456  000002                          .WORD   2
4796  014460  000000                          .WORD   0
4797  014462  003710                          .WORD   R2EROR
4798  014464                                  CKLOOP
4799  014464  104406                          TRAP    C$CLP1
4800
4801                                          ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 1 AND CHECK THAT THE
4802                                          ;PATTERN WAS LOADED CORRECTLY. A DATA PATTERN OF ALL ONES WILL BE
4803                                          ;WRITTEN, HOWEVER, WHEN THE LOCATION IS READ BACK, A DATA PATTERN
4804                                          ;EQUAL TO ONE WILL BE READ. SETTING THE LEAST SIGNIFICANT BIT IN
4805                                          ;THE DATA PATTERN WILL FORCE THE MORE SIGNIFICANT BITS TO ZERO.
4806
4807  014466  012737  000017  002302  5$:     MOV     #BIT3!BIT2.BIT1.BIT0,R6LOAD ;GET DATA PATTERN TO BE LOADED
4808  014474  012737  000001  002304          MOV     #BIT0,R6GOOD            ;SETUP EXPECTED DATA PATTERN
4809  014502  004737  005512                  JSR     PC,LDRD6R               ;GO LOAD,READ AND CHECK RAM 1
4810  014506  001404                          BEQ     6$                      ;IF DATA OK THEN CONTINUE
4811  014510                                  ERRDF   4,MSGMD1,ALINFO         ;DATA ERROR IN MUDULE SELECT RAM 1
4812  014510  104455                          TRAP    C$ERDF
4813  014512  000004                          .WORD   4
4814  014514  002553                          .WORD   MSGMD1
4815  014516  004050                          .WORD   ALINFO
4816  014520                          6$:     ENDSEG
```

```
4817  014520                          10000$:
4818  014520  104405                          TRAP    C$ESEG
4819
4820  014522  012702  010000                  MOV     #4096.,R2               ;SETUP 4K WORD COUNTER TO DO 1 RAM
4821
4822  014526                          7$:     BGNSEG
4823  014526  104404                          TRAP    C$BSEG
4824
4825                                          ;CLEAR BITS MSEL0 H, MSEL1 H, MSAD17 H AND MSAD16 H IN CONTROL REGISTER
4826                                          ;2. MSEL0 H AND MSEL1 H ON A ZERO WILL CAUSE THE SIGNAL SSM L TO BE
4827                                          ;ASSERTED ON A WRITE OR READ TO CONTROL REGISTER 6. ON A WRITE OR READ
4828                                          ;TO CONTROL REGISTER 6, DATA WILL BE WRITTEN OR READ FROM THE MEMORY
4829                                          ;SIMULATOR RAM SELECTED. THE SIMULATOR MEMORY IS SELECTED VIA THE
4830                                          ;MODULE SELECT RAM'S (0 AND 1).
4831
4832  014530  005037  002264                  CLR     R2LOAD                  ;SETUP TO CLEAR REGISTER 2
4833  014534  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REG 2
4834  014540  001405                          BEQ     8$                      ;IF LOADED OK THEN CONTINUE
4835  014542                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
4836  014542  104455                          TRAP    C$ERDF
4837  014544  000002                          .WORD   2
4838  014546  000000                          .WORD   0
4839  014550  003710                          .WORD   R2EROR
4840  014552                                  CKLOOP
4841  014552  104406                          TRAP    C$CLP1
4842
4843                                          ;LOAD MEMORY SIMULATOR ADDRESS TO BE TESTED INTO CONTROL REGISTER 4
4844
4845  014554  004737  005460          8$:     JSR     PC,LDRDR4               ;GO LOAD,READ AND CHECK REGISTER 4
4846  014560  001405                          BEQ     9$                      ;IF LOADED OK THEN CONTINUE
4847  014562                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
4848  014562  104455                          TRAP    C$ERDF
4849  014564  000003                          .WORD   3
4850  014566  000000                          .WORD   0
4851  014570  003776                          .WORD   R4EROR
4852  014572                                  CKLOOP
4853  014572  104406                          TRAP    C$CLP1
4854
4855                                          ;LOAD DATA PETTERN 125252 INTO MEMORY SIMULATOR RAM. ON A WRITE OR
4856                                          ;READ TO CONTROL REGISTER 6, THE SIGNAL SSM L WILL BE ASSERTED. THIS
4857                                          ;SIGNAL WILL ENABLE BOTH MODULE SELECT RAMS AND ENABLE THE MEMORY
4858                                          ;SIMULATOR RAM TO BE WRITTEN OR READ.
4859
4860  014574  012737  125252  002302  9$:     MOV     #125252,R6LOAD          ;SETUP DATA PATTERN TO BE LOADED
4861  014602  005037  002306                  CLR     R6MASK                  ;SETUP TO CHECK ALL 16 BITS OF RAM
4862  014606  004737  005504                  JSR     PC,LDRDR6               ;GO LOAD, READ AND CHECK RAM LOCATION
4863  014612  001405                          BEQ     10$                     ;IF DATA OK THEN GO COMPLEMENT IT
4864  014614                                  ERRDF   4,MSGMSR,ALINFO         ;DATA ERROR IN MEMORY SIMULATOR RAM
4865  014614  104455                          TRAP    C$ERDF
4866  014616  000004                          .WORD   4
4867  014620  002730                          .WORD   MSGMSR
4868  014622  004050                          .WORD   ALINFO
4869  014624                                  CKLOOP
4870  014624  104406                          TRAP    C$CLP1
4871
4872                                          ;LOAD DATA PATTERN 052525 INTO SAME MEMORY SIMULATOR RAM LOCATION
```

```
                                        ;AND CHECK THAT IT WAS LOADED CORRECTLY
                                        
014626  012737  052525  002302  10$:    MOV     #052525,R6LOAD          ;SETUP DATA PATTERN TO BE LOADED
014634  004737  005504                   JSR     PC,LDRDR6               ;GO LOAD, READ AND CHECK RAM LOCATION
014640  001404                          BEQ     11$                     ;IF DATA OK THEN CONTINUE
014642                                  ERRDF   4,MSGMSR,ALINFO         ;DATA ERROR IN MEMORY SIMULATOR RAM
014642  104455                          TRAP    C$ERDF
014644  000004                          .WORD   4
014646  002730                          .WORD   MSGMSR
014650  004050                          .WORD   ALINFO
014652                          11$:    ENDSEG
014652                          10001$:
014652  104405                          TRAP    C$ESEG
                                        
                                        ;UPDATE MEMORY SIMULATOR ADDRESS TO NEXT ADDRESS TO BE LOADED
                                        
014654  062737  000002  002276          ADD     #2,R4LOAD               ;UPDATE ADDRESS BY 2
014662  005302                          DEC     R2                      ;DECREMENT 2K WORD COUNTER
014664  001320                          BNE     7$                      ;IF NOT DONE DO NEXT ADDRESS
                                        
                                        ;UPDATE MODULE SELECT RAM ENABLE BITS TO BE LOADED INTO MODULE
                                        ;SELECT RAM 0 ON NEXT PASS OF THIS TEST
                                        
014666  006301                          ASL     R1                      ;MOVE MODULE SELECT RAM 0 DATA PATTERN
                                                                        ;LEFT TO ENABLE NEXT MEMORY SIMULATOR RAM
014670  032701  000020                  BIT     #BIT4,R1                ;CHECK IF ALL RAMS HAVE BEEN TESTED
014674  001002                          BNE     12$                     ;IF YES THEN EXIT THE TEST
014676  000137  014330                  JMP     1$                      ;OTHERWISE DO NEXT RAM
014702                          12$:    ENDTST
014702                          L10055:
014702  104401                          TRAP    C$ETST
                                        
```

```
                                .SBTTL TEST 23: CHECK MEM SIM RAM'S CHIP SELECT LOGIC

                                ;++
                                ; THIS TES  WILL CHECK THAT EACH MEMORY SIMULATOR RAM IS SELECTED BY WRITING
                                ; A DIFFFRENT PATTERN INTO LOCATION 0 OF EACH RAM. THE TEST WILL THEN READ
                                ; LOCATION 0 OF EACH RAM CHECKING THE DATA PATTERN TO BE THAT WHICH WAS
                                ; WRITTEN PREVIOUSLY. IF A DATA ERROR OCCURS DURING THE RE-READING OF THE
                                ; RAM'S, THE RAM SELECT LOGIC IS PROBABLY AT FAULT. THE DATA PATTERNS WRITTEN
                                ; INTO MEMORY SIMULATOR RAM'S 0, 1, 2, AND 3 ARE 11111, 22222, 33333, AND
                                ; 44444 RESPECTIVELY.
                                ;
                                ; NOTE:
                                ;       WHEN AN ERROR OCCURS, THE INFORMATION PROVIDED IN CONTROL REGISTER 4
                                ;       INDICATES THE ADDRESS BEING TESTED AND THE RAM BEING SELECTED.  THE
                                ;       TABLE BELOW INDICATES THE 4K RAM SELECTED FOR THE ADDRESSES IN ERROR.
                                ;           1ST 4K OF RAM - ADDRESSES 000000 - 017777
                                ;           2ND 4K OF RAM - ADDRESSES 020000 - 037777
                                ;           3RD 4K OF RAM - ADDRESSES 040000 - 057777
                                ;           4TH 4K OF RAM - ADDRESSES 060000 - 077777
                                ;
                                ;--

                                        BGNTST
                                T23::
014704  004737  004546                  JSR     PC,INITMS               ;SELECT AND INIT MEMORY SIMULATOR

014710  005037  002276                  CLR     R4LOAD                  ;SET 1ST ADDRESS TO BE TESTED TO 0
014714  012701  000001                  MOV     #BIT0,R1                ;SETUP DATA FOR MODULE SELECT RAM 0
014720  012702  011111                  MOV     #11111,R2               ;SETUP DATA PATTERN FOR MEM SIM RAM 0

014724                          1$:     BGNSEG
014724  104404                          TRAP    C$BSEG

                                ;SET SIGNAL MSEL0 H TO A ONE AND CLEAR SIGNALS MSEL1 H, MSAD17 H AND
                                ;MSAD16 H IN CONTROL REGISTER 2. MSEL0 H ON A ONE AND MSEL1 H ON
                                ;A ZERO WILL CAUSE THE SIGNAL SMDS0 L TO BE ASSERTED ON A WRITE OR
                                ;READ COMMAND TO CONTROL REGISTER 6. SMDS0 L WILL SELECT MODULE SELECT
                                ;RAM 0.

014726  012737  000004  002264          MOV     #MSEL0,R2LOAD           ;SETUP BITS TO BE LOADED
014734  012737  177740  002270          MOV     #177740,R2MASK          ;SETUP REGISTER 2 MASK WORD
014742  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD,READ AND CHECK REG 2
014746  001405                          BEQ     2$                      ;IF LOADED OK THEN CONT
014750                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
014750  104455                          TRAP    C$ERDF
014752  000002                          .WORD   2
014754  000000                          .WORD   0
014756  003710                          .WORD   R2EROR
014760                                  CKLOOP
014760  104406                          TRAP    C$CLP1

                                ;LOAD THE ADDRESS TO BE TESTED INTO CONTROL REGISTER 4. THE ADDRESS
                                ;WILL APPEAR ON MSAD BITS 15-0 WHICH IS THE OUTPUT OF CONTROL REGISTER 4
                                ;THE ADDRESSES TO BE TESTED ARE 000000, 020000, 040000, AND 060000.

014762  004737  005460          2$:     JSR     PC,LDRDR4               ;GO LOAD,READ AND CHECK REGISTER 4
```

```
4961  014766  001405                          BEQ     3$                      ;IF LOADED OK THEN CONTINUE
4962  014770                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
4963  014770  104455                          TRAP    C$ERDF
4964  014772  000003                          .WORD   3
4965  014774  000000                          .WORD   0
4966  014776  003776                          .WORD   R4EROR
4967  015000                                  CKLOOP
4968  015000  104406                          TRAP    C$CLP1
4969
4970                                          ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 0 AND CHECK THAT IT WAS
4971                                          ;WRITTEN CORRECTLY. THE DATA PATTERNS THAT WILL BE WRITTEN ARE 1, 2, 4,
4972                                          ;10.  THESE PATTERNS WILL ENABLE THE SIGNALS EN0 H, EN1 H, EN2 H, AND
4973                                          ;EN3 H RESPECTIVELY. THESE SIGNALS ARE USED TO SELECT MEMORY SIMULATOR
4974                                          ;MEMORY WHEN THE SIGNAL SSM L IS ASSERTED LATER ON IN THE TEST.
4975
4976  015002  010137  002302          3$:     MOV     R1,R6LOAD               ;GET THE PATTERN TO BE LOADED
4977  015006  012737  177760  002306          MOV     #177760,R6MASK          ;SETUP REGISTER 6 MASK WORD
4978  015014  004737  005504                  JSR     PC,LDRDR6               ;GO LOAD, READ AND CHECK RAM 0
4979  015020  001405                          BEQ     4$                      ;IF OK THEN CONTINUE
4980  015022                                  ERRDF   4,MSGMD0,ALINFO         ;DATA ERROR IN MODULE SELECT RAM 0
4981  015022  104455                          TRAP    C$ERDF
4982  015024  000004                          .WORD   4
4983  015026  002511                          .WORD   MSGMD0
4984  015030  004050                          .WORD   ALINFO
4985  015032                                  CKLOOP
4986  015032  104406                          TRAP    C$CLP1
4987
4988                                          ;SET SIGNALS MSEL0 H AND MSEL1 H TO A ONE AND SIGNALS MSAD17 H AND MSAD16 H
4989                                          ;TO A ZERO IN CONTROL REGISTER 2. MSEL0 H AND MSEL1 H ON A ONE, WILL
4990                                          ;CAUSE THE SIGNAL SMDS1 L TO BE ASSERTED ON A WRITE OR READ TO CONTROL
4991                                          ;REGISTER 6. SMDS1 L WILL SELECT MODULE SELECT RAM 1. MSAD BITS 17 AND 16
4992                                          ;ON A ZERO WILL SELECT ADDRESS 0 OF MODULE SELECT RAM 1.
4993
4994  015034  012737  000014  002264  4$:     MOV     #MSEL0!MSEL1,R2LOAD     ;SETUP BITS TO BE LOADED
4995  015042  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REGISTER 2
4996  015046  001405                          BEQ     5$                      ;IF LOADED OK THEN CONTINUE
4997  015050                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
4998  015050  104455                          TRAP    C$ERDF
4999  015052  000002                          .WORD   2
5000  015054  000000                          .WORD   0
5001  015056  003710                          .WORD   R2EROR
5002  015060                                  CKLOOP
5003  015060  104406                          TRAP    C$CLP1
5004
5005                                          ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 1 AND CHECK THAT THE
5006                                          ;PATTERN WAS LOADED CORRECTLY. A DATA PATTERN OF ALL ONES WILL BE
5007                                          ;WRITTEN, HOWEVER, WHEN THE LOCATION IS READ BACK, A DATA PATTERN
5008                                          ;EQUAL TO ONE WILL BE READ. SETTING THE LEAST SIGNIFICANT BIT IN
5009                                          ;THE DATA PATTERN WILL FORCE THE MORE SIGNIFICANT BITS TO ZERO.
5010
5011  015062  012737  000017  002302  5$:     MOV     #BIT3!BIT2!BIT1.BIT0,R6LOAD ;GET DATA PATTERN TO BE LOADED
5012  015070  012737  000001  002304          MOV     #BIT0,R6GOOD            ;SETUP EXPECTED DATA PATTERN
5013  015076  004737  005512                  JSR     PC,LDRD6R               ;GO LOAD,READ AND CHECK RAM 1
5014  015102  001405                          BEQ     6$                      ;IF DATA OK THEN CONTINUE
5015  015104                                  ERRDF   4,MSGMD1,ALINFO         ;DATA ERROR IN MODULE SELECT RAM 1
5016  015104  104455                          TRAP    C$ERDF
```

```
                                        .WORD   4
                                        .WORD   MSGMD1
                                        .WORD   ALINFO
                                        CKLOOP
                                        TRAP    C$CLP1

                                        ;CLEAR BITS MSEL0 H, MSEL1 H, MSAD17 H AND MSAD16 H IN CONTROL REGISTER
                                        ;2. MSEL0 H AND MSEL1 H ON A ZERO WILL CAUSE THE SIGNAL SSM L TO BE
                                        ;ASSERTED ON A WRITE OR READ TO CONTROL REGISTER 6. ON A WRITE OR READ
                                        ;TO CONTROL REGISTER 6, DATA WILL BE WRITTEN OR READ FROM THE MEMORY
                                        ;SIMULATOR RAM SELECTED. THE SIMULATOR MEMORY IS SELECTED VIA THE
                                        ;MODULE SELECT RAM'S (0 AND 1).

015116  005037  002264        6$:     CLR     R2LOAD                  ;SETUP TO CLEAR REGISTER 2
015122  004737  005412                JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REG 2
015126  001405                        BEQ     7$                      ;IF LOADED OK THEN CONTINUE
015130                                ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
015130  104455                        TRAP    C$ERDF
015132  000002                        .WORD   2
015134  000000                        .WORD   0
015136  003710                        .WORD   R2EROR
015140                                CKLOOP
015140  104406                        TRAP    C$CLP1

                                        ;LOAD DATA PATTERN INTO MEMORY SIMULATOR RAM AND CHECK THAT THE CORRECT
                                        ;PATTERN WAS LOADED. THE SIGNAL SSM L WILL BE ASSERTED. THIS SIGNAL
                                        ;WILL ENABLE BOTH MODULE SELECT RAMS AND ENABLE THE MEMORY SIMULATOR RAM
                                        ;TO BE WRITTEN OR READ.

015142  010237  002302        7$:     MOV     R2,R6LOAD               ;SETUP DATA PATTERN TO BE LOADED
015146  005037  002306                CLR     R6MASK                  ;SET MASK WORD TO 0
015152  004737  005504                JSR     PC,LDRDR6               ;GO LOAD, READ AND CHECK RAM LOCATION
015156  001404                        BEQ     8$                      ;IF DATA OK THEN CONTINUE
015160                                ERRDF   4,MSGMSR,ALINFO         ;DATA ERROR IN MEMORY SIMULATOR RAM
015160  104455                        TRAP    C$ERDF
015162  000004                        .WORD   4
015164  002730                        .WORD   MSGMSR
015166  004050                        .WORD   ALINFO
015170                          8$:     ENDSEG
015170                          10000$:
015170  104405                        TRAP    C$ESEG

015172  006301                        ASL     R1                      ;UPDATE MODULE SELECT RAM 0'S DATA
015174  062737  020000  002276        ADD     #MSAD13,R4LOAD          ;UPDATE ADDRESS TO ADDRESS 0 OF NEXT RAM
015202  062702  011111                ADD     #11111,R2               ;UPDATE DATA PATTERN FOR NEXT RAM
015206  032701  000020                BIT     #BIT4,R1                ;CHECK IF ALL 4 RAM'S WRITTEN
015212  001002                        BNE     9$                      ;IF YES THEN GO READ THEM AGAIN
015214  000137  014724                JMP     1$                      ;GO WRITE ADDRESS 0 OF NEXT RAM

015220  005037  002276        9$:     CLR     R4LOAD                  ;RESET ADDRESS TO ADDRESS 0
015224  012702  011111                MOV     #11111,R2               ;RESET DATA PATTERN

                                        ;THE FOLLOWING SECTION OF CODE WILL READ ADDRESS 0 OF EACH RAM AND
                                        ;CHECK THAT THE DATA PATTERNS WRITTEN PREVIOUSLY REMAIN UNCHANGED.

015230                          10$:    BGNSEG
```

```
5073  015230  104404                          TRAP    C$BSEG
5074
5075                                          ;CLEAR BITS MSEL0 H, MSEL1 H, MSAD17 H AND MSAD 16 H IN CONTROL REGISTER
5076                                          ;2. MSEL0 H AND MSEL1 H ON A ZERO WILL CAUSE THE SIGNAL SSM L TO BE
5077                                          ;ASSERTED ON A READ COMMAND TO CONTROL REGISTER 6. THE SIGNAL SSM L
5078                                          ;WILL ENABLE THE SIMULATOR RAM MEMORY'S TO BE READ.
5079
5080  015232  005037  002264                  CLR     R2LOAD                  ;CLEAR ALL BITS IN CONTROL REGISTER 2
5081  015236  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REGISTER 2
5082  015242  001405                          BEQ     11$                     ;IF LOADED OK THEN CONTINUE
5083  015244                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
5084  015244  104455                          TRAP    C$ERDF
5085  015246  000002                          .WORD   2
5086  015250  000000                          .WORD   0
5087  015252  003710                          .WORD   R2EROR
5088  015254                                  CKLOOP
5089  015254  104406                          TRAP    C$CLP1
5090
5091                                          ;LOAD ADDRESS 0 OF THE RAM TO BE TESTED INTO CONTROL REGISTER 4.
5092
5093  015256  004737  005460          11$:    JSR     PC,LDRDR4               ;GO LOAD, READ AND CHECK REGISTER 4
5094  015262  001405                          BEQ     12$                     ;IF LOADED OK THEN CONTINUE
5095  015264                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
5096  015264  104455                          TRAP    C$ERDF
5097  015266  000003                          .WORD   3
5098  015270  000000                          .WORD   0
5099  015272  003776                          .WORD   R4EROR
5100  015274                                  CKLOOP
5101  015274  104406                          TRAP    C$CLP1
5102
5103                                          ;READ MEMORY SIMULATOR RAM AND CHECK THAT THE DATA PREVIOUSLY WRITTEN
5104                                          ;DID NOT CHANGE.
5105
5106  015276  010237  002302          12$:    MOV     R2,R6LOAD               ;SETUP DATA PREVIOUSLY WRITTEN
5107  015302  013737  002302  002304          MOV     R6LOAD,R6GOOD           ;SETUP EXPECTED DATA
5108  015310  005037  002306                  CLR     R6MASK                  ;CLEAR THE MASK WORD
5109  015314  004737  005520                  JSR     PC,READR6               ;READ AND CHECK RAM LOCATION
5110  015320  001404                          BEQ     13$                     ;IF DATA OK THEN CONTINUE
5111  015322                                  ERRDF   4,MSGMSC,ALINFO         ;CHIP ENABLE ERROR - MEM SIM RAM
5112  015322  104455                          TRAP    C$ERDF
5113  015324  000004                          .WORD   4
5114  015326  002773                          .WORD   MSGMSC
5115  015330  004050                          .WORD   ALINFO
5116  015332                          13$:    ENDSEG
5117  015332                          10001$:
5118  015332  104405                          TRAP    C$ESEG
5119
5120  015334  062702  011111                  ADD     #11111,R2               ;UPDATE DATA PATTERN
5121  015340  062737  020000  002276          ADD     #MSAD13,R4LOAD          ;UPDATE ADDRESS TO 0 OF NEXT RAM
5122  015346  100330                          BPL     10$                     ;GO CHECK NEXT ADDRESS IF NOT DONE
5123  015350                                  ENDTST
5124  015350                          L10056:
5125  015350  104401                          TRAP    C$ETST
5126
```

```
                                .SBTTL TEST 24: CHECK EACH ADDRESS OF MEM SIM RAM TO BE ADDRESSED

                                ;++
                                ; TEST TO CHECK THAT EACH ADDRESS OF THE MEMORY SIMULATOR CAN BE ADDRESSED
                                ; CORRECTLY AND THAT NO ADDRESSES ARE SHORTED TOGETHER. TO DO THIS, THE
                                ; PROGRAM WRITES EACH LOCATION OF THE MEMORY SIMULATOR RAM WITH DATA EQUAL
                                ; TO THE ADDRESS OF THE LOCATION. AS EACH LOCATION IS WRITTEN THE PROGRAM
                                ; READS THE LOCATION AND CHECKS THAT THE LOCATION IS WRITTEN CORRECTLY. WHEN
                                ; ALL MEMORY SIMULATOR RAM LOCATIONS HAVE BEEN WRITTEN, THE PROGRAM WILL RE-
                                ; READ THE RAMS CHECKING THAT ALL LOCATIONS CONTAIN AS DATA THEIR ADDRESS.
                                ; THE PROGRAM WILL THEN RESET THE POINTER TO THE BEGINNING ADDRESS OF THE
                                ; RAM'S AND DO THE FOLLOWING:
                                ;    1. READ THE LOCATION AND CHECK THAT IT CONTAINS ITS ADDRESS
                                ;    2. WRITE THE LOCATION WITH THE ONE'S COMPLEMENT OF THE ADDRESS AND CHECK
                                ;       THAT THE ONES COMPLEMENT CAN BE READ BACK
                                ;    3. REPEAT STEPS ONE AND TWO FOR EACH ADDRESS OF THE RAM.
                                ; THE TEST WILL THEN RESET THE ADDRESS POINTER TO THE BEGINNING ADDRESS OF THE
                                ; RAM AND CHECK EACH LOCATION OF THE RAM TO CONTAIN THE ONES COMPLEMENT OF ITS
                                ; ADDRESS.
                                ;
                                ; NOTE:
                                ;       WHEN AN ERROR OCCURS, THE INFORMATION PROVIDED IN CONTROL REGISTER 4
                                ;       INDICATES THE ADDRESS BEING TESTED AND THE RAM BEING SELECTED.  THE
                                ;       TABLE BELOW INDICATES THE 4K RAM SELECTED FOR THE ADDRESSES IN ERROR.
                                ;           1ST 4K OF RAM - ADDRESSES 000000 - 017777
                                ;           2ND 4K OF RAM - ADDRESSES 020000 - 037777
                                ;           3RD 4K OF RAM - ADDRESSES 040000 - 057777
                                ;           4TH 4K OF RAM - ADDRESSES 060000 - 077777
                                ;
                                ;--

015352                                  BGNTST
015352                          T24::
015352  004737  004546                  JSR     PC,INITMS               ;SELECT AND INIT MEMORY SIMULATOR
015356  005037  002276                  CLR     R4LOAD                  ;SET 1ST ADDRESS TO BE TESTED TO 0
015362  012701  000001                  MOV     #BIT0,R1                ;SETUP DATA FOR MODULE SELECT RAM 0

015366                          1$:     BGNSEG
015366  104404                          TRAP    C$BSEG

                                        ;SET SIGNAL MSEL0 H TO A ONE AND CLEAR SIGNALS MSEL1 H, MSAD17 H AND
                                        ;MSAD16 H IN CONTROL REGISTER 2. MSEL0 H ON A ONE AND MSEL1 H ON
                                        ;A ZERO WILL CAUSE THE SIGNAL SMDS0 L TO BE ASSERTED ON A WRITE OR
                                        ;READ COMMAND TO CONTROL REGISTER 6. SMDS0 L WILL SELECT MODULE SELECT
                                        ;RAM 0.

015370  012737  000004  002264          MOV     #MSEL0,R2LOAD           ;SETUP BITS TO BE LOADED
015376  012737  177740  002270          MOV     #177740,R2MASK          ;SETUP REGISTER 2 MASK WORD
015404  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD,READ AND CHECK REG 2
015410  001405                          BEQ     2$                      ;IF LOADED OK THEN CONT
015412                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
015412  104455                          TRAP    C$ERDF
015414  000002                          .WORD   2
015416  000000                          .WORD   0
015420  003710                          .WORD   R2EROR
015422                                  CKLOOP
```

```
5183  015422  104406                          TRAP    C$CLP1
5184
5185                                          ;LOAD THE ADDRESS TO BE TESTED INTO CONTROL REGISTER 4. THE ADDRESS
5186                                          ;WILL APPEAR ON MSAD BITS 15-0 WHICH IS THE OUTPUT OF CONTROL REGISTER 4
5187
5188  015424  004737  005460          2$:     JSR     PC,LDRDR4               ;GO LOAD,READ AND CHECK REGISTER 4
5189  015430  001405                          BEQ     3$                      ;IF LOADED OK THEN CONTINUE
5190  015432                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
5191  015432  104455                          TRAP    C$ERDF
5192  015434  000003                          .WORD   3
5193  015436  000000                          .WORD   0
5194  015440  003776                          .WORD   R4EROR
5195  015442                                  CKLOOP
5196  015442  104406                          TRAP    C$CLP1
5197
5198                                          ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 0 AND CHECK THAT IS WAS
5199                                          ;WRITTEN CORRECTLY. THE DATA PATTERNS THAT WILL BE WRITTEN ARE 1, 2, 4,
5200                                          ;10.  THESE PATTERNS WILL ENABLE THE SIGNALS EN0 H, EN1 H, EN2 H, AND
5201                                          ;EN3 H RESPECTIVELY. THESE SIGNALS ARE USED TO SELECT MEMORY SIMULATOR
5202                                          ;MEMORY WHEN THE SIGNAL SSM L IS ASSERTED LATER ON IN THE TEST.
5203
5204  015444  010137  002302          3$:     MOV     R1,R6LOAD               ;GET THE PATTERN TO BE LOADED
5205  015450  012737  177760  002306          MOV     #177760,R6MASK          ;SETUP REGISTER 6 MASK WORD
5206  015456  004737  005504                  JSR     PC,LDRDR6               ;GO LOAD, READ AND CHECK RAM 0
5207  015462  001405                          BEQ     4$                      ;IF OK THEN CONTINUE
5208  015464                                  ERRDF   4,MSGMD0,ALINFO         ;DATA ERROR IN MODULE SELECT RAM 0
5209  015464  104455                          TRAP    C$ERDF
5210  015466  000004                          .WORD   4
5211  015470  002511                          .WORD   MSGMD0
5212  015472  004050                          .WORD   ALINFO
5213  015474                                  CKLOOP
5214  015474  104406                          TRAP    C$CLP1
5215
5216                                          ;SET SIGNALS MSEL0 H AND MSEL1 H TO A ONE AND SIGNALS MSAD17 H AND MSAD16 H
5217                                          ;TO A ZERO IN CONTROL REGISTER 2. MSEL0 H AND MSEL1 H ON A ONE, WILL
5218                                          ;CAUSE THE SIGNAL SMDS1 L TO BE ASSERTED ON A WRITE OR READ TO CONTROL
5219                                          ;REGISTER 6. SMDS1 L WILL SELECT MODULE SELECT RAM 1. MSAD BITS 17 AND 16
5220                                          ;ON A ZERO WILL SELECT ADDRESS 0 OF MODULE SELECT RAM 1.
5221
5222  015476  012737  000014  002264  4$:     MOV     #MSEL0!MSEL1,R2LOAD     ;SETUP BITS TO BE LOADED
5223  015504  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REGISTER 2
5224  015510  001405                          BEQ     5$                      ;IF LOADED OK THEN CONTINUE
5225  015512                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
5226  015512  104455                          TRAP    C$ERDF
5227  015514  000002                          .WORD   2
5228  015516  000000                          .WORD   0
5229  015520  003710                          .WORD   R2EROR
5230  015522                                  CKLOOP
5231  015522  104406                          TRAP    C$CLP1
5232
5233                                          ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 1 AND CHECK THAT THE
5234                                          ;PATTERN WAS LOADED CORRECTLY. A DATA PATTERN OF ALL ONES WILL BE
5235                                          ;WRITTEN, HOWEVER, WHEN THE LOCATION IS READ BACK, A DATA PATTERN
5236                                          ;EQUAL TO ONE WILL BE READ. SETTING THE LEAST SIGNIFICANT BIT IN
5237                                          ;THE DATA PATTERN WILL FORCE THE MORE SIGNIFICANT BITS TO ZERO.
5238
```

```
 5239  015524  012737  000017  002302  5$:     MOV     #BIT3!BIT2!BIT1!BIT0,R6LOAD ;GET DATA PATTERN TO BE LOADED
 5240  015532  012737  000001  002304          MOV     #BIT0,R6GOOD                ;SETUP EXPECTED DATA PATTERN
 5241  015540  004737  005512                  JSR     PC,LDRD6R                   ;GO LOAD,READ AND CHECK RAM 1
 5242  015544  001404                          BEQ     6$                          ;IF DATA OK THEN CONTINUE
 5243  015546                                  ERRDF   4,MSGMD1,ALINFO             ;DATA ERROR IN MUDULE SELECT RAM 1
 5244  015546  104455                          TRAP    C$ERDF
 5245  015550  000004                          .WORD   4
 5246  015552  002553                          .WORD   MSGMD1
 5247  015554  004050                          .WORD   ALINFO
 5248  015556                          6$:     ENDSEG
 5249  015556                          10000$:
 5250  015556  104405                          TRAP    C$ESEG
 5251
 5252  015560  012702  010000                  MOV     #4096.,R2                   ;SETUP 4K WORD COUNTER TO DO 1 RAM
 5253
 5254  015564                          7$:     BGNSEG
 5255  015564  104404                          TRAP    C$BSEG
 5256
 5257                                          ;CLEAR BITS MSEL0 H, MSEL1 H, MSAD17 H AND MSAD16 H IN CONTROL REGISTER
 5258                                          ;2. MSEL0 H AND MSEL1 H ON A ZERO WILL CAUSE THE SIGNAL SSM L TO BE
 5259                                          ;ASSERTED ON A WRITE OR READ TO CONTROL REGISTER 6. ON A WRITE OR READ
 5260                                          ;TO CONTROL REGISTER 6, DATA WILL BE WRITTEN OR READ FROM THE MEMORY
 5261                                          ;SIMULATOR RAM SELECTED. THE SIMULATOR MEMORY IS SELECTED VIA THE
 5262                                          ;MODULE SELECT RAM'S (0 AND 1).
 5263
 5264  015566  005037  002264                  CLR     R2LOAD                      ;SETUP TO CLEAR REGISTER 2
 5265  015572  004737  005412                  JSR     PC,LDRDR2                   ;GO LOAD, READ AND CHECK REG 2
 5266  015576  001405                          BEQ     8$                          ;IF LOADED OK THEN CONTINUE
 5267  015600                                  ERRDF   2,,R2EROR                   ;REGISTER 2 NOT EQUAL EXPECTED
 5268  015600  104455                          TRAP    C$ERDF
 5269  015602  000002                          .WORD   2
 5270  015604  000000                          .WORD   0
 5271  015606  003710                          .WORD   R2EROR
 5272  015610                                  CKLOOP
 5273  015610  104406                          TRAP    C$CLP1
 5274
 5275                                          ;LOAD MEMORY SIMULATOR ADDRESS TO BE TESTED INTO CONTROL REGISTER 4
 5276
 5277  015612  004737  005460          8$:     JSR     PC,LDRDR4                   ;GO LOAD,READ AND CHECK REGISTER 4
 5278  015616  001405                          BEQ     9$                          ;IF LOADED OK THEN CONTINUE
 5279  015620                                  ERRDF   3,,R4EROR                   ;REGISTER 4 NOT EQUAL EXPECTED
 5280  015620  104455                          TRAP    C$ERDF
 5281  015622  000003                          .WORD   3
 5282  015624  000000                          .WORD   0
 5283  015626  003776                          .WORD   R4EROR
 5284  015630                                  CKLOOP
 5285  015630  104406                          TRAP    C$CLP1
 5286
 5287                                          ;LOAD DATA PATTERN EQUAL TO ADDRESS INTO MEMORY SIMULATOR RAM. ON A
 5288                                          ;WRITE OR READ TO CONTROL REGISTER 6, THE SIGNAL SSM L WILL BE ASSERTED.
 5289                                          ;THIS SIGNAL WILL ENABLE BOTH MODULE SELECT RAMS AND ENABLE THE MEMORY
 5290                                          ;SIMULATOR RAM TO BE WRITTEN OR READ.
 5291
 5292  015632  013737  002276  002302  9$:     MOV     R4LOAD,R6LOAD               ;SETUP DATA PATTERN TO BE LOADED
 5293  015640  005037  002306                  CLR     R6MASK                      ;SETUP TO CHECK ALL 16 BITS OF RAM
 5294  015644  004737  005504                  JSR     PC,LDRDR6                   ;GO LOAD, READ AND CHECK RAM LOCATION
```

```
5295  015650  001404                          BEQ     10$                     ;IF DATA OK THEN GO COMPLEMENT IT
5296  015652                                  ERRDF   4,MSGMSR,ALINFO         ;DATA ERROR IN MEMORY SIMULATOR RAM
5297  015652  104455                          TRAP    C$ERDF
5298  015654  000004                          .WORD   4
5299  015656  002730                          .WORD   MSGMSR
5300  015660  004050                          .WORD   ALINFO
5301  015662                          10$:    ENDSEG
5302  015662                          10001$:
5303  015662  104405                          TRAP    C$ESEG
5304
5305                                          ;UPDATE MEMORY SIMULATOR ADDRESS TO NEXT ADDRESS TO BE LOADED
5306
5307  015664  062737  000002  002276          ADD     #2,R4LOAD               ;UPDATE ADDRESS BY 2
5308  015672  005302                          DEC     R2                      ;DECREMENT 2K WORD COUNTER
5309  015674  001333                          BNE     7$                      ;IF NOT DONE DO NEXT ADDRESS
5310
5311                                          ;UPDATE MODULE SELECT RAM ENABLE BITS TO BE LOADED INTO MODULE
5312                                          ;SELECT RAM 0 ON NEXT PASS OF THIS TEST
5313
5314  015676  006301                          ASL     R1                      ;MOVE MODULE SELECT RAM 0 DATA PATTERN
5315                                                                          ;LEFT TO ENABLE NEXT MEMORY SIMULATOR RAM
5316  015700  032701  000020                  BIT     #BIT4,R1                ;CHECK IF ALL RAMS HAVE BEEN TESTED
5317  015704  001002                          BNE     11$                     ;IF YES THEN EXIT THE TEST
5318  015706  000137  015366                  JMP     1$                      ;OTHERWISE DO NEXT RAM
5319
5320                                          ;THE FOLLOWING SECTION OF CODE WILL READ THE MEMORY SIMULATOR RAMS
5321                                          ;AND CHECK THAT EACH ADDRESS OF THE RAM CONTAINS AS DATA ITS ADDRESS.
5322
5323  015712  005037  002276          11$:    CLR     R4LOAD                  ;RESET THE ADDRESS TO 0
5324
5325  015716                          12$:    BGNSEG
5326  015716  104404                          TRAP    C$BSEG
5327
5328                                          ;LOAD THE ADDRESS TO BE CHECKED INTO CONTROL REGISTER 4. THE MODULE
5329                                          ;SELECT RAMS HAVE BEEN INITIALIZED PREVIOUSLY.
5330
5331  015720  004737  005460                  JSR     PC,LDRDR4               ;GO LOAD, READ AND CHECK REGISTER 4
5332  015724  001405                          BEQ     13$                     ;IF LOADED OK THEN CONTINUE
5333  015726                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED.
5334  015726  104455                          TRAP    C$ERDF
5335  015730  000003                          .WORD   3
5336  015732  000000                          .WORD   0
5337  015734  003776                          .WORD   R4EROR
5338  015736                                  CKLOOP
5339  015736  104406                          TRAP    C$CLP1
5340
5341                                          ;READ THE MEMORY SIMULATOR RAM AND CHECK THAT THE ADDRESS CONTAINS IT'
5342                                          ;ADDRESS AS DATA.
5343
5344  015740  013737  002276  002302  13$:    MOV     R4LOAD,R6LOAD           ;SETUP DATA THAT WAS PREVIOUSLY WRITTEN
5345  015746  013737  002302  002304          MOV     R6LOAD,R6GOOD           ;SETUP EXPECTED DATA
5346  015754  005037  002306                  CLR     R6MASK                  ;SET REGISTER 6 MASK TO ZERO
5347  015760  004737  005520                  JSR     PC,READR6               ;GO READ AND CHECK THE RAM LOCATION
5348  015764  001404                          BEQ     14$                     ;IF DATA EQUALS ADDRESS THEN CONTINUE
5349  015766                                  ERRDF   4,MSGMSR,ALINFO         ;DATA ERROR IN MEMORY SIMULATOR RAM
5350  015766  104455                          TRAP    C$ERDF
```

```
 5351  015770  000004                          .WORD   4
 5352  015772  002730                          .WORD   MSGMSR
 5353  015774  004050                          .WORD   ALINFO
 5354  015776                          14$:    ENDSEG
 5355  015776                          10002$:
 5356  015776  104405                          TRAP    C$ESEG
 5357
 5358  016000  062737  000002  002276          ADD     #2,R4LOAD               ;UPDATE THE ADDRESS TO THE NEXT LOCATION
 5359  016006  100343                          BPL     12$                     ;IF NOT DONE CHECK NEXT ADDRESS
 5360
 5361                                          ;THE FOLLOWING SECTION OF CODE WILL READ THE CONTENTS OF A LOCATION
 5362                                          ;CHECKING THAT THE CONTENTS EQUALS THE ADDRESS. THE TEST WILL THEN
 5363                                          ;WRITE THE ONES COMPLEMENT OF THE ADDRESS INTO THE LOCATION AND CHECK
 5364                                          ;THAT THE LOCATION WAS WRITTEN CORRECTLY.
 5365
 5366  016010  005037  002276                  CLR     R4LOAD                  ;RESET THE ADDRESS TO ADDRESS 0
 5367
 5368  016014                          15$:    BGNSEG
 5369  016014  104404                          TRAP    C$BSEG
 5370
 5371                                          ;LOAD THE ADDRESS TO BE TESTED INTO CONTROL REGISTER 4
 5372
 5373  016016  004737  005460                  JSR     PC,LDRDR4               ;GO LOAD, READ, AND CHECK REGISTER 4
 5374  016022  001405                          BEQ     16$                     ;IF LOADED OK THEN CONTINUE
 5375  016024                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
 5376  016024  104455                          TRAP    C$ERDF
 5377  016026  000003                          .WORD   3
 5378  016030  000000                          .WORD   0
 5379  016032  003776                          .WORD   R4EROR
 5380  016034                                  CKLOOP
 5381  016034  104406                          TRAP    C$CLP1
 5382
 5383                                          ;READ THE LOCATION CHECKING THAT THE DATA STORED IN LOCATION EQUALS
 5384                                          ;THE ADDRESS OF THE LOCATION
 5385
 5386  016036  013737  002276  002302  16$:    MOV     R4LOAD,R6LOAD           ;GET THE ADDRESS
 5387  016044  013737  002302  002304          MOV     R6LOAD,R6GOOD           ;SETUP FOR EXPECTED DATA
 5388  016052  004737  005520                  JSR     PC,READR6               ;GO READ AND CHECK THE DATA
 5389  016056  001405                          BEQ     17$                     ;IF ADDRESS EQUALS DATA THEN CONT
 5390  016060                                  ERRDF   4,MSGMSR,ALINFO         ;ADDRESS SHORT OR DATA NEQ EXPECTED
 5391  016060  104455                          TRAP    C$ERDF
 5392  016062  000004                          .WORD   4
 5393  016064  002730                          .WORD   MSGMSR
 5394  016066  004050                          .WORD   ALINFO
 5395  016070                                  CKLOOP
 5396  016070  104406                          TRAP    C$CLP1
 5397
 5398                                          ;WRITE THE COMPLEMENT OF THE ADDRESS IN THE LOCATION ADDRESSED.
 5399
 5400  016072  013737  002276  002302  17$:    MOV     R4LOAD,R6LOAD           ;GET THE ADDRESS
 5401  016100  005137  002302                  COM     R6LOAD                  ;COMPLEMENT IT
 5402  016104  004737  005504                  JSR     PC,LDRDR6               ;GO LOAD, READ AND CHECK RAM
 5403  016110  001404                          BEQ     18$                     ;IF DATA OK THEN CONTINUE
 5404  016112                                  ERRDF   4,MSGMSR,ALINFO         ;DATA ERROR IN MEMORY SIMULATOR RAM
 5405  016112  104455                          TRAP    C$ERDF
 5406  016114  000004                          .WORD   4
```

```
                                        .WORD   MSGMSR
                                        .WORD   ALINFO
                                18$:    ENDSEG
                                10003$:
                                        TRAP    C$ESEG

                                        ADD     #2,R4LOAD               ;UPDATE THE TEST ADDRESS BY 2
                                        BPL     15$                     ;IF NOT DONE GO DO NEXT ADDRESS

                                        ;THE FOLLOWING SECTION OF CODE WILL READ THE MEMORY SIMULATOR RAM'S
                                        ;CHECKING THAT EACH ADDRESS CONTAINS ITS ONES COMPLEMENT AS DATA

                                        CLR     R4LOAD                  ;RESET THE ADDDRESS POINTER TO 0

                                19$:    BGNSEG
                                        TRAP    C$BSEG

                                        ;LOAD ADDRESS TO BE TESTED INTO CONTROL REGISTER 4

                                        JSR     PC,LDRDR4               ;GO LOAD, READ AND CHECK REGISTER 4
                                        BEQ     20$                     ;IF LOADED OK THEN CONTINUE
                                        ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
                                        TRAP    C$ERDF
                                        .WORD   3
                                        .WORD   0
                                        .WORD   R4EROR
                                        CKLOOP
                                        TRAP    C$CLP1

                                        ;READ THE MEMORY SIMULATER RAM AND CHECK THAT THE ADDRESS CONTAINS
                                        ;ITS COMPLEMENT AS DATA

                                20$:    MOV     R4LOAD,R6LOAD           ;GET THE ADDRESS
                                        COM     R6LOAD                  ;SET IT TO ITS ONES COMPLEMENT
                                        MOV     R6LOAD,R6GOOD           ;SETUP EXPECTED DATA
                                        JSR     PC,READR6               ;GO READ AND CHEK THE RAM LOCATION
                                        BEQ     21$                     :IF DATA OK THEN CONTINUE
                                        ERRDF   4,MSGMSR,ALINFO         ;LOCATION NOT EQUAL TO ITS ONES COMP
                                        TRAP    C$ERDF
                                        .WORD   4
                                        .WORD   MSGMSR
                                        .WORD   ALINFO
                                                                        ;OR AN ADDRESS SHORT IN RAM
                                21$:    ENDSEG
                                10004$:
                                        TRAP    C$ESEG

                                        ADD     #2,R4LOAD               ;UPDATE THE ADDRESS TO BE TESTED
                                        BPL     19$                     ;GO DO NEXT ADDRESS
                                        ENDTST
                                L10057:
                                        TRAP    C$ETST
```

TEST 25: CHECK THAT WORDS ARE WRITTEN IN SIM RAM ON ODD ADDRESS SEQ 0107

```
                                .SBTTL TEST 25: CHECK THAT WORDS ARE WRITTEN IN SIM RAM ON ODD ADDRESS
                                
                                ;++
                                ; THE FOLLOWING TEST WILL WRITE THE MEMORY SIMULATOR RAM'S WITH A COMPLEMENTING
                                ; DATA PATTERN. THE INITIAL DATA PATTERN WILL BE 125252. EACH CONSECATIVE
                                ; LOCATION WILL CONTAIN THE COMPLEMENTED DATA OF THE PREVIOUS LOCATION (I.E.
                                ; 125252, 052525, 125252 .... ETC). THE ADDRESSES TO THE MEMORY SIMULATOR RAMS
                                ; WILL BE LOADED AS ODD ADDRESSES. THIS TEST IS DONE TO INSURE THAT ODD ADDRESSES
                                ; IN 16 BIT MODE DO NOT DISABLE THE WRITING OR READING OF THE LOW BYTE OF THE
                                ; MEMORY SIMULATOR RAM. AFTER ALL THE RAM'S HAVE BEEN LOADED WITH THE COMPLE-
                                ; MENTING DATA PATTERN, THE TEST WILL REREAD THE MEMORY SIMULATOR RAMS USING
                                ; EVEN ADDRESSES AND CHECKING THAT THE RAM'S CONTAIN THE COMPLEMENTING DATA
                                ; PATTERN
                                ;
                                ; NOTE:
                                ;       WHEN AN ERROR OCCURS, THE INFORMATION PROVIDED IN CONTROL REGISTER 4
                                ;       INDICATES THE ADDRESS BEING TESTED AND THE RAM BEING SELECTED.  THE
                                ;       TABLE BELOW INDICATES THE 4K RAM SELECTED FOR THE ADDRESSES IN ERROR.
                                ;           1ST 4K OF RAM - ADDRESSES 000000 - 017777
                                ;           2ND 4K OF RAM - ADDRESSES 020000 - 037777
                                ;           3RD 4K OF RAM - ADDRESSES 040000 - 057777
                                ;           4TH 4K OF RAM - ADDRESSES 060000 - 077777
                                ;
                                ;--
                                
016234                                  BGNTST
016234                          T25::
016234  004737  004546                  JSR     PC,INITMS               ;SELECT AND INIT MEMORY SIMULATOR
016240  012737  000001  002276          MOV     #BIT0,R4LOAD            ;SET 1ST ADDRESS TO BE TESTED TO 1
016246  012701  000001                  MOV     #BIT0,R1                ;SETUP DATA FOR MODULE SELECT RAM 0
016252  012703  125252                  MOV     #125252,R3              ;INIT DATA PATTERN TO 125252
                                
016256                          1$:     BGNSEG
016256  104404                          TRAP    C$BSEG
                                
                                        ;SET SIGNAL MSEL0 H TO A ONE AND CLEAR SIGNALS MSEL1 H, MSAD17 H AND
                                        ;MSAD16 H IN CONTROL REGISTER 2. MSEL0 H ON A ONE AND MSEL1 H ON
                                        ;A ZERO WILL CAUSE THE SIGNAL SMDS0 L TO BE ASSERTED ON A WRITE OR
                                        ;READ COMMAND TO CONTROL REGISTER 6. SMDS0 L WILL SELECT MODULE SELECT
                                        ;RAM 0.
                                
016260  012737  000004  002264          MOV     #MSEL0,R2LOAD           ;SETUP BITS TO BE LOADED
016266  012737  177740  002270          MOV     #177740,R2MASK          ;SETUP REGISTER 2 MASK WORD
016274  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD,READ AND CHECK REG 2
016300  001405                          BEQ     2$                      ;IF LOADED OK THEN CONT
016302                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
016302  104455                          TRAP    C$ERDF
016304  000002                          .WORD   2
016306  000000                          .WORD   0
016310  003710                          .WORD   R2EROR
016312                                  CKLOOP
016312  104406                          TRAP    C$CLP1
                                
                                        ;LOAD THE ADDRESS TO BE TESTED INTO CONTROL REGISTER 4. THE ADDRESS
                                        ;WILL APPEAR ON MSAD BITS 15-0 WHICH IS THE OUTPUT OF CONTROL REGISTER 4
```

```
5516  016314  004737  005460          2$:     JSR     PC,LDRDR4               ;GO LOAD,READ AND CHECK REGISTER 4
5517  016320  001405                          BEQ     3$                      ;IF LOADED OK THEN CONTINUE
5518  016322                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
5519  016322  104455                          TRAP    C$ERDF
5520  016324  000003                          .WORD   3
5521  016326  000000                          .WORD   0
5522  016330  003776                          .WORD   R4EROR
5523  016332                                  CKLOOP
5524  016332  104406                          TRAP    C$CLP1
5525
5526                                          ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 0 AND CHECK THAT IS WAS
5527                                          ;WRITTEN CORRECTLY. THE DATA PATTERNS THAT WILL BE WRITTEN ARE 1, 2, 4,
5528                                          ;10.  THESE PATTERNS WILL ENABLE THE SIGNALS EN0 H, EN1 H, EN2 H, AND
5529                                          ;EN3 H RESPECTIVELY. THESE SIGNALS ARE USED TO SELECT MEMORY SIMULATOR
5530                                          ;MEMORY WHEN THE SIGNAL SSM L IS ASSERTED LATER ON IN THE TEST.
5531
5532  016334  010137  002302          3$:     MOV     R1,R6LOAD               ;GET THE PATTERN TO BE LOADED
5533  016340  012737  177760  002306          MOV     #177760,R6MASK          ;SETUP REGISTER 6 MASK WORD
5534  016346  004737  005504                  JSR     PC,LDRDR6               ;GO LOAD, READ AND CHECK RAM 0
5535  016352  001405                          BEQ     4$                      ;IF OK THEN CONTINUE
5536  016354                                  ERRDF   4,MSGMD0,ALINFO         ;DATA ERROR IN MODULE SELECT RAM 0
5537  016354  104455                          TRAP    C$ERDF
5538  016356  000004                          .WORD   4
5539  016360  002511                          .WORD   MSGMD0
5540  016362  004050                          .WORD   ALINFO
5541  016364                                  CKLOOP
5542  016364  104406                          TRAP    C$CLP1
5543
5544                                          ;SET SIGNALS MSEL0 H AND MSEL1 H TO A ONE AND SIGNALS MSAD17 H AND MSAD16 H
5545                                          ;TO A ZERO IN CONTROL REGISTER 2. MSEL0 H AND MSEL1 H ON A ONE, WILL
5546                                          ;CAUSE THE SIGNAL SMDS1 L TO BE ASSERTED ON A WRITE OR READ TO CONTROL
5547                                          ;REGISTER 6. SMDS1 L WILL SELECT MODULE SELECT RAM 1. MSAD BITS 17 AND 16
5548                                          ;ON A ZERO WILL SELECT ADDRESS 0 OF MODULE SELECT RAM 1.
5549
5550  016366  012737  000014  002264  4$:     MOV     #MSEL0!MSEL1,R2LOAD     ;SETUP BITS TO BE LOADED
5551  016374  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REGISTER 2
5552  016400  001405                          BEQ     5$                      ;IF LOADED OK THEN CONTINUE
5553  016402                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
5554  016402  104455                          TRAP    C$ERDF
5555  016404  000002                          .WORD   2
5556  016406  000000                          .WORD   0
5557  016410  003710                          .WORD   R2EROR
5558  016412                                  CKLOOP
5559  016412  104406                          TRAP    C$CLP1
5560
5561                                          ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 1 AND CHECK THAT THE
5562                                          ;PATTERN WAS LOADED CORRECTLY. A DATA PATTERN OF ALL ONES WILL BE
5563                                          ;WRITTEN, HOWEVER, WHEN THE LOCATION IS READ BACK, A DATA PATTERN
5564                                          ;EQUAL TO ONE WILL BE READ. SETTING THE LEAST SIGNIFICANT BIT IN
5565                                          ;THE DATA PATTERN WILL FORCE THE MORE SIGNIFICANT BITS TO ZERO.
5566
5567  016414  012737  000017  002302  5$:     MOV     #BIT3!BIT2!BIT1!BIT0,R6LOAD ;GET DATA PATTERN TO BE LOADED
5568  016422  012737  000001  002304          MOV     #BIT0,R6GOOD            ;SETUP EXPECTED DATA PATTERN
5569  016430  004737  005512                  JSR     PC,LDRD6R               ;GO LOAD,READ AND CHECK RAM 1
5570  016434  001404                          BEQ     6$                      ;IF DATA OK THEN CONTINUE
5571  016436                                  ERRDF   4,MSGMD1,ALINFO         ;DATA ERROR IN MUDULE SELECT RAM 1
```

```
5572  016436  104455                          TRAP    C$ERDF
5573  016440  000004                          .WORD   4
5574  016442  002553                          .WORD   MSGMD1
5575  016444  004050                          .WORD   ALINFO
5576  016446                          6$:     ENDSEG
5577  016446                          10000$:
5578  016446  104405                          TRAP    C$ESEG
5579
5580  016450  012702  010000                  MOV     #4096.,R2               ;SETUP 4K WORD COUNTER TO DO 1 RAM
5581
5582  016454                          7$:     BGNSEG
5583  016454  104404                          TRAP    C$BSEG
5584
5585                                          ;CLEAR BITS MSEL0 H, MSEL1 H, MSAD17 H AND MSAD16 H IN CONTROL REGISTER
5586                                          ;2. MSEL0 H AND MSEL1 H ON A ZERO WILL CAUSE THE SIGNAL SSM L TO BE
5587                                          ;ASSERTED ON A WRITE OR READ TO CONTROL REGISTER 6. ON A WRITE OR READ
5588                                          ;TO CONTROL REGISTER 6, DATA WILL BE WRITTEN OR READ FROM THE MEMORY
5589                                          ;SIMULATOR RAM SELECTED. THE SIMULATOR MEMORY IS SELECTED VIA THE
5590                                          ;MODULE SELECT RAM'S (0 AND 1).
5591
5592  016456  005037  002264                  CLR     R2LOAD                  ;SETUP TO CLEAR REGISTER 2
5593  016462  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REG 2
5594  016466  001405                          BEQ     8$                      ;IF LOADED OK THEN CONTINUE
5595  016470                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
5596  016470  104455                          TRAP    C$ERDF
5597  016472  000002                          .WORD   2
5598  016474  000000                          .WORD   0
5599  016476  003710                          .WORD   R2EROR
5600  016500                                  CKLOOP
5601  016500  104406                          TRAP    C$CLP1
5602
5603                                          ;LOAD MEMORY SIMULATOR ADDRESS TO BE TESTED INTO CONTROL REGISTER 4
5604
5605  016502  004737  005460          8$:     JSR     PC,LDRDR4               ;GO LOAD,READ AND CHECK REGISTER 4
5606  016506  001405                          BEQ     9$                      ;IF LOADED OK THEN CONTINUE
5607  016510                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
5608  016510  104455                          TRAP    C$ERDF
5609  016512  000003                          .WORD   3
5610  016514  000000                          .WORD   0
5611  016516  003776                          .WORD   R4EROR
5612  016520                                  CKLOOP
5613  016520  104406                          TRAP    C$CLP1
5614
5615                                          ;LOAD DATA PATTERN 125252 OR 052525 INTO MEMORY SIMULATOR RAM. ON A
5616                                          ;WRITE OR READ TO CONTROL REGISTER 6, THE SIGNAL SSM L WILL BE ASSERTED.
5617                                          ;THIS SIGNAL WILL ENABLE BOTH MODULE SELECT RAMS AND ENABLE THE MEMORY
5618                                          ;SIMULATOR RAM TO BE WRITTEN OR READ.
5619
5620  016522  010337  002302          9$:     MOV     R3,R6LOAD               ;SETUP DATA PATTERN TO BE LOADED
5621  016526  005037  002306                  CLR     R6MASK                  ;SETUP TO CHECK ALL 16 BITS OF RAM
5622  016532  004737  005504                  JSR     PC,LDRDR6               ;GO LOAD, READ AND CHECK RAM LOCATION
5623  016536  001404                          BEQ     10$                     ;IF DATA OK THEN GO COMPLEMENT IT
5624  016540                                  ERRDF   4,MSGMSR,ALINFO         ;DATA ERROR IN MEMORY SIMULATOR RAM
5625  016540  104455                          TRAP    C$ERDF
5626  016542  000004                          .WORD   4
5627  016544  002730                          .WORD   MSGMSR
```

```
5628  016546  004050                          .WORD   ALINFO
5629  016550                          10$:    ENDSEG
5630  016550                          10001$:
5631  016550  104405                          TRAP    C$ESEG
5632
5633                                          ;UPDATE MEMORY SIMULATOR ADDRESS TO NEXT ADDRESS TO BE LOADED
5634                                          ;COMPLEMENT THE DATA PATTERN FOR NEXT LOAD TO RAM
5635
5636  016552  005103                          COM     R3                      ;COMPLEMENT THE DATA
5637  016554  062737  000002  002276          ADD     #2,R4LOAD               ;UPDATE ADDRESS BY 2
5638  016562  005302                          DEC     R2                      ;DECREMENT 2K WORD COUNTER
5639  016564  001333                          BNE     7$                      ;IF NOT DONE DO NEXT ADDRESS
5640
5641                                          ;UPDATE MODULE SELECT RAM ENABLE BITS TO BE LOADED INTO MODULE
5642                                          ;SELECT RAM 0 ON NEXT PASS OF THIS TEST
5643
5644  016566  006301                          ASL     R1                      ;MOVE MODULE SELECT RAM 0 DATA PATTERN
5645                                                                          ;LEFT TO ENABLE NEXT MEMORY SIMULATOR RAM
5646  016570  032701  000020                  BIT     #BIT4,R1                ;CHECK IF ALL RAMS HAVE BEEN TESTED
5647  016574  001002                          BNE     11$                     ;IF YES THEN EXIT THE TEST
5648  016576  000137  016256                  JMP     1$                      ;OTHERWISE DO NEXT RAM
5649
5650                                          ;THE FOLLOWING SECTION OF CODE WILL READ THE MEMORY SIMULATOR RAMS
5651                                          ;AND CHECK EACH ADDRESS TO CONTAIN EITHER 125252 OR 052525.
5652
5653  016602  005037  002276          11$:    CLR     R4LOAD                  ;RESET THE ADDRESS TO 0
5654  016606  012703  125252                  MOV     #125252,R3              ;RESET THE INITIAL DATA PATTERN
5655
5656  016612                          12$:    BGNSEG
5657  016612  104404                          TRAP    C$BSEG
5658
5659                                          ;LOAD THE ADDRESS TO BE CHECKED INTO CONTROL REGISTER 4. THE MODULE
5660                                          ;SELECT RAMS HAVE BEEN INITIALIZED PREVIOUSLY.
5661
5662  016614  004737  005460                  JSR     PC,LDRDR4               ;GO LOAD, READ AND CHECK REGISTER 4
5663  016620  001405                          BEQ     13$                     ;IF LOADED OK THEN CONTINUE
5664  016622                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED.
5665  016622  104455                          TRAP    C$ERDF
5666  016624  000003                          .WORD   3
5667  016626  000000                          .WORD   0
5668  016630  003776                          .WORD   R4EROR
5669  016632                                  CKLOOP
5670  016632  104406                          TRAP    C$CLP1
5671
```

```
5672
5673                                        ;READ THE MEMORY SIMULATOR RAM AND CHECK THE COMPLEMENTING DATA PATTERN
5674
5675  016634  010337  002302        13$:   MOV     R3,R6LOAD              ;SETUP DATA THAT WAS PREVIOUSLY WRITTEN
5676  016640  013737  002302  002304       MOV     R6LOAD,R6GOOD          ;SETUP EXPECTED DATA
5677  016646  005037  002306               CLR     R6MASK                 ;SET REGISTER 6 MASK TO ZERO
5678  016652  004737  005520               JSR     PC,READR6              ;GO READ AND CHECK THE RAM LOCATION
5679  016656  001404                       BEQ     14$                    ;IF DATA EQUALS ADDRESS THEN CONTINUE
5680  016660                               ERRDF   4,MSGMSR,ALINFO        ;DATA ERROR IN MEMORY SIMULATOR RAM
5681  016660  104455                       TRAP    C$ERDF
5682  016662  000004                       .WORD   4
5683  016664  002730                       .WORD   MSGMSR
5684  016666  004050                       .WORD   ALINFO
5685  016670                       14$:    ENDSEG
5686  016670                       10002$:
5687  016670  104405                       TRAP    C$ESEG
5688
5689  016672  005103                       COM     R3                     ;COMPLEMENT THE DATA PATTERN
5690  016674  062737  000002  002276       ADD     #2,R4LOAD              ;UPDATE THE ADDRESS TO THE NEXT LOCATION
5691  016702  100343                       BPL     12$                    ;IF NOT DONE CHECK NEXT ADDRESS
5692
5693  016704                               ENDTST
5694  016704                       L10060:
5695  016704  104401                       TRAP    C$ETST
5696
```

```
                                .SBTTL TEST 26: CHECK MEM SIM RAM'S USING 'MOVB'' INSTRUCTION

                                ;++
                                ; THE FOLLOWING TEST WILL WRITE INTO AND READ FROM THE MEMORY SIMULATOR RAMS
                                ; USING THE 'MOVB'' INSTRUCTION. THIS TEST IS DONE IN 16 BIT MODE. THE PURPOSE
                                ; OF THIS TEST IS TO CHECK THAT WHEN WRITING TTHE LOW BYTE OF AN ADDRESS, THE
                                ; HIGH BYTE IS NOT AFFECTED, AND WHEN WRITING THE HIGH BYTE OF AN ADDRESS, THE
                                ; LOW BYTE IS NOT AFFECTED. THE TEST SEQUENCE IS AS FOLLOWS:
                                ;    1. WRITE THE ADDRESS TO BE TESTED IN CONTROL REGISTER 4
                                ;    2. WRITE THE MODULE SELECT RAM'S TO ENABLE THE MEMORY SIMULATOR
                                ;    3. WRITE AND CHECK LOW BYTE FOR DATA PATTERN OF 125
                                ;    4. WRITE AND CHECK HIGH BYTE FOR DATA PATTERN OF 252
                                ;    5. READ LOCATION AND CHECK WORD FOR DATA PATTERN OF 125125
                                ;    6. WRITE AND CHECK LOW BYTE FOR DATA PATTERN OF 252
                                ;    7. READ LOCATION AND CHECK WORD FOR DATA PATTERN OF 125252
                                ;    8. WRITE AND CHECK HIGH BYTE FOR DATA PATTERN OF 125
                                ;    9. READ LOCATION AND CHECK WORD FOR DATA PATTERN OF 052652
                                ;   10. REPEAT STEPS 1-9 FOR ADDRESS 0 OF EACH MEMORY SIMULATOR RAM
                                ;
                                ; NOTE:
                                ;       WHEN AN ERROR OCCURS, THE INFORMATION PROVIDED IN CONTROL REGISTER 4
                                ;       INDICATES THE ADDRESS BEING TESTED AND THE RAM BEING SELECTED.  THE
                                ;       TABLE BELOW INDICATES THE 4K RAM SELECTED FOR THE ADDRESSES IN ERROR.
                                ;           1ST 4K OF RAM - ADDRESSES 000000 - 017777
                                ;           2ND 4K OF RAM - ADDRESSES 020000 - 037777
                                ;           3RD 4K OF RAM - ADDRESSES 040000 - 057777
                                ;           4TH 4K OF RAM - ADDRESSES 060000 - 077777
                                ;
                                ;--

016706                                  BGNTST
016706                          T26::
016706  004737  004546                  JSR     PC,INITMS               ;SELECT AND INIT MEMORY SIMULATOR
016712  005037  002276                  CLR     R4LOAD                  ;SET 1ST ADDRESS TO BE TESTED TO 0
016716  012701  000001                  MOV     #BIT0,R1                ;SETUP DATA FOR MODULE SELECT RAM 0

016722                          1$:     BGNSEG
016722  104404                          TRAP    C$BSEG

                                        ;SET SIGNAL MSEL0 H TO A ONE AND CLEAR SIGNALS MSEL1 H, MSAD17 H AND
                                        ;MSAD16 H IN CONTROL REGISTER 2. MSEL0 H ON A ONE AND MSEL1 H ON
                                        ;A ZERO WILL CAUSE THE SIGNAL SMDS0 L TO BE ASSERTED ON A WRITE OR
                                        ;READ COMMAND TO CONTROL REGISTER 6. SMDS0 L WILL SELECT MODULE SELECT
                                        ;RAM 0.

016724  012737  000004  002264          MOV     #MSEL0,R2LOAD           ;SETUP BITS TO BE LOADED
016732  012737  177740  002270          MOV     #177740,R2MASK          ;SETUP REGISTER 2 MASK WORD
016740  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD,READ AND CHECK REG 2
016744  001405                          BEQ     2$                      ;IF LOADED OK THEN CONT
016746                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
016746  104455                          TRAP    C$ERDF
016750  000002                          .WORD   2
016752  000000                          .WORD   0
016754  003710                          .WORD   R2EROR
016756                                  CKLOOP
016756  104406                          TRAP    C$CLP1
```

```
5753
5754                                        ;LOAD THE ADDRESS TO BE TESTED INTO CONTROL REGISTER 4. THE ADDRESS
5755                                        ;WILL APPEAR ON MSAD BITS 15-0 WHICH IS THE OUTPUT OF CONTROL REGISTER 4
5756
5757  016760  004737  005460        2$:     JSR     PC,LDRDR4               ;GO LOAD,READ AND CHECK REGISTER 4
5758  016764  001405                        BEQ     3$                      ;IF LOADED OK THEN CONTINUE
5759  016766                                ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
5760  016766  104455                        TRAP    C$ERDF
5761  016770  000003                        .WORD   3
5762  016772  000000                        .WORD   0
5763  016774  003776                        .WORD   R4EROR
5764  016776                                CKLOOP
5765  016776  104406                        TRAP    C$CLP1
5766
5767                                        ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 0 AND CHECK THAT IS WAS
5768                                        ;WRITTEN CORRECTLY. THE DATA PATTERNS THAT WILL BE WRITTEN ARE 1, 2, 4,
5769                                        ;10.  THESE PATTERNS WILL ENABLE THE SIGNALS EN0 H, EN1 H, EN2 H, AND
5770                                        ;EN3 H RESPECTIVELY. THESE SIGNALS ARE USED TO SELECT MEMORY SIMULATOR
5771                                        ;MEMORY WHEN THE SIGNAL SSM L IS ASSERTED LATER ON IN THE TEST.
5772
5773  017000  010137  002302        3$:     MOV     R1,R6LOAD               ;GET THE PATTERN TO BE LOADED
5774  017004  012737  177760  002306        MOV     #177760,R6MASK          ;SETUP REGISTER 6 MASK WORD
5775  017012  004737  005504                JSR     PC,LDRDR6               ;GO LOAD, READ AND CHECK RAM 0
5776  017016  001405                        BEQ     4$                      ;IF OK THEN CONTINUE
5777  017020                                ERRDF   4,MSGMD0,ALINF0         ;DATA ERROR IN MODULE SELECT RAM 0
5778  017020  104455                        TRAP    C$ERDF
5779  017022  000004                        .WORD   4
5780  017024  002511                        .WORD   MSGMD0
5781  017026  004050                        .WORD   ALINF0
5782  017030                                CKLOOP
5783  017030  104406                        TRAP    C$CLP1
5784
5785                                        ;SET SIGNALS MSEL0 H AND MSEL1 H TO A ONE AND SIGNALS MSAD17 H AND MSAD16 H
5786                                        ;TO A ZERO IN CONTROL REGISTER 2. MSEL0 H AND MSEL1 H ON A ONE, WILL
5787                                        ;CAUSE THE SIGNAL SMDS1 L TO BE ASSERTED ON A WRITE OR READ TO CONTROL
5788                                        ;REGISTER 6. SMDS1 L WILL SELECT MODULE SELECT RAM 1. MSAD BITS 17 AND 16
5789                                        ;ON A ZERO WILL SELECT ADDRESS 0 OF MODULE SELECT RAM 1.
5790
5791  017032  012737  000014  002264  4$:     MOV     #MSEL0!MSEL1,R2LOAD     ;SETUP BITS TO BE LOADED
5792  017040  004737  005412                JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REGISTER 2
5793  017044  001405                        BEQ     5$                      ;IF LOADED OK THEN CONTINUE
5794  017046                                ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
5795  017046  104455                        TRAP    C$ERDF
5796  017050  000002                        .WORD   2
5797  017052  000000                        .WORD   0
5798  017054  003710                        .WORD   R2EROR
5799  017056                                CKLOOP
5800  017056  104406                        TRAP    C$CLP1
5801
5802                                        ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 1 AND CHECK THAT THE
5803                                        ;PATTERN WAS LOADED CORRECTLY. A DATA PATTERN OF ALL ONES WILL BE
5804                                        ;WRITTEN, HOWEVER, WHEN THE LOCATION IS READ BACK, A DATA PATTERN
5805                                        ;EQUAL TO ONE WILL BE READ. SETTING THE LEAST SIGNIFICANT BIT IN
5806                                        ;THE DATA PATTERN WILL FORCE THE MORE SIGNIFICANT BITS TO ZERO.
5807
5808  017060  012737  000017  002302  5$:     MOV     #BIT3!BIT2!BIT1!BIT0,R6LOAD ;GET DATA PATTERN TO BE LOADED
```

```
5809  017066  012737  000001  002304          MOV     #BIT0,R6GOOD            ;SETUP EXPECTED DATA PATTERN
5810  017074  004737  005512                  JSR     PC,LDRD6R               ;GO LOAD,READ AND CHECK RAM 1
5811  017100  001405                          BEQ     6$                      ;IF DATA OK THEN CONTINUE
5812  017102                                   ERRDF   4,MSGMD1,ALINFO         ;DATA ERROR IN MUDULE SELECT RAM 1
5813  017102  104455                          TRAP    C$ERDF
5814  017104  000004                          .WORD   4
5815  017106  002553                          .WORD   MSGMD1
5816  017110  004050                          .WORD   ALINFO
5817  017112                                   CKLOOP
5818  017112  104406                          TRAP    C$CLP1
5819
5820                                           ;CLEAR BITS MSEL0 H, MSEL1 H, MSAD17 H AND MSAD16 H IN CONTROL REGISTER
5821                                           ;2. MSEL0 H AND MSEL1 H ON A ZERO WILL CAUSE THE SIGNAL SSM L TO BE
5822                                           ;ASSERTED ON A WRITE OR READ TO CONTROL REGISTER 6. ON A WRITE OR READ
5823                                           ;TO CONTROL REGISTER 6, DATA WILL BE WRITTEN OR READ FROM THE MEMORY
5824                                           ;SIMULATOR RAM SELECTED. THE SIMULATOR MEMORY IS SELECTED VIA THE
5825                                           ;MODULE SELECT RAM'S (0 AND 1).
5826
5827  017114  005037  002264          6$:     CLR     R2LOAD                  ;SETUP TO CLEAR REGISTER 2
5828  017120  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REG 2
5829  017124  001404                          BEQ     7$                      ;IF LOADED OK THEN CONTINUE
5830  017126                                   ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
5831  017126  104455                          TRAP    C$ERDF
5832  017130  000002                          .WORD   2
5833  017132  000000                          .WORD   0
5834  017134  003710                          .WORD   R2EROR
5835  017136                           7$:     ENDSEG
5836  017136                           10000$:
5837  017136  104405                          TRAP    C$ESEG
5838
5839  017140  013705  002242                  MOV     REG6,R5                 ;PUT THE ADDR       ( NTROL REGISTER
5840                                                                           ;6 INTO R5
5841
5842  017144                           8$:     BGNSEG
5843  017144  104404                          TRAP    C$BSEG
5844
5845                                           ;WRITE DATA PATTERN OF 125 INTO LOW BYTE OF MEMORY SIMULATOR ADDRESS
5846
5847  017146  005037  002306                  CLR     R6MASK                  ;CLEAR REGISTER 6 MASK WORD
5848  017152  012737  000125  002302          MOV     #125,R6LOAD             ;SETUP BYTE TO BE LOADED
5849  017160  013737  002302  002304          MOV     R6LOAD,R6GOOD           ;SETUP EXPECTED DATA PATTERN
5850  017166  005037  002310                  CLR     R6READ                  ;CLEAR DATA TO BE READ (HIGH BYTE)
5851  017172  113715  002302                  MOVB    R6LOAD,(R5)             ;WRITE LOW BYTE INTO MEM SIM RAM
5852  017176  111537  002310                  MOVB    (R5),R6READ             ;READ THE LOW BYTE BACK FROM RAM
5853  017202  013737  002310  002312          MOV     R6READ,R6BAD            ;COPY DATA FOR POSSIBLE ERROR REPORT
5854  017210  023737  002304  002312          CMP     R6GOOD,R6BAD            ;CHECK BYTE WRITTEN AGAINST WORD READ
5855  017216  001405                          BEQ     9$                      ;IF LOW BYTE OK THEN CONTINUE
5856  017220                                   ERRDF   4,MSGMSR,ALINFO         ;DATA ERROR READING LOW BYTE OF RAM
5857  017220  104455                          TRAP    C$ERDF
5858  017222  000004                          .WORD   4
5859  017224  002730                          .WORD   MSGMSR
5860  017226  004050                          .WORD   ALINFO
5861  017230                                   CKLOOP
5862  017230  104406                          TRAP    C$CLP1
5863
5864                                           ;WRITE DATA PATTERN OF 252 INTO HIGH BYTE OF MEMORY SIMULATOR ADDRESS
```

```
5865
5866  017232  012737  000252  002302  9$:     MOV     #252,R6LOAD             ;SETUP DATA PATTERN TO BE LOADED
5867  017240  013737  002302  002304          MOV     R6LOAD,R6GOOD           ;SETUP EXPECTED DATA
5868  017246  005037  002310                  CLR     R6READ                  ;CLEAR DATA WORD TO BE READ
5869  017252  113765  002304  000001          MOVB    R6GOOD,1(R5)            ;WRITE HIGH BYTE INTO MEM SIM RAM
5870  017260  116537  000001  002310          MOVB    1(R5),R6READ            ;READ HIGH BYTE BACK FROM RAM
5871  017266  013737  002310  002312          MOV     R6READ,R6BAD            ;COPY DATA FOR POSSIBLE ERROR REPORT
5872  017274  023737  002304  002312          CMP     R6GOOD,R6BAD            ;CHECK BYTE WRITTEN AGAINST BYTE READ
5873  017302  001405                          BEQ     10$                     ;IF HIGH BYTE Ok THEN CONTINUE
5874  017304                                  ERRDF   4,MSGMSR,ALINFO         ;DATA ERROR READING HIGH BYTE OF RAM
5875  017304  104455                          TRAP    C$ERDF
5876  017306  000004                          .WORD   4
5877  017310  002730                          .WORD   MSGMSR
5878  017312  004050                          .WORD   ALINFO
5879  017314                                  CKLOOP
5880  017314  104406                          TRAP    C$CLP1
5881
5882                                          ;READ THE RAM LOCATION AS A WORD CHECKING THAT THE LOW AND HIGH BYTE
5883                                          ;WERE WRITTEN INTO THE CORRECT BYTE. THE WORD SHOULD BE 125125 IF THE
5884                                          ;BYTES WERE WRITTEN CORRECTLY
5885
5886  017316  012737  000125  002302  10$:     MOV     #125,R6LOAD             ;SETUP LOW BYTE THAT WAS WRITTEN
5887  017324  112737  000252  002303          MOVB    #252,R6LOAD+1           ;SETUP HIGH BYTE THAT WAS WRITTEN
5888  017332  013737  002302  002304          MOV     R6LOAD,R6GOOD           ;SETUP EXPECTED DATA PATTERN
5889  017340  004737  005520                  JSR     PC,READR6               ;GO READ AND CHECK RAM LOCATION AS WORD
5890  017344  001405                          BEQ     11$                     ;IF LOW AND HIGH BYTE OK THEN CONTINUE
5891  017346                                  ERRDF   4,MSGMSR,ALINFO         ;DATA ERROR - BYTE OPERATION FAILED
5892  017346  104455                          TRAP    C$ERDF
5893  017350  000004                          .WORD   4
5894  017352  002730                          .WORD   MSGMSR
5895  017354  004050                          .WORD   ALINFO
5896  017356                                  CKLOOP
5897  017356  104406                          TRAP    C$CLP1
5898
5899                                          ;WRITE DATA PATTERN OF 252 INTO LOW BYTE OF MEM ORY SIMULATOR ADDRESS
5900
5901  017360  012737  000252  002302  1$:     MOV     #252,R6LOAD             ;SETUP DATA TO BE LOADED
5902  017366  013737  002302  002304          MOV     R6LOAD,R6GOOD           ;SETUP EXPECTED DATA PATTERN
5903  017374  005037  002310                  CLR     R6READ                  ;CLEAR DATA WORD TO BE READ
5904  017400  113715  002302                  MOVB    R6LOAD,(R5)             ;WRITE LOw BYTE INTO RAM LOCATION
5905  017404  111537  002310                  MOVB    (R5),R6READ             ;READ LOW BYTE BACK FROM RAM LOCATION
5906  017410  013737  002310  002312          MOV     R6READ,R6BAD            ;COPY DATA READ FOR POSSIBLE ERROR
5907  017416  023737  002304  002312          CMP     R6GOOD,R6BAD            ;CHECK BYTE WRITTEN AGAINST BYTE READ
5908  017424  001405                          BEQ     12$                     ;IF DATA BYTE OK THEN CONTINUE
5909  017426                                  ERRDF   4,MSGMSR,ALINFO         ;DATA ERROR READING LOW BYTE OF RAM
5910  017426  104455                          TRAP    C$ERDF
5911  017430  000004                          .WORD   4
5912  017432  002730                          .WORD   MSGMSR
5913  017434  004050                          .WORD   ALINFO
5914  017436                                  CKLOOP
5915  017436  104406                          TRAP    C$CLP1
5916
5917                                          ;READ RAM LOCATION AS A WORD CHECKING THE DATA PATTERN TO BE 125252
5918
5919  017440  012737  000252  002302  12$:     MOV     #252,R6LOAD             ;BYTE WHICH WAS JUST LOADED
5920  017446  112737  000252  002303          MOVB    #252,R6LOAD+1           ;BYTE THAT WAS PREVIOUSLY WRITTEN
```

```
5921  017454  013737  002302  002304          MOV     R6LOAD,R6GOOD           ;SETUP EXPECTED DATA PATTERN
5922  017462  004737  005520                  JSR     PC,READR6               ;GO READ AND CHECK RAM LOC AS A WORD
5923  017466  001405                          BEQ     13$                     ;IF DATA OK THEN CONTINUE
5924  017470                                  ERRDF   4,MSGMSR,ALINFO         ;WRITING LOW BYTE CHANGED HIGH BYTE
5925  017470  104455                          TRAP    C$ERDF
5926  017472  000004                          .WORD   4
5927  017474  002730                          .WORD   MSGMSR
5928  017476  004050                          .WORD   ALINFO
5929  017500                                  CKLOOP
5930  017500  104406                          TRAP    C$CLP1
5931
5932                                          ;WRITE DATA PATTERN OF 125 INTO HIGH BYTE OF MEMORY SIMULATOR ADDRESS
5933
5934  017502  012737  000125  002302  13$:    MOV     #125,R6LOAD             ;SETUP DATA PATTERN TO BE LOADED
5935  017510  013737  002302  002304          MOV     R6LOAD,R6GOOD           ;SETUP EXPECTED DATA
5936  017516  005037  002310                  CLR     R6READ                  ;CLEAR DATA WORD TO BE READ
5937  017522  113765  002302  000001          MOVB    R6LOAD,1(R5)            ;WRITE HIGH BYTE INTO RAM LOCATION
5938  017530  116537  000001  002310          MOVB    1(R5),R6READ            ;READ HIGH BYTE BACK FROM RAM LOCATION
5939  017536  013737  002310  002312          MOV     R6READ,R6BAD            ;COPY DATA READ FOR POSSIBLE ERROR
5940  017544  023737  002304  002312          CMP     R6GOOD,R6BAD            ;CHECK BYTE WRITTEN AGAINST BYTE READ
5941  017552  001405                          BEQ     14$                     ;IF HIGH BYTE OK THEN CONTINUE
5942  017554                                  ERRDF   4,MSGMSR,ALINFO         ;DATA ERROR READING HIGH BYTE
5943  017554  104455                          TRAP    C$ERDF
5944  017556  000004                          .WORD   4
5945  017560  002730                          .WORD   MSGMSR
5946  017562  004050                          .WORD   ALINFO
5947  017564                                  CKLOOP
5948  017564  104406                          TRAP    C$CLP1
5949
5950                                          ;READ RAM LOCATION CHECKING DATA PATTERN TO BE 052652
5951
5952  017566  012737  000252  002302  14$:    MOV     #252,R6LOAD             ;SETUP LOW BYTE THAT WAS WRITTEN
5953  017574  112737  000125  002303          MOVB    #125,R6LOAD+1           ;SETUP HIGH BYTE THAT WAS WRITTEN
5954  017602  013737  002302  002304          MOV     R6LOAD,R6GOOD           ;SETUP EXPECTED WORD TO BE READ
5955  017610  004737  005520                  JSR     PC,READR6               ;GO READ RAM LOCATION AS A WORD
5956  017614  001404                          BEQ     15$                     ;IF DATA OK THEN CONTINUE
5957  017616                                  ERRDF   4,MSGMSR,ALINFO         ;WRITING HIGH BYTE CHANGED LOW BYTE
5958  017616  104455                          TRAP    C$ERDF
5959  017620  000004                          .WORD   4
5960  017622  002730                          .WORD   MSGMSR
5961  017624  004050                          .WORD   ALINFO
5962  017626                          15$:    ENDSEG
5963  017626                          10001$:
5964  017626  104405                          TRAP    C$ESEG
5965
5966  017630  006301                          ASL     R1                      ;UPDATE MODULE SELECT RAM 0 DATA
5967  017632  062737  020000  002276          ADD     #MSAD13,R4LOAD          ;UPDATE ADDRESS TO 0 OF NEXT RAM
5968  017640  100402                          BMI     16$                     ;IF ADDRESS 0 OF EACH RAM DONE - EXIT
5969  017642  000137  016722                  JMP     1$                      ;RETURN TO WRITE MODULE SELECT RAM AND
5970                                                                          ;MEMORY SIMULATOR ADDRESS
5971  017646                          16$:    ENDTST
5972  017646                          L10061:
5973  017646  104401                          TRAP    C$ETST
5974
```

```
                                .SBTTL TEST 27: CHECK MEM SIM RAM'S IN 8 BIT MODE

                                ;++
                                ; THE FOLLOWING TEST WILL CHECK THAT THE MEMORY SIMULATOR RAM CAN BE WRITTEN
                                ; AND READ IN 8 BIT MODE. ADDRESS 0 OF EACH MEMORY SIMULATOR RAM WILL BE TESTED
                                ; IN 8 BIT MODE. THIS TEST WILL CHECK 8 BIT MODE AS FOLLOWS:
                                ;    1. SET MEMORY SIMULATOR TO 16 BIT MODE
                                ;    2. LOAD ADDRESS 0 OF MEMORY SIMULATOR RAM SELECTED
                                ;    3. WRITE DATA PATTERN 125252 INTO RAM AND CHECK THE DATA PATTERN
                                ;    4. SET MEMORY SIMULATOR TTO 8 BIT MODE
                                ;    5. WRITE DATA PATTERN 031463 INTO RAM + CHECK LOW BYTE FOR DATA OF 063
                                ;    6. SET MEMORY SIMULATOR TO 16 BIT MODE
                                ;    7. READ DATA FROM MEMORY SIMULATOR CHECKING WORD TO BE 125063
                                ;    8. SET MEMORY SIMULATOR TO 8 BIT MODE
                                ;    9. LOAD ADDRESS 1 OF MEMORY SIMULATOR RAM SELECTED
                                ;   10. WRITE DATA PATTERN 146314 INTO RAM + CHECK HIGH BYTE FOR DATA OF 146000
                                ;   11. SET MEMORY SIMULATOR TO 16 BIT MODE
                                ;   12. READ DATA FROM MEMORY SIMULATOR CHECKING WORD TO BE 146063
                                ;
                                ; NOTE:
                                ;       WHEN AN ERROR OCCURS, THE INFORMATION PROVIDED IN CONTROL REGISTER 4
                                ;       INDICATES THE ADDRESS BEING TESTED AND THE RAM BEING SELECTED.  THE
                                ;       TABLE BELOW INDICATES THE 4K RAM SELECTED FOR THE ADDRESSES IN ERROR.
                                ;           1ST 4K OF RAM - ADDRESSES 000000 - 017777
                                ;           2ND 4K OF RAM - ADDRESSES 020000 - 037777
                                ;           3RD 4K OF RAM - ADDRESSES 040000 - 057777
                                ;           4TH 4K OF RAM - ADDRESSES 060000 - 077777
                                ;
                                ;--

 017650                                 BGNTST
 017650                         T27::
 017650  004737  004546                 JSR     PC,INITMS               ;SELECT AND INIT MEMORY SIMULATOR
 017654  005037  002276                 CLR     R4LOAD                  ;SET 1ST ADDRESS TO BE TESTED TO 0
 017660  012701  000001                 MOV     #BIT0,R1                ;SETUP DATA FOR MODULE SELECT RAM 0

 017664                         1$:     BGNSEG
 017664  104404                         TRAP    C$BSEG

                                        ;SET SIGNAL MSEL0 H TO A ONE AND CLEAR SIGNALS MSEL1 H, MSAD17 H AND
                                        ;MSAD16 H IN CONTROL REGISTER 2. MSEL0 H ON A ONE AND MSEL1 H ON
                                        ;A ZERO WILL CAUSE THE SIGNAL SMDS0 L TO BE ASSERTED ON A WRITE OR
                                        ;READ COMMAND TO CONTROL REGISTER 6. SMDS0 L WILL SELECT MODULE SELECT
                                        ;RAM 0.

 017666  012737  000004  002264         MOV     #MSEL0,R2LOAD           ;SETUP BITS TO BE LOADED
 017674  012737  177740  002270         MOV     #177740,R2MASK          ;SETUP REGISTER 2 MASK WORD
 017702  004737  005412                 JSR     PC,LDRDR2               ;GO LOAD,READ AND CHECK REG 2
 017706  001405                         BEQ     2$                      ;IF LOADED OK THEN CONT
 017710                                 ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
 017710  104455                         TRAP    C$ERDF
 017712  000002                         .WORD   2
 017714  000000                         .WORD   0
 017716  003710                         .WORD   R2EROR
 017720                                 CKLOOP
 017720  104406                         TRAP    C$CLP1
```

```
6031
6032                                        ;LOAD THE ADDRESS TO BE TESTED INTO CONTROL REGISTER 4. THE ADDRESS
6033                                        ;WILL APPEAR ON MSAD BITS 15-0 WHICH IS THE OUTPUT OF CONTROL REGISTER 4
6034
6035  017722  004737  005460        2$:     JSR     PC,LDRDR4               ;GO LOAD,READ AND CHECK REGISTER 4
6036  017726  001405                        BEQ     3$                      ;IF LOADED OK THEN CONTINUE
6037  017730                                ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
6038  017730  104455                        TRAP    C$ERDF
6039  017732  000003                        .WORD   3
6040  017734  000000                        .WORD   0
6041  017736  003776                        .WORD   R4EROR
6042  017740                                CKLOOP
6043  017740  104406                        TRAP    C$CLP1
6044
6045                                        ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 0 AND CHECK THAT IS WAS
6046                                        ;WRITTEN CORRECTLY. THE DATA PATTERNS THAT WILL BE WRITTEN ARE 1, 2, 4,
6047                                        ;10.  THESE PATTERNS WILL ENABLE THE SIGNALS EN0 H, EN1 H, EN2 H, AND
6048                                        ;EN3 H RESPECTIVELY. THESE SIGNALS ARE USED TO SELECT MEMORY SIMULATOR
6049                                        ;MEMORY WHEN THE SIGNAL SSM L IS ASSERTED LATER ON IN THE TEST.
6050
6051  017742  010137  002302        3$:     MOV     R1,R6LOAD               ;GET THE PATTERN TO BE LOADED
6052  017746  012737  177760  002306        MOV     #177760,R6MASK          ;SETUP REGISTER 6 MASK WORD
6053  017754  004737  005504                JSR     PC,LDRDR6               ;GO LOAD, READ AND CHECK RAM 0
6054  017760  001405                        BEQ     4$                      ;IF OK THEN CONTINUE
6055  017762                                ERRDF   4,MSGMD0,ALINFO         ;DATA ERROR IN MODULE SELECT RAM 0
6056  017762  104455                        TRAP    C$ERDF
6057  017764  000004                        .WORD   4
6058  017766  002511                        .WORD   MSGMD0
6059  017770  004050                        .WORD   ALINFO
6060  017772                                CKLOOP
6061  017772  104406                        TRAP    C$CLP1
6062
6063                                        ;SET SIGNALS MSEL0 H AND MSEL1 H TO A ONE AND SIGNALS MSAD17 H AND MSAD16 H
6064                                        ;TO A ZERO IN CONTROL REGISTER 2. MSEL0 H AND MSEL1 H ON A ONE, WILL
6065                                        ;CAUSE THE SIGNAL SMDS1 L TO BE ASSERTED ON A WRITE OR READ TO CONTROL
6066                                        ;REGISTER 6. SMDS1 L WILL SELECT MODULE SELECT RAM 1. MSAD BITS 17 AND 16
6067                                        ;ON A ZERO WILL SELECT ADDRESS 0 OF MODULE SELECT RAM 1.
6068
6069  017774  012737  000014  002264  4$:     MOV     #MSEL0.MSEL1,R2LOAD     ;SETUP BITS TO BE LOADED
6070  020002  004737  005412                JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REGISTER 2
6071  020006  001405                        BEQ     5$                      ;IF LOADED OK THEN CONTINUE
6072  020010                                ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
6073  020010  104455                        TRAP    C$ERDF
6074  020012  000002                        .WORD   2
6075  020014  000000                        .WORD   0
6076  020016  003710                        .WORD   R2EROR
6077  020020                                CKLOOP
6078  020020  104406                        TRAP    C$CLP1
6079
6080                                        ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 1 AND CHECK THAT THE
6081                                        ;PATTERN WAS LOADED CORRECTLY. A DATA PATTERN OF ALL ONES WILL BE
6082                                        ;WRITTEN, HOWEVER, WHEN THE LOCATION IS READ BACK, A DATA PATTERN
6083                                        ;EQUAL TO ONE WILL BE READ. SETTING THE LEAST SIGNIFICANT BIT IN
6084                                        ;THE DATA PATTERN WILL FORCE THE MORE SIGNIFICANT BITS TO ZERO.
6085
6086  020022  012737  000017  002302  5$:     MOV     #BIT3!BIT2!BIT1!BIT0,R6LOAD ;GET DATA PATTERN TO BE LOADED
```

```
6087  020030  012737  000001  002304          MOV     #BIT0,R6GOOD            ;SETUP EXPECTED DATA PATTERN
6088  020036  004737  005512                  JSR     PC,LDRD6R               ;GO LOAD,READ AND CHECK RAM 1
6089  020042  001405                          BEQ     6$                      ;IF DATA OK THEN CONTINUE
6090  020044                                  ERRDF   4,MSGMD1,ALINFO         ;DATA ERROR IN MUDULE SELECT RAM 1
6091  020044  104455                          TRAP    C$ERDF
6092  020046  000004                          .WORD   4
6093  020050  002553                          .WORD   MSGMD1
6094  020052  004050                          .WORD   ALINFO
6095  020054                                  CKLOOP
6096  020054  104406                          TRAP    C$CLP1
6097
6098                                          ;CLEAR BITS MSEL0 H, MSEL1 H, MSAD17 H AND MSAD16 H IN CONTROL REGISTER
6099                                          ;2. MSEL0 H AND MSEL1 H ON A ZERO WILL CAUSE THE SIGNAL SSM L TO BE
6100                                          ;ASSERTED ON A WRITE OR READ TO CONTROL REGISTER 6. ON A WRITE OR READ
6101                                          ;TO CONTROL REGISTER 6, DATA WILL BE WRITTEN OR READ FROM THE MEMORY
6102                                          ;SIMULATOR RAM SELECTED. THE SIMULATOR MEMORY IS SELECTED VIA THE
6103                                          ;MODULE SELECT RAM'S (0 AND 1).
6104
6105  020056  005037  002264          6$:     CLR     R2LOAD                  ;SETUP TO CLEAR REGISTER 2
6106  020062  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REG 2
6107  020066  001404                          BEQ     7$                      ;IF LOADED OK THEN CONTINUE
6108  020070                                  ERRDF   2,,R2EROR               ;REGISTER 2 NOT EQUAL EXPECTED
6109  020070  104455                          TRAP    C$ERDF
6110  020072  000002                          .WORD   2
6111  020074  000000                          .WORD   0
6112  020076  003710                          .WORD   R2EROR
6113  020100                          7$:     ENDSEG
6114  020100                          10000$:
6115  020100  104405                          TRAP    C$ESEG
6116
6117  020102                          8$:     BGNSEG
6118  020102  104404                          TRAP    C$BSEG
6119
6120                                          ;SET MEMORY SIMULATOR TO 16 BIT MODE
6121
6122  020104  042737  000010  002252          BIC     #BIT8H,R0LOAD           ;SETUP TO CLEAR 8 BIT MODE IF SET
6123  020112  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REG 0
6124  020116  001405                          BEQ     9$                      ;IF LOADED OK THEN CONTINUE
6125  020120                                  ERRDF   1,,R0EROR               ;REGISTER 0 NOT EQUAL EXPECTED
6126  020120  104455                          TRAP    C$ERDF
6127  020122  000001                          .WORD   1
6128  020124  000000                          .WORD   0
6129  020126  003622                          .WORD   R0EROR
6130  020130                                  CKLOOP
6131  020130  104406                          TRAP    C$CLP1
6132
6133                                          ;LOAD THE ADDRESS TO BE TESTED INTO CONTROL REGISTER 4
6134
6135  020132  042737  000001  002276  9$:     BIC     #BIT0,R4LOAD            ;CLEAR ODD ADDRESS BIT IF SET
6136  020140  004737  005460                  JSR     PC,LDRDR4               ;GO LOAD, READ AND CHECK REGISTER 4
6137  020144  001405                          BEQ     10$                     ;IF LOADED OK THEN CONTINUE
6138  020146                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQUAL EXPECTED
6139  020146  104455                          TRAP    C$ERDF
6140  020150  000003                          .WORD   3
6141  020152  000000                          .WORD   0
6142  020154  003776                          .WORD   R4EROR
```

```
                                                CKLOOP
                                                TRAP    C$CLP1

                                                ;WRITE DATA PATTERN OF 125252 INTO MEMORY SIMULATOR RAM ADDRESSES
                                                ;BY CONTROL REGISTER 4

                                        10$:    CLR     R6MASK                  ;CLEAR REGISTER 6 MASK WORD
                                                MOV     #125252,R6LOAD          ;SETUP WORD TO BE LOADED
                                                JSR     PC,LDRDR6               ;GO LOAD, READ AND CHECK RAM LOCATION
                                                BEQ     11$                     ;IF DATA OK THEN CONTINUE
                                                ERRDF   4,MSGMSR,ALINFO         ;DATA ERROR IN MEMORY SIMULATOR RAM
                                                TRAP    C$ERDF
                                                .WORD   4
                                                .WORD   MSGMSR
                                                .WORD   ALINFO
                                                CKLOOP
                                                TRAP    C$CLP1

                                                ;SET MEMORY SIMULATOR TO 8 BIT MODE

                                        11$:    BIS     #BIT8H,R0LOAD           ;SETUP TO SET 8 BIT MODE
                                                JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REGISTER 0
                                                BEQ     12$                     ;IF LOADED OK THEN CONTINUE
                                                ERRDF   1,,R0EROR               ;REGISTER 0 NOT EQUAL EXPECTED
                                                TRAP    C$ERDF
                                                .WORD   1
                                                .WORD   0
                                                .WORD   R0EROR
                                                CKLOOP
                                                TRAP    C$CLP1

                                                ;WRITE DATA PATTERN OF 031463 INTO MEMORY SIMULATOR RAM. ONLY THE
                                                ;LOW BYTE 063 SHOULD BE WRITTEN WHEN THE ADDRESS SELECTED IS EVEN.

                                        12$:    MOV     #31463,R6LOAD           ;SETUP BITS TO BE WRITTEN
                                                MOV     #63,R6GOOD              ;SETUP EXPECTED DATA TO BE READ
                                                MOV     #177400,R6MASK          ;SETUP TO IGNORE HIGH BYTE ON READ
                                                JSR     PC,LDRD6R               ;GGO LOAD, READ AND CHECK LOW BYTE
                                                BEQ     13$                     ;IF LOW BYTE OK THEN CONTINUE
                                                ERRDF   4,MSGMSR,ALINFO         ;8 BIT MODE FAILED TO WRITE OR READ
                                                TRAP    C$ERDF
                                                .WORD   4
                                                .WORD   MSGMSR
                                                .WORD   ALINFO
                                                CKLOOP
                                                TRAP    C$CLP1
                                                                                ;LOW BYTE OF MEMORY SIMULATOR RAM

                                                ;RESET MEMORY SIMULATOR TO 16 BIT MODE

                                        13$:    BIC     #BIT8H,R0LOAD           ;SETUP TO CLEAR 8 BIT MODE
                                                JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REGISTER 0
                                                BEQ     14$                     ;IF LOADED OK THEN CONTINUE
                                                ERRDF   1,,R0EROR               ;REGISTER 0 NOT EQUAL EXPECTED
                                                TRAP    C$ERDF
                                                .WORD   1
```

```
6199  020322  000000                          .WORD   0
6200  020324  003622                          .WORD   R0EROR
6201  020326                                  CKLOOP
6202  020326  104406                          TRAP    C$CLP1
6203
6204                                          ;READ ADDRESS 0 OF MEMORY SIMULATOR RAM CHECKING THAT ONLY THE LOW
6205                                          ;BYTE WAS CHANGED DURING 8 BIT MODE.
6206
6207  020330  012737  125063  002302  14$:    MOV     #125063,R6LOAD          ;SETUP PREVIOUS LOADED BITS
6208  020336  013737  002302  002304          MOV     R6LOAD,R6GOOD           ;SETUP EXPECTED DATA
6209  020344  C05037  002306                  CLR     R6MASK                  ;SETUP TO CHECK ALL 16 BITS
6210  020350  004737  005520                  JSR     PC,READR6               ;GO READ AND CHECK 16 BIT WORD
6211  020354  001405                          BEQ     15$                     ;IF DATA OK THEN CONTINUE
6212  020356                                  ERRDF   4,MSGMSR,ALINFO         ;8 BIT MODE CHANGED HIGH BYTE
6213  020356  104455                          TRAP    C$ERDF
6214  020360  000004                          .WORD   4
6215  020362  002730                          .WORD   MSGMSR
6216  020364  004050                          .WORD   ALINFO
6217  020366                                  CKLOOP
6218  020366  104406                          TRAP    C$CLP1
6219
6220                                          ;RESET MEMORY SIMULATOR TO 8 BIT MODE
6221
6222  020370  052737  000010  002252  15$:    BIS     #BIT8H,R0LOAD           ;SETUP TO SET 8 BIT MODE
6223  020376  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REG 0
6224  020402  001405                          BEQ     16$                     ;IF LOADED OK THEN CONTINUE
6225  020404                                  ERRDF   1,,R0EROR               ;REGISTER 0 NOT EQUAL EXPECTED
6226  020404  104455                          TRAP    C$ERDF
6227  020406  000001                          .WORD   1
6228  020410  000000                          .WORD   0
6229  020412  003622                          .WORD   R0EROR
6230  020414                                  CKLOOP
6231  020414  104406                          TRAP    C$CLP1
6232
6233                                          ;SET THE MEMORY SIMULATOR ADDRESS IN CONTROL REGGISTER 4 TO AN
6234                                          ;ODD ADDRESS
6235
6236  020416  052737  000001  002276  16$:    BIS     #BIT0,R4LOAD            ;SETUP TO SET AN ODD ADDRESS
6237  020424  004737  005460                  JSR     PC,LDRDR4               ;GO LOAD, READ AND CHECK REGISTER 4
6238  020430  001405                          BEQ     17$                     ;IF LOADED OK THEN CONTINUE
6239  020432                                  ERRDF   3,,R4EROR               ;REGISTER 4 NOT EQULA EXPECTED
6240  020432  104455                          TRAP    C$ERDF
6241  020434  000003                          .WORD   3
6242  020436  000000                          .WORD   0
6243  020440  003776                          .WORD   R4EROR
6244  020442                                  CKLOOP
6245  020442  104406                          TRAP    C$CLP1
6246
6247                                          ;WRITE DATA PATTERN OF 146314 INTO MEMORY SIMULATOR RAM, ONLY THE
6248                                          ;HIGH BYTE, 314, SHOULD BE WRITTEN WHEN THE ADDRESS IS ODD.
6249
6250  020444  012737  146314  002302  17$:    MOV     #146314,R6LOAD          ;SETUP DATA TO BE LOADED
6251  020452  012737  146000  002304          MOV     #146000,R6GOOD          ;SETUP EXPECTED DATA
6252  020460  012737  000377  002306          MOV     #377,R6MASK             ;SETUP MASK TO IGNORE LOW BYTE
6253  020466  004737  005512                  JSR     PC,LDRD6R               ;GO LOAD, READ AND CHECK HIGH BYTE
6254  020472  001405                          BEQ     18$                     ;IF HIGH BYTE OK THEN CONTINUE
```

```
                                        ERRDF   4,MSGMSR,ALINFO         ;FAILED TO WRITE HIGH BYTE INTO MEMORY
                                        TRAP    C$ERDF
                                        .WORD   4
                                        .WORD   MSGMSR
                                        .WORD   ALINFO
                                        CKLOOP
                                        TRAP    C$CLP1
                                                                        ;SIMULATOR RAM IN 8 BIT MODE

                                        ;RESET THE MEMORY SIMULATOR TO 16 BIT MODE

020506  042737  000010  002252  18$:    BIC     #BIT8H,R0LOAD           ;SETUP TO CLEAR 8 BIT MODE
020514  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REGISTER 0
020520  001405                          BEQ     19$                     ;IF LOADED OK THEN CONTINUE
020522                                  ERRDF   1,,R0EROR               ;REGISTER 0 NOT EQUAL EXPECTED DATA
020522  104455                          TRAP    C$ERDF
020524  000001                          .WORD   1
020526  000000                          .WORD   0
020530  003622                          .WORD   R0EROR
020532                                  CKLOOP
020532  104406                          TRAP    C$CLP1

                                        ;READ ADDRESS 0 AND 1 OF MEMORY SIMULATOR RAM CHECKING THAT ONLY THE
                                        ;HIGH BYTE WAS CHANGED DURING A WRITE OPERATION IN 8 BIT MODE

020534  012737  146063  002302  19$:    MOV     #146063,R6LOAD          ;SETUP THE DATA PREVIOUSLY WRITTEN
020542  013737  002302  002304          MOV     R6LOAD,R6GOOD           ;SETUP EXPECTED DATA
020550  005037  002306                  CLR     R6MASK                  ;SETUP TO READ ALL BITS
020554  004737  005520                  JSR     PC,READR6               ;GO READ AND CHECK 16 BIT WORD
020560  001404                          BEQ     20$                     ;IF DATA OK THEN CONTINUE
020562                                  ERRDF   4,MSGMSR,ALINFO         ;WRITING HIGH BYTE IN 8 BIT MODE CHANGED
020562  104455                          TRAP    C$ERDF
020564  000004                          .WORD   4
020566  002730                          .WORD   MSGMSR
020570  004050                          .WORD   ALINFO
                                                                        ;LOW BYTE OF DATA
020572  042737  000001  002276  20$:    BIC     #BIT0,R4LOAD            ;RESET ADDRESS TO BE LOADED TO 0
020600                                  ENDSEG
020600                          10001$:
020600  104405                          TRAP    C$ESEG

020602  006301                          ASL     R1                      ;UPDATE MODULE SELECT RAM 0 DATA PATTERN
020604  062737  020000  002276          ADD     #MSAD13,R4LOAD          ;UPDATE ADDRESS TO NEXT MEM SIM RAM
020612  100402                          BMI     21$                     ;IF DONE - THEN EXIT
020614  000137  017664                  JMP     1$                      ;GO SETUP FOR NEXT MEMORY SIMULATOR RAM
020620                          21$:    ENDTST
020620                          L10062:
020620  104401                          TRAP    C$ETST
```

TEST 28: CHECK THAT INIT L CLEARS REG 0 AND INIT H PRESETS RDV AND WRV SEQ 0123

```
                                .SBTTL TEST 28: CHECK THAT INIT L CLEARS REG 0 AND INIT H PRESETS RDV AND WRV

                                ;++
                                ;THE FOLLOWING TEST WILL CHECK THAT INIT L CAN CLEAR THE LOW BYTE OF CONTROL
                                ;REGISTER 0, AND THAT INIT H CAN PRESET THE WRV AND RDV F/F'S.  THIS IS
                                ;DONE BY CLEARING(1) RDV AND WRV F/F'S AND THEN ISSUING A BRESET INSTRUCTION
                                ;WHICH SHOULD PRESET(0) RDV AND WRV F/F'S.  THEN ALL ONES ARE LOADED INTO
                                ;THE LOW BYTE OF REGISTER 0 AND A BRESET INSTRUCTION IS AGAIN ISSUED WHICH
                                ;SHOULD CLEAR THE LOW BYTE OF REGISTER 0.
                                ;--

020622                                  BGNTST
020622                          T28::
020622  004737  004546                  JSR     PC,INITMS               ;SELECT AND INITIALIZE MEM SIM

020626                                  BGNSUB
020626                          T28.1:
020626  104402                          TRAP    C$BSUB

                                        ;SET SIGNAL RST H TO A ONE, AND THEN TO A ZERO

020630                                  BGNSEG
020630  104404                          TRAP    C$BSEG
020632  112737  000001  002252          MOVB    #RSTH,R0LOAD            ;SET UP TO SET RST H IN REG 0
020640  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REG 0
020644  001405                          BEQ     1$                      ;IF OK THEN CONTINUE
020646                                  ERRDF   1,,R0EROR               ;REGISTER NOT EQUAL EXPECTED
020646  104455                          TRAP    C$ERDF
020650  000001                          .WORD   1
020652  000000                          .WORD   0
020654  003622                          .WORD   R0EROR
020656                                  CKLOOP
020656  104406                          TRAP    C$CLP1
020660  042737  000001  002252  1$:     BIC     #RSTH,R0LOAD            ;SET UP TO CLEAR RST H IN REG 0
020666  004737  005344                  JSR     PC,LDRDR0               ;GO LOAD, READ AND CHECK REG 0
020672  001405                          BEQ     2$                      ;IF OK THEN CONTINUE
020674                                  ERRDF   1,,R0EROR               ;REGISTER NOT EQUAL EXPECTED
020674  104455                          TRAP    C$ERDF
020676  000001                          .WORD   1
020700  000000                          .WORD   0
020702  003622                          .WORD   R0EROR
020704                                  CKLOOP
020704  104406                          TRAP    C$CLP1

                                        ;SET SIGNAL MSEL1 H TO A ONE AND MSAD BITS 17 AND 16 TO A ZERO.

020706  012737  000010  002264  2$:     MOV     #MSEL1,R2LOAD           ;SET UP TO SET MSEL1 H TO A ONE
020714  012737  177740  002270          MOV     #177740,R2MASK          ;SET UP TO READ LOWER 4 BITS
020722  004737  005412                  JSR     PC,LDRDR2               ;GO LOAD, READ AND CHECK REG 2
020726  001405                          BEQ     3$                      ;IF OKAY THEN CONTINUE
020730                                  ERRDF   2,,R2EROR               ;REG 2 NOT EQUAL EXPECTED
020730  104455                          TRAP    C$ERDF
020732  000002                          .WORD   2
020734  000000                          .WORD   0
020736  003710                          .WORD   R2EROR
020740                                  CKLOOP
```

```
                                        TRAP    C$CLP1

                                        ;LOAD MSAD BIT 15 THROUGH 8 TO A 0 TO SELECT ADDRESS 0

020742  005037  002276         3$:    CLR     R4LOAD                  ;SET UP TO SET ALL BITS TO 0
020746  004737  005460                JSR     PC,LDRDR4               ;GO LOAD, READ AND CHECK REG 4
020752  001405                        BEQ     4$                      ;IF NO ERRORS THEN CONTINUE
020754                                ERRDF   3,,R4EROR               ;REGISTER 4 NOT ALL ZEROS
020754  104455                        TRAP    C$ERDF
020756  000003                        .WORD   3
020760  000000                        .WORD   0
020762  003776                        .WORD   R4EROR
020764                                CKLOOP
020764  104406                        TRAP    C$CLP1

                                        ;SET BITS MPIN H AND MUTB H TO A ONE, AND WRE H AND RDE H TO A ZERO.
                                        ;WRE H AND RDE H ON A ZERO WILL CLEAR(1) THE WRV AND THE RDV FLIP-
                                        ;FLOPS, WHEN BIT CK H IS TOGGLED IN CONTROL REGISTER 0.

020766  012737  000011  002302  4$:    MOV     #MPINH!MUTBH,R6LOAD     ;SET UP TO LOAD THE MAP PROTECT RAM
020774  012737  177760  002306        MOV     #177760,R6MASK          ;SET UP REG 6 MASK WORD
021002  004737  005504                JSR     PC,LDRDR6               ;GO LOAD, READ AND CHECK REG 6
021006  001405                        BEQ     5$                      ;IF OK THEN CONTINUE
021010                                ERRDF   4,MSGMP,ALINFO          ;MAP PROTECT RAM DATA ERROR
021010  104455                        TRAP    C$ERDF
021012  000004                        .WORD   4
021014  002354                        .WORD   MSGMP
021016  004050                        .WORD   ALINFO
021020                                CKLOOP
021020  104406                        TRAP    C$CLP1

                                        ;SET SIGNAL CK H IN CONTROL REGISTER 0 TO CLOCK THE WRV AND RDV F/F'S
                                        ;BOTH F/F'S SHOULD BE CLEAR(1)

021022  052737  000160  002254  5$:    BIS     #CKH!WRVH!RDVH,ROGOOD   ;SET UP EXPECTED DATA
021030  052737  000100  002252        BIS     #CKH,ROLOAD             ;SET UP BIT TO BE LOADED
021036  004737  005352                JSR     PC,LDRDOR               ;GO LOAD, READ AND CHECK REG 0
021042  001404                        BEQ     6$                      ;IF OK THEN CONTINUE
021044                                ERRDF   1,MSGMPL,ALROIN         ;WRV F/F OR RDV F/F PROBABLY NOT SET(0)
021044  104455                        TRAP    C$ERDF
021046  000001                        .WORD   1
021050  002461                        .WORD   MSGMPL
021052  004076                        .WORD   ALROIN
021054                          6$:   ENDSEG
021054                          10000$:
021054  104405                        TRAP    C$ESEG

                                        ;ISSUE A BRESET INSTRUCTION.  BOTH RDV AND WRV FLIP-FLOPS SHOULD BE SET

021056                                BGNSEG
021056  104404                        TRAP    C$BSEG
021060                                BRESET                          ;PRESET THE RDV AND WRV F/F'S
021060  104433                        TRAP    C$RESET
021062                                SETVEC  #4,#7$,#PRIO7
021062  012746  000340                MOV     #PRIO7,-(SP)
021066  012746  021134                MOV     #7$,-(SP)
```

```
6415  021072  012746  000004                  MOV     #4,-(SP)
6416  021076  012746  000003                  MOV     #3,-(SP)
6417  021102  104437                          TRAP    C$SVEC
6418  021104  062706  000010                  ADD     #10,SP
6419  021110  013705  002234                  MOV     REG0,R5                 ;SAVE ADDRESS OF REG 0
6420  021114  113765  002245  000001          MOVB    IDDEV+1,1(R5)           ;SAVE ID NUMBER
6421  021122  000240                          NOP
6422  021124                                  CLRVEC  #4                      ;RELEASE TIMEOUT VECTOR
6423  021124  012700  000004                  MOV     #4,R0
6424  021130  104436                          TRAP    C$CVEC
6425  021132  000420                          BR      8$                      ;IF NO DEVICE TIMEOUT THEN CONTINUE
6426
6427                                          ;A DEVICE TIMEOUT OCCURED WHICH INDICATES THAT THERE IS NO DEVICE 0
6428                                          ;IN THE SYSTEM, THEREFORE, THE MEMORY SIMULATOR HAS TO BE RESELECTED
6429                                          ;BY DOING A "MOV WORD" OPERATION.  A "MOVB" OPERATION PERFORMED ABOVE
6430                                          ;DOES A READ/MODIFY WRITE.  THEREFORE, IF THERE IS NO DEVICE 0 IN THE
6431                                          ;SYSTEM, A DEVICE TIMEOUT WILL OCCUR TO ADDRESS 4.
6432
6433  021134  005726                  7$:     TST     (SP)+                   ;CLEAN UP THE STACK FROM TIMEOUT
6434  021136  005726                          TST     (SP)+                   ;
6435  021140                                  CLRVEC  #4                      ;RELEASE DEVICE TIMEOUT VECTOR
6436  021140  012700  000004                  MOV     #4,R0
6437  021144  104436                          TRAP    C$CVEC
6438  021146  105037  002252                  CLRB    R0LOAD                  ;SETUP TO LOAD ALL ZEORES (RESULT OF INIT)
6439  021152  004737  005344                  JSR     PC,LDRDR0               ;RESELECT THE DEVICE AFTER "INIT"
6440  021156  001417                          BEQ     9$                      ;IF LOADED OK THEN CONTINUE
6441  021160                                  ERRDF   1,,R0EROR               ;"INIT" FAILED TO 0 RDV OR WRV F/F'S
6442  021160  104455                          TRAP    C$ERDF
6443  021162  000001                          .WORD   1
6444  021164  000000                          .WORD   0
6445  021166  003622                          .WORD   R0EROR
6446  021170                                  CKLOOP
6447  021170  104406                          TRAP    C$CLP1
6448  021172  000411                          BR      9$                      ;GO TO END OF SEGMENT IF NO LOOPING
6449
6450  021174  105037  002254          8$:     CLRB    R0GOOD                  ;CLEAR LOWER BYTE OF EXPECTED DATA
6451  021200  004737  005360                  JSR     PC,READR0               ;GO READ AND CHECK REG 0
6452  021204  001404                          BEQ     9$                      ;IF OK THEN CONTINUE
6453  021206                                  ERRDF   1,,R0EROR               ;REGISTER 0 NOT EQUAL EXPECTED
6454  021206  104455                          TRAP    C$ERDF
6455  021210  000001                          .WORD   1
6456  021212  000000                          .WORD   0
6457  021214  003622                          .WORD   R0EROR
6458  021216                          9$:     ENDSEG
6459  021216                          10001$:
6460  021216  104405                          TRAP    C$ESEG
6461  021220                                  ENDSUB
6462  021220                          L10064:
6463  021220  104403                          TRAP    C$ESUB
6464
6465  021222                                  BGNSUB
6466  021222                          T28.2:
6467  021222  104402                          TRAP    C$BSUB
6468
6469                                          ;CHECK THAT RST H, CTS H, MP H, 8 BIT H AND CK H CAN BE SET TO 1.
6470                                          ;RDV H AND WRV H WILL BE CHECKED TO BE 0.
```

```
6471
6472  021224                                    BGNSEG
6473  021224  104404                            TRAP    C$BSEG
6474  021226  052737  000117  002252            BIS     #RSTH!CTSH!MPH!BIT8H!CKH,R0LOAD ;SET ALL R/W BITS TO 1
6475  021234  004737  005344                    JSR     PC,LDRDR0                       ;GO LOAD, READ AND COMPARE REG 0
6476  021240  001404                            BEQ     1$                              ;IF OK THEN CONTINUE
6477  021242                                    ERRDF   1,,R0EROR                       ;R/W BITS NOT ALL SET
6478  021242  104455                            TRAP    C$ERDF
6479  021244  000001                            .WORD   1
6480  021246  000000                            .WORD   0
6481  021250  003622                            .WORD   R0EROR
6482  021252                            1$:     ENDSEG
6483  021252                            10000$:
6484  021252  104405                            TRAP    C$ESEG
6485
6486                                            ;ISSUE A BRESET INSTRUCTION.  THE R/W BITS SHOULD THEN BE ZEROS.
6487
6488  021254                                    BGNSEG
6489  021254  104404                            TRAP    C$BSEG
6490  021256                                    BRESET                                  ;CLEAR THE LOW BYTE OF REG 0
6491  021256  104433                            TRAP    C$RESET
6492  021260                                    SETVEC  #4,#2$,#PRI07                   ;SETUP INCASE OF DEVICE TIMEOUT
6493  021260  012746  000340                    MOV     #PRI07,-(SP)
6494  021264  012746  021332                    MOV     #2$,-(SP)
6495  021270  012746  000004                    MOV     #4,-(SP)
6496  021274  012746  000003                    MOV     #3,-(SP)
6497  021300  104437                            TRAP    C$SVEC
6498  021302  062706  000010                    ADD     #10,SP
6499  021306  013705  002234                    MOV     REG0,R5                         ;SAVE ADDRESS OF REG 0
6500  021312  113765  002245  000001            MOVB    IDDEV+1,1(R5)                   ;SAVE ID NUMBER
6501  021320  000240                            NOP
6502  021322                                    CLRVEC  #4                              ;RELEASE TIMEOUT VECTOR
6503  021322  012700  000004                    MOV     #4,R0
6504  021326  104436                            TRAP    C$CVEC
6505  021330  000420                            BR      3$                              ;NO DEVICE TIMEOUT - CONTINUE
6506
6507                                            ;A DEVCICE TIMEOUT OCCURED WHICH INDICATES THAT THERE IS NO DEVICE #0
6508                                            ;IN THE SYSTEM, THERFORE, THE MEMORY SIMULATOR HAS TO BE RESELECTED BY
6509                                            ;DOING A "MOV WORD" OPERATION.  A "MOVB" OPERATION PERFORMED ABOVE DOES
6510                                            ;A READ/MODIFY WRITE.  THERFORE, IF THERE IS NO DEVICE #0 IN THE SYSTEM,
6511                                            ;A DEVICE TIMEOUT WILL OCCUR TO ADDRESS 4.
6512
6513  021332  005726                    2$:     TST     (SP)+                           ;CLEAN UP THE STACK FROM TIMEOUT
6514  021334  005726                            TST     (SP)+                           ;
6515  021336                                    CLRVEC  #4                              ;RELEASE TIMEOUT VECTOR
6516  021336  012700  000004                    MOV     #4,R0
6517  021342  104436                            TRAP    C$CVEC
6518  021344  105037  002252                    CLRB    R0LOAD                          ;SETUP TO LOAD ALL ZEROES
6519  021350  004737  005344                    JSR     PC,LDRDR0                       ;GO LOAD, READ AND CHECK REG 0
6520  021354  001417                            BEQ     4$                              ;IF LOADED OK THEN CONTINUE
6521  021356                                    ERRDF   1,,R0EROR                       ;REGISTER 0 NOT EQUAL EXPECTED
6522  021356  104455                            TRAP    C$ERDF
6523  021360  000001                            .WORD   1
6524  021362  000000                            .WORD   0
6525  021364  003622                            .WORD   R0EROR
6526  021366                                    CKLOOP
```

TEST 28: CHECK THAT INIT L CLEARS REG 0 AND INIT H PRESETS RDV AND WRV SEQ 0127

```
6527  021366  104406                  TRAP    C$CLP1
6528  021370  000411                  BR      4$                      ;CONTINUE IF NO LOOPING
6529
6530  021372  105037  002254    3$:   CLRB    R0GOOD                  ;CLEAR LOWER BYTE OF EXPECTED
6531  021376  004737  005360          JSR     PC,READR0               ;GO READ AND CHECK REG 0
6532  021402  001404                  BEQ     4$                      ;IF OK THEN CONTINUE
6533  021404                          ERRDF   1,,R0EROR               ;REG 0 NOT EQUAL TO EXPECTED
6534  021404  104455                  TRAP    C$ERDF
6535  021406  000001                  .WORD   1
6536  021410  000000                  .WORD   0
6537  021412  003622                  .WORD   R0EROR
6538  021414                  4$:     ENDSEG
6539  021414                  10001$:
6540  021414  104405                  TRAP    C$ESEG
6541  021416                          ENDSUB
6542  021416                  L10065:
6543  021416  104403                  TRAP    C$ESUB
6544  021420                          ENDTST
6545  021420                  L10063:
6546  021420  104401                  TRAP    C$ETST
6547  021422                          ENDMOD
```

```
                                .TITLE PARAMETER CODING

                                .SBTTL  HARDWARE PARAMETER CODING SECTION

021422                                  BGNMOD

                                ;++
                                ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--

021422                                  BGNHRD
021422  000011                          .WORD L10066-L$HARD/2
021424                          L$HARD::

                                ;
                                ;HARDWARE P-TABLE QUESTIONS
                                ;
                                ;       ASK FOR CDS MEMORY SIMULATOR CSR ADDRESS
                                ;       ASK FOR CDS MEMORY SIMULATOR DEVICE NUMBER
                                ;

021424                                  GPRMA   MSG1,0,0,0,177777,YES
021424  000031                          .WORD   T$CODE
021426  021446                          .WORD   MSG1
021430  000000                          .WORD   T$LOLIM
021432  177777                          .WORD   T$HILIM
021434                                  GPRMD   MSG2,2,0,177777,0,000017,YES
021434  001032                          .WORD   T$CODE
021436  021462                          .WORD   MSG2
021440  177777                          .WORD   177777
021442  000000                          .WORD   T$LOLIM
021444  000017                          .WORD   T$HILIM


021446                                  ENDHRD
                                        .EVEN
021446                          L10066:

                                ;
                                ;HARDWARE P-TABLE MESSAGES
                                ;

021446  051503  020122  042101  MSG1:   .ASCIZ  /CSR ADDRESS/
021454  051104  051505  000123
021462  042504  044526  042503  MSG2:   .ASCIZ  /DEVICE NUMBER/
021470  047040  046525  042502
021476  000122

                                        .EVEN

                                .SBTTL  SOFTWARE PARAMETER CODING SECTION
```

```
                                ;++
                                ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--

021500                                  BGNSFT
021500  000000                          .WORD L10067-L$SOFT/2
021502                          L$SOFT::


                                        .EVEN

021502                                  ENDSFT
                                        .EVEN
021502                          L10067:

021502                          $PATCH::
021502  000010                          .BLKW   10

021522                                  LASTAD
                                        .EVEN
021522  021536                          .WORD T$FREE
021524  000004                          .WORD T$SIZE
021526                          L$LAST::
021526                                  ENDMOD

021526                                  BGNSETUP        1.
021526                                  BGNPTAB
021526  000000                          .WORD   0
021530  000002                          .WORD   L10072-./2-1
021532                          L10070:
021532  163010                          .WORD   163010
021534  000000                          .WORD   0
021536                                  ENDPTAB
021536                          L10072:
021536                                  ENDSETUP
        000001                  .END
```

```
ADR    = 000020 G   1344#
ALINFO  004050 G    1732#  2944   2957   3110   3139   3231   3299   3319   3424   3510   3704   3790   3984
                    4184   4270   4348   4381   4412   4444   4515   4570   4641   4685   4780   4815   4868
                    4882   4984   5019   5054   5115   5212   5247   5300   5353   5394   5408   5448   5540
                    5575   5628   5684   5781   5816   5860   5878   5895   5913   5928   5946   5961   6059
                    6094   6157   6186   6216   6259   6289   6386
ALR0IN  004076 G    1745#  3436   3467   3481   3522   3552   3579   3592   3716   3747   3761   3802   3832
                    3859   3872   3996   4027   4042   4069   4082   6401
ALR2IN  004124 G    1758#  3125   3154   3452   3494   3535   3564   3606   3732   3774   3815   3844   3886
                    4012   4055   4095
ASSEMB= 000010      1093
BIT0   = 000001 G   1317#  1382   1405   1426   1432   4166   4252   4730   4807   4806   4932   5011   5012
                    5162   5239   5240   5488   5489   5567   5568   5731   5808   5809   6009   6086   6087
                    6135   6236   6291
BIT00  = 000001 G   1306#  1317
BIT01  = 000002 G   1305#  1316
BIT02  = 000004 G   1304#  1315
BIT03  = 000010 G   1303#  1314
BIT04  = 000020 G   1302#  1313
BIT05  = 000040 G   1301#  1312
BIT06  = 000100 G   1300#  1311
BIT07  = 000200 G   1299#  1310
BIT08  = 000400 G   1298#  1309
BIT09  = 001000 G   1297#  1308
BIT1   = 000002 G   1316#  1380   1404   1425   1433   4168   4170   4254   4256   4341   4404   4807   5011
                    5239   5567   5808   6086
BIT10  = 002000 G   1296#  1369   1416
BIT11  = 004000 G   1295#  1368   1415
BIT12  = 010000 G   1294#  1414
BIT13  = 020000 G   1293#  1413
BIT14  = 040000 G   1292#  1412
BIT15  = 100000 G   1291#  1361   1411
BIT2   = 000004 G   1315#  1379   1397   1424   1434   4168   4172   4174   4254   4258   4260   4807   5011
                    5239   5567   5808   6086
BIT3   = 000010 G   1314#  1378   1396   1423   1435   4168   4172   4176   4254   4258   4262   4374   4436
                    4807   5011   5239   5567   5808   6086
BIT4   = 000020 G   1313#  1377   1422   2526   4279   4648   4692   4898   5062   5316   5646
BIT5   = 000040 G   1312#  1376   1392   1421
BIT6   = 000100 G   1311#  1375   1391   1420
BIT7   = 000200 G   1310#  1390   1419
BIT8   = 000400 G   1309#  1371   1418   2791   2819
BIT8H  = 000010 G   1378#  2410   2474   6122   6163   6193   6222   6266   6474
BIT9   = 001000 G   1308#  1370   1417
BOE    = 000400 G   1348#
CKH    = 000100 G   1375#  2410   2456   2528   2530   3459   3460   3473   3474   3544   3545   3739   3740
                    3753   3754   3824   3825   4019   4020   4034   4035   6393   6394   6474
CTSH   = 000002 G   1380#  2410   2474   6474
C$AU   = 000052     1093#  2345
C$AUTO= 000061      1093#  2274
C$BRK  = 000022     1093#
C$BSEG  000004      1093#  1926   1971   1988   2006   2024   2067   2409   2427   2455   2473   2509   2561
                    2578   2604   2621   2656   2690   2706   2736   2752   2779   2808   2841   2901   3033
                    3189   3254   3305   3349   3629   3908   4124   4223   4302   4469   4527   4611   4655
                    4733   4823   4936   5073   5165   5255   5326   5369   5422   5493   5583   5657   5734
                    5843   6012   6118   6325   6409   6473   6489
C$BSUB= 000002      1093#  2403   2447   2555   2596   3185   3245   6320   6467
```

```
C$CEFG= 000045  1093#
C$CLCK= 000062  1093#
C$CLEA= 000012  1093#  23..
C$CLOS= 000035  1093#
C$CLP1= 000006  1093#  1952  2047  2086  2856  2917  2930  2946  3052  3066  3082  3095  3112
                3127  3141  3204  3217  3269  3282  3368  3382  3397  3410  3426  3438  3454
                3469  3483  3496  3512  3524  3537  3554  3566  3581  3594  3648  3662  3677
                3690  3706  3718  3734  3749  3763  3776  3792  3804  3817  3834  3846  3861
                3874  3928  3942  3957  3970  3986  3998  4014  4029  4044  4057  4071  4084
                4141  4155  4241  4320  4334  4350  4367  4383  4397  4414  4429  4484  4499
                4541  4554  4626  4669  4751  4764  4782  4799  4841  4853  4870  4954  4968
                4986  5003  5021  5039  5089  5101  5183  5196  5214  5231  5273  5285  5339
                5381  5396  5434  5511  5524  5542  5559  5601  5613  5670  5752  5765  5783
                5800  5818  5862  5880  5897  5915  5930  5948  6030  6043  6061  6078  6096
                6131  6144  6159  6172  6188  6202  6218  6231  6245  6261  6275  6335  6345
                6359  6372  6388  6447  6527
C$CVEC= 000036  1093#  1963  2058  2097  6424  6437  6504  6517
C$DCLN= 000044  1093#
C$DODU= 000051  1093#
C$DRPT= 000024  1093#
C$DU  = 000053  1093#  2322
C$EDIT= 000003  1093#  1155
C$ERDF= 000055  1093#  1947  1957  1977  1995  2011  2042  2052  2081  2091  2415  2433  2461
                2479  2517  2566  2583  2609  2626  2660  2695  2711  2741  2757  2783  2812
                2851  2866  2912  2925  2941  2954  3047  3061  3077  3090  3107  3122  3136
                3151  3199  3212  3228  3264  3277  3296  3316  3363  3377  3392  3405  3421
                3433  3449  3464  3478  3491  3507  3519  3532  3549  3561  3576  3589  3603
                3643  3657  3672  3685  3701  3713  3729  3744  3758  3771  3787  3799  3812
                3829  3841  3856  3869  3883  3923  3937  3952  3965  3981  3993  4009  4024
                4039  4052  4066  4079  4092  4136  4150  4181  4236  4267  4315  4329  4345
                4362  4378  4392  4409  4424  4441  4479  4494  4512  4536  4549  4567  4621
                4638  4664  4682  4746  4759  4777  4794  4812  4836  4848  4865  4879  4949
                4963  4981  4998  5016  5034  5051  5084  5096  5112  5178  5191  5209  5226
                5244  5268  5280  5297  5334  5350  5376  5391  5405  5429  5445  5506  5519
                5537  5554  5572  5596  5608  5625  5665  5681  5747  5760  5778  5795  5813
                5831  5857  5875  5892  5910  5925  5943  5958  6025  6038  6056  6073  6091
                6109  6126  6139  6154  6167  6183  6197  6213  6226  6240  6256  6270  6286
                6330  6340  6354  6367  6383  6398  6442  6454  6478  6522  6534
C$ERHR= 000056  1093#
C$ERRO= 000060  1093#
C$ERSF= 000054  1093#
C$ERSO= 000057  1093#
C$ESCA= 000010  1093#
C$ESEG= 000005  1093#  1966  1983  2001  2017  2061  2100  2421  2439  2467  2485  2523  2572
                2589  2615  2632  2666  2701  2717  2747  2763  2789  2818  2872  2960  3157
                3234  3302  3322  3609  3889  4098  4188  4273  4447  4518  4573  4644  4688
                4818  4885  5057  5118  5250  5303  5356  5411  5452  5578  5631  5687  5837
                5964  6115  6294  6404  6460  6484  6540
C$ESUB= 000003  1093#  2443  2489  2592  2635  3242  3330  6463  6543
C$ETST= 000001  1093#  2383  2493  2534  2638  2672  2720  2766  2795  2823  2875  2969  3166
                3333  3612  3892  4101  4198  4283  4450  4593  4703  4903  5125  5458  5695
                5973  6302  6546
C$EXIT= 000032  1093#  2249  2292  4579  4695
C$GETB= 000026  1093#
C$GETW= 000027  1093#
C$GMAN- 000043  1093#
```

```
C$GPHR= 000042           1093#   2227
C$GPLO= 000030           1093#
C$GPRI= 000040           1093#
C$INIT= 000011           1093#   2257
C$INLP= 000020           1093#
C$MANI= 000050           1093#
C$MEM = 000031           1093#
C$MSG = 000023           1093#   1686    1709    1729    1742    1755    1768    1780    1792    1804
C$OPEN= 000034           1093#
C$PNTB= 000014           1093#   1666    1694    1717    1737    1750    1763    1776    1788    1800
C$PNTF= 000017           1093#
C$PNTS= 000016           1093#
C$PNTX= 000015           1093#   1677    1705    1725    1810    1821    1827    1838    1844    1852    1858    1869
C$QIO = 000377           1093#
C$RDBU= 000007           1093#
C$REFG= 000047           1093#   2198    2203    2208    2215    2221
C$RESE= 000033           1093#   2212    6411    6491
C$REVI= 000003           1093#   1154
C$RFLA= 000021           1093#
C$RPT = 000025           1093#   2169
C$SEFG= 000046           1093#
C$SPRI= 000041           1093#   2245
C$SVEC= 000037           1093#   1932    2030    2073    6417    6497
C$TPRI= 000013           1093#
DFPTBL  002216 G         1247#
DIAGMC= 000000           1093
EF.CON= 000036 G         1324#   2220
EF.NEW= 000035 G         1325#   2214
EF.PWR= 000034 G         1326#   2207
EF.RES= 000037 G         1323#   2202
EF.STA= 000040 G         1322#   2197
EMSGR0  003044 G         1580#   1663    1747
EMSGR2  003074 G         1584#   1691    1760
EMSGR4  003124 G         1588#   1714
EMSGR6  003154 G         1592#   1734
ERRBLK  002232 G         1449#
ERRMSG  002230 G         1448#
ERRNBR  002226 G         1447#
ERRTYP  002224 G         1446#
ESRH  = 000040 G         1392#   3117    3118    3146    3444    3445    3487    3528    3724    3725    3767    3808    4004
                         4005    4048
EVL   = 000004 G         1342#
E$END = 002100           1093#
E$LOAD= 000035           1093#   1178
FRMTR0  003254 G         1604#   1674    1702    1818    1835    1866
FRMTR4  003361 G         1616#   1722    1849
F$AU  = 000015           1093#   2333    2344
F$AUTO= 000020           1093#   2269    2273
F$BGN = 000040           1093#   1100    1275    1280    1661    1689    1712    1732    1745    1758    1771    1783    1795
                         1926    1971    1988    2006    2024    2067    2145    2150    2157    2163    2179    2195    2249
                         2269    2284    2292    2310    2316    2333    2339    2348    2354    2375    2382    2398    2402
                         2409    2427    2442    2446    2455    2473    2488    2492    2505    2509    2533    2549    2554
                         2561    2578    2591    2595    2604    2621    2634    2637    2650    2656    2671    2684    2690
                         2706    2719    2730    2736    2752    2765    2775    2779    2794    2804    2808    2822    2838
                         2841    2874    2895    2901    2968    3027    3033    3165    3181    3184    3189    3241    3244
                         3254    3305    3329    3332    3346    3349    3611    3626    3629    3891    3905    3908    4100
```

```
                    4118    4124    4197    4217    4223    4282    4298    4302    4449    4463    4469    4527    4579
                    4592    4605    4611    4655    4695    4702    4727    4733    4823    4902    4928    4936    5073
                    5124    5159    5165    5255    5326    5369    5422    5457    5486    5493    5583    5657    5694
                    5728    5734    5843    5972    6006    6012    6118    6301    6315    6319    6325    6409    6462
                    6466    6473    6489    6542    6545    6548    6553    6564    6615    6634    6636    6637    6643
                    6645
F$CLEA= 000007       1093#   2284    2299
F$DU  = 000016       1093#   2310    2321
F$END = 000041       1093#   1100    1275    1280    1687    1710    1730    1743    1756    1769    1781    1793    1805
                     1967    1984    2002    2018    2062    2101    2145    2150    2161    2170    2249    2258    2275
                     2292    2301    2314    2323    2337    2346    2348    2354    2375    2382    2384    2398    2402
                     2422    2440    2442    2444    2446    2468    2486    2488    2490    249c    2494    2505    2524
                     2533    2535    2549    2554    2573    2590    2591    2593    2595    2616    2633    2634    2636
                     2637    2639    2650    2667    2671    2673    2684    2702    2718    2719    2721    2730    2748
                     2764    2765    2767    2775    2790    2794    2796    2804    2819    2822    2824    2838    2873
                     2874    2876    2895    2961    2968    2970    3027    3158    3165    3167    3181    3184    3235
                     3241    3243    3244    3303    3323    3329    3331    3332    3334    3346    3610    3611    3613
                     3626    3890    3891    3893    3905    4099    4100    4102    4118    4189    4197    4199    4217
                     4274    4282    4284    4298    4448    4449    4451    4463    4519    4574    4579    4592    4594
                     4605    4645    4689    4695    4702    4704    4727    4819    4886    4902    4904    4928    5058
                     5119    5124    5126    5159    5251    5304    5357    5412    5453    5457    5459    5486    5579
                     5632    5688    5694    5696    5728    5838    5965    5972    5974    6006    6116    6295    6301
                     6303    6315    6319    6405    6461    6462    6464    6466    6485    6541    6542    6544    6545
                     6547    6548    6553    6591    6624    6634    6636    6637    6643    6645
F$HARD= 000004       1093#   6564    6589
F$HW  = 000013       1093#   1245    1254
F$INIT= 000006       1093#   2195    2256
F$JMP = 000050       1093#   2161    2249    2292    2314    2337    4579    4695
F$MOD = 000000       1093#   1100    1275    1280    2145    2150    2348    2354    6548    6553    6634
F$MSG = 000011       1093#   1661    1685    1689    1708    1712    1728    1732    1741    1745    1754    1758    1767
                     1771    1779    1783    1791    1795    1803
F$PROT= 000021       1093#   2179    2186
F$PWR = 000017       1093#
F$RPT = 000012       1093#   2157    2168
F$SEG = 000003       1093#   1926    1965    1971    1982    1988    2000    2006    2016    2024    2060    2067    2099
                     2409    2420    2427    2438    2455    2466    2473    2484    2509    2522    2561    2571    2578
                     2588    2604    2614    2621    2631    2656    2665    2690    2700    2706    2716    2736    2746
                     2752    2762    2779    2788    2808    2817    2841    2871    2901    2959    3033    3156    3189
                     3233    3254    3301    3305    3321    3349    3608    3629    3888    3908    4097    4124    4187
                     4223    4272    4302    4446    4469    4517    4527    4572    4611    4643    4655    4687    4733
                     4817    4823    4884    4936    5056    5073    5117    5165    5249    5255    5302    5326    5355
                     5369    5410    5422    5451    5493    5577    5583    5630    5657    5686    5734    5836    5843
                     5963    6012    6114    6118    6293    6325    6403    6409    6459    6473    6483    6489    6539
F$SOFT= 000005       1093#   6615    6622
F$SRV = 000010       1093#
F$SUB = 000002       1093#   2403    2442    2447    2488    2555    2591    2596    2634    3185    3241    3245    3329
                     6320    6462    6467    6542
F$SW  = 000014       1093#   1266    1272
F$TEST= 000001       1093#   2376    2382    2399    2492    2506    2533    2550    2637    2651    2671    2685    2719
                     2731    2765    2776    2794    2805    2822    2839    2874    2896    2968    3028    3165    3182
                     3332    3347    3611    3627    3891    3906    4100    4119    4197    4218    4282    4299    4449
                     4464    4592    4606    4702    4728    4902    4929    5124    5160    5457    5487    5694    5729
                     5972    6007    6301    6316    6545
G$CNTO= 000200       1093#
G$DELM= 000372       1093#
G$DISP= 000003       1093#
```

```
G$EXCP= 000400          1093#
G$HILI= 000002          1093#
G$LOLI= 000001          1093#
G$NO  = 000000          1093#
G$OFFS= 000400          1093#   6575    6580
G$OFSI= 000376          1093#   6575    6580
G$PRMA= 000001          1093#   6575
G$PRMD= 000002          1093#   6580
G$PRML= 000000          1093#
G$RADA= 000140          1093#
G$RADB= 000000          1093#
G$RADD= 000040          1093#
G$RADL= 000120          1093#
G$RADO= 000020          1093#   6575    6580
G$XFER= 000004          1093#
G$YES = 000010          1093#   6575    6580
HELP  = 000000          1091#   1093    1108    1196    1235    1252    1270    1275#   1278    1286    1443    1501    1575
                        1649    1651    1680    1884    1889    1892    1900    1903    1910    1917    1921    1923    1924
                        2146#   2159    2164    2187    2247    2252    2271    2290    2295    2312    2317    2335    2340
                        2350#   2372    2373    2378    2379    6547    6549#   6586    6602    6618    6625    6628    6635
HOE   = 100000 G        1355#
IBE   = 010000 G        1352#
IDDEV   002244 G        1460#   1937    1989    2240*   2241*   2285    6420    6500
IDH   = 100000 G        1361#   1972    2242
IDTYPE  002250 G        1462#   1973    2242*
IDU   = 000040 G        1345#
IER   = 020000 G        1353#
INITMS  004546 G        1925#   2376    2399    2506    2550    2651    2685    2731    2776    2805    2839    2896    3028
                        3182    3347    3627    3906    4119    4218    4299    4464    4606    4728    4929    5160    5487
                        5729    6007    6316
ISR   = 000100 G        1346#
IXE   = 004000 G        1351#
I$AU  = 000041          1093#   2333#   2346#
I$AUTO= 000041          1093#   2269#   2275#
I$CLN = 000041          1093#   2284#   2292    2301#
I$DU  = 000041          1093#   2310#   2323#
I$HRD = 000041          6564#   6591#
I$INIT= 000041          1093#   2195#   2249    2258#
I$MOD = 000041          1093#   1100#   1275#   1280#   2145#   2150#   2348#   2354#   6548#   6553#   6634#
I$MSG = 000041          1093#   1661#   1687#   1689#   1710#   1712#   1730#   1732#   1743#   1745#   1756#   1758#   1769#
                        1771#   1781#   1783#   1793#   1795#   1805#
I$PROT= 000040          1093#   2179#
I$PTAB= 000041          1093#   6637#   6643#
I$PWR = 000041          1093#
I$RPT = 000041          1093#   2157#   2170#
I$SFG = 000041          1093#   1926#   1967#   1971#   1984#   1988#   2002#   2006#   2018#   2024#   2062#   2067#   2101#
                        2375    2398    2402    2409#   2422#   2427#   2440#   2446    2455#   2468#   2473#   2486#   2505
                        2509#   2524#   2549    2554    2561#   2573#   2578#   2590#   2595    2604#   2616#   2621#   2633#
                        2650    2656#   2667#   2684    2690#   2702#   2706#   2718#   2730    2736#   2748#   2752#   2764#
                        2775    2779#   2790#   2804    2808#   2819#   2838    2841#   2873#   2895    2901#   2961#   3027
                        3033#   3158#   3181    3184    3189#   3235#   3244    3254#   3303#   3305#   3323#   3346    3349#
                        3610#   3626    3629#   3890#   3905    3908#   4099#   4118    4124#   4189#   4217    4223#   4274#
                        4298    4302#   4448#   4463    4469#   4519#   4527#   4574#   4605    4611#   4645#   4655#   4689#
                        4727    4733#   4819#   4823#   4886#   4928    4936#   5058#   5073#   5119#   5159    5165#   5251#
                        5255#   5304#   5326#   5357#   5369#   5412#   5422#   5453#   5486    5493#   5579#   5583#   5632#
                        5657#   5688#   5728    5734#   5838#   5843#   5965#   6006    6012#   6116#   6118#   6295#   6315
```

```
                    6319    6325#   6405#   6409#   6461#   6466    6473#   6485#   6489#   6541#
I$SETU= 000041      1093#   6636#   6637    6645#
I$SFT = 000041      6615#   6624#
I$SRV = 000041      1093#
I$SUB = 000041      1093#   2375    2398    2402#   2442#   2444#   2446#   2488#   2490#   2505    2549    2554#   2591#
                    2593#   2595#   2634#   2636#   2650    2684    2730    2775    2804    2838    2895    3027    3181
                    3184#   3241#   3243#   3244#   3329#   3331#   3346    3626    3905    4118    4217    4298    4463
                    4605    4727    4928    5159    5486    5728    6006    6315    6319#   6462#   6464#   6466#   6542#
                    6544#
I$TST = 000041      1093#   2375#   2382#   2384#   2398#   2402    2446    2492#   2494#   2505#   2533#   2535#   2549#
                    2554    2595    2637#   2639#   2650#   2671#   2673#   2684#   2719#   2721#   2730#   2765#   2767#
                    2775#   2794#   2796#   2804#   2822#   2824#   2838#   2874#   2876#   2895#   2968#   2970#   3027#
                    3165#   3167#   3181#   3184    3244    3332#   3334#   3346#   3611#   3613#   3626#   3891#   3893#
                    3905#   4100#   4102#   4118#   4197#   4199#   4217#   4282#   4284#   4298#   4449#   4451#   4463#
                    4579    4592#   4594#   4605#   4695    4702#   4704#   4727#   4902#   4904#   4928#   5124#   5126#
                    5159#   5457#   5459#   5486#   5694#   5696#   5728#   5972#   5974#   6006#   6301#   6303#   6315#
                    6319    6466    6545#   6547#
J$JMP = 000167      1093#   2161    2314    2337
LDRDR0  005344 G    1992    2008    2106#   2412    2430    2458    2476    2514    2848    3044    3058    3360    3374
                    3573    3586    3640    3654    3853    3866    3920    3934    4063    4076    6123    6164    6194
                    6223    6267    6327    6337    6439    6475    6519
LDRDR2  005412 G    2117#   2563    2580    2606    2623    2657    2863    2909    3074    3196    3261    3389    3669
                    3949    4133    4233    4312    4359    4389    4421    4476    4533    4618    4661    4743    4791
                    4833    4946    4995    5031    5081    5175    5223    5265    5503    5551    5593    5744    5792
                    5828    6022    6070    6106    6351
LDRDR4  005460 G    2128#   2692    2708    2738    2754    2780    2809    2922    3087    3209    3274    3402    3682
                    3962    4147    4326    4491    4546    4756    4845    4960    5093    5188    5277    5331    5373
                    5426    5516    5605    5662    5757    6035    6136    6237    6364
LDRDR6  005504 G    2136#   2938    2951    3104    3133    3225    3313    3418    3504    3698    3784    3978    4342
                    4375    4509    4635    4774    4862    4876    4978    5048    5206    5294    5402    5534    5622
                    5775    6053    6151    6380
LDRD0R  005352 G    1974    2107#   3461    3475    3546    3741    3755    3826    4021    4036    6395
LDRD2R  005420 G    2118#
LDRD6R  005512 G    2137#   4178    4264    4809    5013    5241    5569    5810    6088    6180    6253
LOE   = 040000 G    1354#
LOT   = 000010 G    1343#
L$ACP   002110 G    1185#
L$APT   002036 G    1143#
L$AU    006004 G    2333#
L$AUT   002070 G    1169#
L$AUTO  005744 G    1186    2269#
L$CCP   002106 G    1183#
L$CLEA  005746 G    1184    2284#
L$CO    002032 G    1139#
L$DEPO  002011 G    1121#
L$DESC  002324 G    1176    1505#
L$DESP  002076 G    1175#
L$DEVP  002060 G    1161#
L$DISP  002124 G    1146    1205#
L$DLY   002116 G    1191#
L$DTP   002040 G    1145#
L$DTYP  002034 G    1141#
L$DU    005776 G    2310#
L$DUT   002072 G    1171#
L$DVTY  002314 G    1162    1496#
L$EF    002052 G    1156#
```

```
L$ENVI  002044 G   1149#
L$ERRT  002224 G   1445#
L$ETP   002102 G   1179#
L$EXP1  002046 G   1151#
L$EXP4  002064 G   1165#
L$EXP5  002066 G   1167#
L$HARD  021424 G   1128    6564    6565#
L$HIME  002120 G   1193#
L$HPCP  002016 G   1127#
L$HPTP  002022 G   1131#
L$HW    002216 G   1132    1245    1246#
L$ICP   002104 G   1181#
L$INIT  005566 G   1182    2195#
L$LADP  002026 G   1135#
L$LAST  021526 G   1136    6632#   6645
L$LOAD  002100 G   1177#
L$LUN   002074 G   1173#
L$MREV  002050 G   1153#
L$NAME  002000 G   1110#
L$PRIO  002042 G   1147#
L$PROT  005560 G   1188    2179#
L$PRT   002112 G   1187#
L$REPP  002062 G   1163#
L$REV   002010 G   1119#
L$RPT   005552 G   2157#
L$SOFT  021502 G   6615    6616#
L$SPC   002056 G   1159#
L$SPCP  002020 G   1129#
L$SPTP  002024 G   1133#
L$STA   002030 G   1137#
L$SW    002224 G   1266    1267#
L$TEST  002114 G   1189#
L$TIML  002014 G   1125#
L$UNIT  002012 G   1123#
L10000  002222     1245    1254#
L10001  002224     1266    1272#
L10002  003706     1685#
L10003  003774     1708#
L10004  004046     1728#
L10005  004074     1741#
L10006  004122     1754#
L10007  004150     1767#
L10010  004172     1779#
L10011  004214     1791#
L10012  004236     1803#
L10013  005556     2162    2168#
L10015  005742     2250    2256#
L10016  005744     2273#
L10017  005774     2293    2299#
L10020  006002     2315    2321#
L10021  006010     2338    2344#
L10022  006016     2382#
L10023  006222     2492#
L10024  006120     2442#
L10025  006220     2488#
L10026  006336     2533#
```

```
L10027  006520          2637#
L10030  006432          2591#
L10031  006516          2634#
L10032  006576          2671#
L10033  006662          2719#
L10034  006750          2765#
L10035  007020          2794#
L10036  007064          2822#
L10037  007162          2874#
L10040  007362          2968#
L10041  007736          3165#
L10042  010326          3332#
L10043  010112          3241#
L10044  010324          3329#
L10045  011204          3611#
L10046  012062          3891#
L10047  012570          4100#
L10050  013006          4197#
L10051  013202          4282#
L10052  013546          4449#
L10053  014056          4580    4592#
L10054  014312          4696    4702#
L10055  014702          4902#
L10056  015350          5124#
L10057  016232          5457#
L10060  016704          5694#
L10061  017646          5972#
L10062  020620          6301#
L10063  021420          6545#
L10064  021220          6462#
L10065  021416          6542#
L10066  021446          6564    6590#
L10067  021502          6615    6623#
L10070  021532          6639#
L10072  021536          6638    6643#
MPH    = 000004 G       1379#   2410    2456    2846    3042    3358    3572    3638    3852    3918    4062    6474
MPINH  = 000001 G       1432#   2937    3103    3311    3416    3503    3696    3783    3976    6378
MSAD0  = 000001 G       1426#
MSAD1  = 000002 G       1425#
MSAD10= 002000 G        1416#
MSAD11= 004000 G        1415#
MSAD12= 010000 G        1414#
MSAD13- 020000 G        1413#   4194    4520    4576    5060    5121    5967    6297
MSAD14= 040000 G        1412#
MSAD15= 100000 G        1411#
MSAD16= 000001 G        1405#   2562    2605    2964    3161    3237    3325
MSAD17- 000002 G        1404#   2562    2622
MSAD2  = 000004 G       1424#
MSAD3  = 000010 G       1423#
MSAD4  = 000020 G       1422#
MSAD5  = 000040 G       1421#
MSAD6  = 000100 G       1420#
MSAD7  = 000200 G       1419#
MSAD8  = 000400 G       1418#   2962    3159    3235    3323
MSAD9  - 001000 G       1417#
MSBRKH= 000200 G        1390#   3487    3528    3599    3767    3808    3879    4048    4088
```

```
MSEL0 = 000004 G    1397#  2562  2605  2965  3162  3238  3326  4131  4219  4310  4358  4388  4420
                    4474   4532  4607  4651  4741  4790  4944  4994  5173  5222  5501  5550  5742
                    5791   6020  6069
MSEL1 = 000010 G    1396#  2562  2622  2906  3071  3193  3258  3387  3667  3947  4219  4358  4420
                    4607   4651  4790  4994  5222  5550  5791  6069  6349
MSGMDA  002665 G    1554#  4569
MSGMDC  002615 G    1547#  4411  4443
MSGMDO  002511 G    1535#  4183  4347  4514  4779  4983  5211  5539  5780  6058
MSGMD1  002553 G    1541#  4269  4380  4640  4684  4814  5018  5246  5574  5815  6093
MSGMP   002354 G    1517#  2943  2956  3109  3138  3230  3318  3423  3509  3703  3789  3983  6385
MSGMPL  002461 G    1530#  3124  3153  3435  3451  3466  3480  3493  3521  3534  3551  3563  3577
                    3591   3605  3715  3731  3746  3760  3773  3801  3814  3831  3843  3858  3871
                    3885   3995  4011  4026  4041  4054  4068  4081  4094  6400
MSGMPS  002415 G    1523#  3298
MSGMSC  002773 G    1566#  5114
MSGMSR  002730 G    1560#  4867  4881  5053  5299  5352  5393  5407  5447  5627  5683  5859  5877
                    5894   5912  5927  5945  5960  6156  6185  6215  6258  6288
MSGTM0  003415 G    1621#  1773
MSGTM2  003471 G    1629#  1785
MSGTM4  003545 G    1637#  1797
MSG1    021446      6576   6596#
MSG2    021462      6581   6598#
MSRODT  014036      4507   4562  4582#
MSR1DT  014302      4633   4677  4697#
MUTBH = 000010 G    1435#  2937  3132  3311  3416  3503  3696  3783  3976  6378
ONEFIL= 000001         1#  1091#  1094  1275#  1277  2146#  2148  2349  2350#  2352  6548  6549#  6551
O$APTS= 000000      1093#  1137
O$AU  = 000000      1093#  1169
O$BGNR= 000000      1093#  1163
O$BGNS= 000000      1093#  1129
O$DU  = 000000      1093#  1171
O$ERRT= 000000      1093#  1179
O$GNSW= 000000      1093#  1133
O$POIN= 000001      1093#  1107#  1195
O$SETU= 000001      1093#  1107#  1123  6630
PNT   = 001000 G    1349#
PRI   = 002000 G    1350#
PRI00 = 000000 G    1338#
PRI01 = 000040 G    1337#
PRI02 = 000100 G    1336#
PRI03 = 000140 G    1335#
PRI04 = 000200 G    1334#
PRI05 = 000240 G    1333#
PRI06 = 000300 G    1332#
PRI07 = 000340 G    1148   1331#  1928  2026  2069  2244  6413  6493
PRNTAL  004240 G    1739   1752  1765  1806#
RDEH  = 000004 G    1434#  2937  3103  3311  3416  3503  3783
RDVH  = 000020 G    1377#  1938  1991  2411  2429  2457  2475  2510  2513  2847  3043  3359  3639
                    3739   3919  4019  6393
READR0  005360 G    2108#  3430  3516  3710  3796  3990  6451  6531
READR2  005426 G    2119#  3119  3148  3446  3488  3529  3558  3600  3726  3768  3809  3838  3880
                    4006   4049  4089
READR4  005466 G    2129#
READR6  005520 G    2138#  3293  4406  4438  4564  4679  5109  5347  5388  5442  5678  5889  5922
                    5955   6210  6283
REGO    002234 G    1455#  1940* 1941  2107* 2108  2231  2285* 6419  6499
```

```
REGOEQ  003204 G  1596#  1807
REG2    002236 G  1456#  2035*  2036  2118*  2119  2287*
REG2EQ  003216 G  1598#  1824
REG4    002240 G  1457#  2076*  2077  2128*  2129  2288*
REG4EQ  003230 G  1600#  1841
REG6    002242 G  1458#  2137*  2138  5839
REG6EQ  003242 G  1602#  1855
RSTH  = 000001 G  1382#  1990  2007  2410  2456  2511  2846  3042  3057  3358  3373  3572  3585
                  3638  3653  3852  3865  3918  3933  4062  4075  6326  6336  6474
ROBAD   002262 G  1468#  1669  1813  1942*  1943*  1944  2109*  2110*  2111
ROEROR  003622 G  1661#  1950  1980  1998  2014  2418  2436  2464  2482  2520  2854  3050  3064
                  3366  3380  3646  3660  3926  3940  6129  6170  6200  6224  6273  6333  6343
                  6445  6457  6481  6525  6537
ROGOOD  002254 G  1465#  1670  1814  1939*  1944  1973*  2106*  2111  3459*  3473*  3545*  3739*  3753*
                  3825*  4019*  4034*  6393*  6450*  6530*
ROLOAD  002252 G  1464#  1673  1817  1937*  1939  1940  1972*  1989*  1990*  2007*  2106  2107  2410*
                  2428*  2456*  2474*  2507*  2511  2525*  2526  2528  2530*  2846*  3042*  3057*  3357*
                  3358*  3373*  3460*  3474*  3544*  3572*  3585*  3637*  3638*  3653*  3740*  3754*  3824*
                  3852*  3865*  3917*  3918*  3933*  4020*  4035*  4062*  4075*  6122*  6163*  6193*  6222*
                  6266*  6326*  6336*  6394*  6438*  6474*  6518*
ROMASK  002256 G  1466#  1671  1815  1938*  1943  1991*  2110  2411*  2429*  2457*  2475*  2510*  2513*
                  2847*  3043*  3559*  3639*  3919*
ROREAD  002260 G  1467#  1672  1816  1941*  1942  2108*  2109
ROTM    004152 G  1771#  1960
R2BAD   002274 G  1474#  1697  1830  2037*  2038*  2039  2120*  2121*  2122
R2EROR  003710 G  1689#  2045  2569  2586  2612  2629  2663  2869  2915  3080  3202  3267  3395
                  3675  3955  4139  4239  4318  4365  4395  4427  4482  4539  4624  4667  4749
                  4797  4839  4952  5001  5037  5087  5181  5229  5271  5509  5557  5599  5750
                  5798  5834  6028  6076  6112  6357
R2GOOD  002266 G  1471#  1698  1831  2034*  2039  2117*  2122  3118*  3146*  3147*  3445*  3487*  3528*
                  3599*  3725*  3767*  3808*  3879*  4005*  4048*  4088*
R2LOAD  002264 G  1470#  1701  1834  2032*  2034  2035  2117  2118  2562*  2579*  2605*  2622*  2653*
                  2667*  2861*  2906*  2907*  3071*  3072*  3193*  3194*  3258*  3259*  3387*  3667*  3947*
                  4131*  4231*  4310*  4358*  4388*  4420*  4474*  4532*  4616*  4660*  4741*  4790*  4832*
                  4944*  4994*  5030*  5080*  5173*  5222*  5264*  5501*  5550*  5592*  5742*  5791*  5827*
                  6020*  6069*  6105*  6349*
R2MASK  002270 G  1472#  1699  1832  2033*  2038  2121  2551*  2654*  2862*  2908*  3073*  3117*  3195*
                  3260*  3388*  3444*  3668*  3724*  3948*  4004*  4132*  4232*  4311*  4475*  4617*  4742*
                  4945*  5174*  5502*  5743*  6021*  6350*
R2READ  002272 G  1473#  1700  1833  2036*  2037  2119*  2120
R2TM    004174 G  1783#  2055
R4EROR  003776 G  1712#  2084  2698  2714  2744  2760  2786  2815  2928  3093  3215  3280  3408
                  3688  3968  4153  4332  4497  4552  4762  4851  4966  5099  5194  5283  5337
                  5379  5432  5522  5611  5668  5763  6041  6142  6243  6370
R4LOAD  002276 G  1476#  1721  1848  2075*  2076  2078  2128  2130  2691*  2707*  2737*  2753*  2777*
                  2790*  2791  2806*  2819*  2921*  3086*  3208*  3273*  3401*  3681*  3961*  4146*  4325*
                  4490*  4545*  4729*  4889*  4931*  5060*  5066*  5121*  5161*  5292  5307*  5323*  5344
                  5358*  5366*  5386  5400  5413*  5419*  5439  5454*  5488*  5637*  5653*  5690*  5730*
                  5967*  6008*  6135*  6236*  6291*  6297*  6363*
R4READ  002300 G  1477#  1720  1847  2077*  2078  2129*  2130
R4TM    004216 G  1795#  2094
R6BAD   002312 G  1483#  1861  2139*  2140*  2141  5853*  5854  5871*  5872  5906*  5907  5939*  5940
R6GOOD  002304 G  1480#  1862  2136*  2141  3292*  4177*  4263*  4405*  4437*  4563*  4678*  4808*  5012*
                  5107*  5240*  5345*  5387*  5441*  5568*  5676*  5809*  5849*  5854  5867*  5869  5872
                  5888*  5902*  5907  5921*  5935*  5940  5954*  6087*  6178*  6208*  6251*  6281*
R6LOAD  002302 G  1479#  1865  2136  2137  2937*  2950*  3103*  3132*  3224*  3291*  3311*  3416*  3503*
```

```
                    3696* 3783* 3976* 4164* 4250* 4341* 4374* 4404* 4405  4436* 4437  4507* 4562*
                    4563  4633* 4677* 4678  4772* 4807* 4860* 4875* 4976* 5011* 5046* 5106* 5107
                    5204* 5239* 5292* 5344* 5345  5386* 5387  5400* 5401* 5439* 5440* 5441  5532*
                    5567* 5620* 5675* 5676  5773* 5808* 5848* 5849  5851  5866* 5867  5886* 5887*
                    5888  5901* 5902  5904  5919* 5920* 5921  5934* 5935  5937  5952* 5953* 5954
                    6051* 6086* 6150* 6177* 6207* 6208  6250* 6280* 6281  6378*
R6MASK  002306 G    1481# 1863  2140  2936* 3102* 3223* 3290* 3312* 3417* 3697* 3977* 4163* 4249*
                    4340* 4508* 4634* 4773* 4861* 4977* 5047* 5108* 5205* 5293* 5346* 5533* 5621*
                    5677* 5774* 5847* 6052* 6149* 6179* 6209* 6252* 6282* 6379*
R6READ  002310 G    1482# 1864  2138* 2139  5850* 5852* 5853  5868* 5870* 5871  5903* 5905* 5906
                    5936* 5938* 5939
SFPTBL  002224 G    1268#
SIG10H= 002000 G    1369#
SIG11H= 004000 G    1368#
SIG8H = 000400 G    1371# 2242
SIG9H = 001000 G    1370#
SVCGBL= 000000      1093# 1110  1111  1119  1120  1121  1122  1123  1124  1125  1126  1127  1128
                    1129  1130  1131  1132  1133  1134  1135  1136  1137  1138  1139  1140  1141
                    1142  1143  1144  1145  1146  1147  1148  1149  1150  1151  1152  1153  1154
                    1156  1157  1159  1160  1161  1162  1163  1164  1165  1166  1167  1168  1169
                    1170  1171  1172  1173  1174  1175  1176  1177  1178  1179  1180  1181  1182
                    1183  1184  1185  1186  1187  1188  1189  1190  1191  1192  1193  1194  1205
                    1206  1246  1247  1248  1267  1268  1269  1445  1446  1496  1497  1505  1506
                    1661  1662  1689  1690  1712  1713  1732  1733  1745  1746  1758  1759  1771
                    1772  1783  1784  1795  1796  2157  2158  2179  2180  2195  2196  2269  2270
                    2284  2285  2310  2311  2333  2334  6565  6566  6616  6617  6632# 6633
SVCINS= 000000      1093# 1111  1112  1113  1114  1115  1116  1117  1118  1119  1120  1121  1122
                    1123  1124  1125  1126  1127  1128  1129  1130  1131  1132  1133  1134  1135
                    1136  1137  1138  1139  1140  1141  1142  1143  1144  1145  1146  1147  1148
                    1149  1150  1151  1152  1153  1154  1155  1156  1157  1158  1159  1160  1161
                    1162  1163  1164  1165  1166  1167  1168  1169  1170  1171  1172  1173  1174
                    1175  1176  1177  1178  1179  1180  1181  1182  1183  1184  1185  1186  1187
                    1188  1189  1190  1191  1192  1193  1194  1195  1204  1205  1206  1207  1208
                    1209  1210  1211  1212  1213  1214  1215  1216  1217  1218  1219  1220  1221
                    1222  1223  1224  1225  1226  1227  1228  1229  1230  1231  1232  1233  1234
                    1245  1246  1266  1267  1497  1499  1500  1506  1510  1511  1663  1664  1665
                    1666  1667  1668  1669  1670  1671  1672  1673  1674  1675  1676  1677  1678
                    1679  1686  1687  1691  1692  1693  1694  1695  1696  1697  1698  1699  1700
                    1701  1702  1703  1704  1705  1706  1707  1709  1710  1714  1715  1716  1717
                    1718  1719  1720  1721  1722  1723  1724  1725  1726  1727  1729  1730  1734
                    1735  1736  1737  1738  1739  1742  1743  1747  1748  1749  1750  1751  1752
                    1755  1756  1760  1761  1762  1763  1764  1765  1768  1769  1773  1774  1775
                    1776  1777  1778  1780  1781  1785  1786  1787  1788  1789  1790  1792  1793
                    1797  1798  1799  1800  1801  1802  1804  1805  1807  1808  1809  1810  1811
                    1812  1813  1814  1815  1816  1817  1818  1819  1820  1821  1822  1823  1824
                    1825  1826  1827  1828  1829  1830  1831  1832  1833  1834  1835  1836  1837
                    1838  1839  1840  1841  1842  1843  1844  1845  1846  1847  1848  1849  1850
                    1851  1852  1853  1854  1855  1856  1857  1858  1859  1860  1861  1862  1863
                    1864  1865  1866  1867  1868  1869  1870  1871  1926  1927  1928  1929  1930
                    1931  1932  1933  1934  1947  1948  1949  1950  1951  1952  1953  1957  1958
                    1959  1960  1961  1962  1963  1964  1966  1967  1971  1972  1977  1978  1979
                    1980  1981  1983  1984  1988  1989  1995  1996  1997  1998  1999  2001  2002
                    2006  2007  2011  2012  2013  2014  2015  2017  2018  2024  2025  2026  2027
                    2028  2029  2030  2031  2032  2042  2043  2044  2045  2046  2047  2048  2052
                    2053  2054  2055  2056  2057  2058  2059  2061  2062  2067  2068  2069  2070
                    2071  2072  2073  2074  2075  2081  2082  2083  2084  2085  2086  2087  2091
```

2092 2093 2094 2095 2096 2097 2098 2100 2101 2161 2162 2163 2169
2170 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208
2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2220 2221 2222
2223 2224 2226 2227 2228 2229 2230 2231 2244 2245 2246 2249 2250
2251 2257 2258 2274 2275 2292 2293 2294 2300 2301 2314 2315 2316
2322 2323 2337 2338 2339 2345 2346 2383 2384 2403 2404 2409 2410
2415 2416 2417 2418 2419 2421 2422 2427 2428 2433 2434 2435 2436
2437 2439 2440 2443 2444 2447 2448 2455 2456 2461 2462 2463 2464
2465 2467 2468 2473 2474 2479 2480 2481 2482 2483 2485 2486 2489
2490 2493 2494 2509 2510 2517 2518 2519 2520 2521 2523 2524 2534
2535 2555 2556 2561 2562 2566 2567 2568 2569 2570 2572 2573 2578
2579 2583 2584 2585 2586 2587 2589 2590 2592 2595 2596 2597 2604
2605 2609 2610 2611 2612 2613 2615 2616 2621 2622 2626 2627 2628
2629 2630 2632 2633 2635 2636 2638 2639 2656 2657 2660 2661 2662
2663 2664 2666 2667 2672 2673 2690 2691 2695 2696 2697 2698 2699
2701 2702 2706 2707 2711 2712 2713 2714 2715 2717 2718 2720 2721
2736 2737 2741 2742 2743 2744 2745 2747 2748 2752 2753 2757 2758
2759 2760 2761 2763 2764 2766 2767 2779 2780 2783 2784 2785 2786
2787 2789 2790 2795 2796 2808 2809 2812 2813 2814 2815 2816 2818
2819 2823 2824 2841 2842 2851 2852 2853 2854 2855 2856 2857 2866
2867 2868 2869 2870 2872 2873 2875 2876 2901 2902 2912 2913 2914
2915 2916 2917 2918 2925 2926 2927 2928 2929 2930 2931 2941 2942
2943 2944 2945 2946 2947 2954 2955 2956 2957 2958 2960 2961 2969
2970 3033 3034 3047 3048 3049 3050 3051 3052 3053 3061 3062 3063
3064 3065 3066 3067 3077 3078 3079 3080 3081 3082 3083 3090 3091
3092 3093 3094 3095 3096 3107 3108 3109 3110 3111 3112 3113 3122
3123 3124 3125 3126 3127 3128 3136 3137 3138 3139 3140 3141 3142
3151 3152 3153 3154 3155 3157 3158 3166 3167 3185 3186 3189 3190
3199 3200 3201 3202 3203 3204 3205 3212 3213 3214 3215 3216 3217
3218 3228 3229 3230 3231 3232 3234 3235 3242 3243 3245 3246 3254
3255 3264 3265 3266 3267 3268 3269 3270 3277 3278 3279 3280 3281
3282 3283 3296 3297 3298 3299 3300 3302 3303 3305 3306 3316 3317
3318 3319 3320 3322 3323 3330 3331 3333 3334 3349 3350 3363 3364
3365 3366 3367 3368 3369 3377 3378 3379 3380 3381 3382 3383 3392
3393 3394 3395 3396 3397 3398 3405 3406 3407 3408 3409 3410 3411
3421 3422 3423 3424 3425 3426 3427 3433 3434 3435 3436 3437 3438
3439 3449 3450 3451 3452 3453 3454 3455 3464 3465 3466 3467 3468
3469 3470 3478 3479 3480 3481 3482 3483 3484 3491 3492 3493 3494
3495 3496 3497 3507 3508 3509 3510 3511 3512 3513 3519 3520 3521
3522 3523 3524 3525 3532 3533 3534 3535 3536 3537 3538 3549 3550
3551 3552 3553 3554 3555 3561 3562 3563 3564 3565 3566 3567 3576
3577 3578 3579 3580 3581 3582 3589 3590 3591 3592 3593 3594 3595
3603 3604 3605 3606 3607 3609 3610 3612 3613 3629 3630 3643 3644
3645 3646 3647 3648 3649 3657 3658 3659 3660 3661 3662 3663 3672
3673 3674 3675 3676 3677 3678 3685 3686 3687 3688 3689 3690 3691
3701 3702 3703 3704 3705 3706 3707 3713 3714 3715 3716 3717 3718
3719 3729 3730 3731 3732 3733 3734 3735 3744 3745 3746 3747 3748
3749 3750 3758 3759 3760 3761 3762 3763 3764 3771 3772 3773 3774
3775 3776 3777 3787 3788 3789 3790 3791 3792 3793 3799 3800 3801
3802 3803 3804 3805 3812 3813 3814 3815 3816 3817 3818 3829 3830
3831 3832 3833 3834 3835 3841 3842 3843 3844 3845 3846 3847 3856
3857 3858 3859 3860 3861 3862 3869 3870 3871 3872 3873 3874 3875
3883 3884 3885 3886 3887 3889 3890 3892 3893 3908 3909 3923 3924
3925 3926 3927 3928 3929 3937 3938 3939 3940 3941 3942 3943 3952
3953 3954 3955 3956 3957 3958 3965 3966 3967 3968 3969 3970 3971
3981 3982 3983 3984 3985 3986 3987 3993 3994 3995 3996 3997 3998

3999	4009	4010	4011	4012	4013	4014	4015	4024	4025	4026	4027	4028
4029	4030	4039	4040	4041	4042	4043	4044	4045	4052	4053	4054	4055
4056	4057	4058	4066	4067	4068	4069	4070	4071	4072	4079	4080	4081
4082	4083	4084	4085	4092	4093	4094	4095	4096	4098	4099	4101	4102
4124	4125	4136	4137	4138	4139	4140	4141	4142	4150	4151	4152	4153
4154	4155	4156	4181	4182	4183	4184	4185	4188	4189	4198	4199	4223
4224	4236	4237	4238	4239	4240	4241	4242	4267	4268	4269	4270	4271
4273	4274	4283	4284	4302	4303	4315	4316	4317	4318	4319	4320	4321
4329	4330	4331	4332	4333	4334	4335	4345	4346	4347	4348	4349	4350
4351	4362	4363	4364	4365	4366	4367	4368	4378	4379	4380	4381	4382
4383	4384	4392	4393	4394	4395	4396	4397	4398	4409	4410	4411	4412
4413	4414	4415	4424	4425	4426	4427	4428	4429	4430	4441	4442	4443
4444	4445	4447	4448	4450	4451	4469	4470	4479	4480	4481	4482	4483
4484	4485	4494	4495	4496	4497	4498	4499	4500	4512	4513	4514	4515
4516	4518	4519	4527	4528	4536	4537	4538	4539	4540	4541	4542	4549
4550	4551	4552	4553	4554	4555	4567	4568	4569	4570	4571	4573	4574
4579	4580	4581	4593	4594	4611	4612	4621	4622	4623	4624	4625	4626
4627	4638	4639	4640	4641	4642	4644	4645	4655	4656	4664	4665	4666
4667	4668	4669	4670	4682	4683	4684	4685	4686	4688	4689	4695	4696
4697	4703	4704	4733	4734	4746	4747	4748	4749	4750	4751	4752	4759
4760	4761	4762	4763	4764	4765	4777	4778	4779	4780	4781	4782	4783
4794	4795	4796	4797	4798	4799	4800	4812	4813	4814	4815	4816	4818
4819	4823	4824	4836	4837	4838	4839	4840	4841	4842	4848	4849	4850
4851	4852	4853	4854	4865	4866	4867	4868	4869	4870	4871	4879	4880
4881	4882	4883	4885	4886	4903	4904	4936	4937	4949	4950	4951	4952
4953	4954	4955	4963	4964	4965	4966	4967	4968	4969	4981	4982	4983
4984	4985	4986	4987	4998	4999	5000	5001	5002	5003	5004	5016	5017
5018	5019	5020	5021	5022	5034	5035	5036	5037	5038	5039	5040	5051
5052	5053	5054	5055	5057	5058	5073	5074	5084	5085	5086	5087	5088
5089	5090	5096	5097	5098	5099	5100	5101	5102	5112	5113	5114	5115
5116	5118	5119	5125	5126	5165	5166	5178	5179	5180	5181	5182	5183
5184	5191	5192	5193	5194	5195	5196	5197	5209	5210	5211	5212	5213
5214	5215	5226	5227	5228	5229	5230	5231	5232	5244	5245	5246	5247
5248	5250	5251	5255	5256	5268	5269	5270	5271	5272	5273	5274	5280
5281	5282	5283	5284	5285	5286	5297	5298	5299	5300	5301	5303	5304
5326	5327	5334	5335	5336	5337	5338	5339	5340	5350	5351	5352	5353
5354	5356	5357	5369	5370	5376	5377	5378	5379	5380	5381	5382	5391
5392	5393	5394	5395	5396	5397	5405	5406	5407	5408	5409	5411	5412
5422	5423	5429	5430	5431	5432	5433	5434	5435	5445	5446	5447	5448
5449	5452	5453	5458	5459	5493	5494	5506	5507	5508	5509	5510	5511
5512	5519	5520	5521	5522	5523	5524	5525	5537	5538	5539	5540	5541
5542	5543	5554	5555	5556	5557	5558	5559	5560	5572	5573	5574	5575
5576	5578	5579	5583	5584	5596	5597	5598	5599	5600	5601	5602	5608
5609	5610	5611	5612	5613	5614	5625	5626	5627	5628	5629	5631	5632
5657	5658	5665	5666	5667	5668	5669	5670	5671	5681	5682	5663	5684
5685	5687	5688	5695	5696	5734	5735	5747	5748	5749	5750	5751	5752
5753	5760	5761	5762	5763	5764	5765	5766	5778	5779	5780	5781	5782
5783	5784	5795	5796	5797	5798	5799	5800	5801	5813	5814	5815	5816
5817	5818	5819	5831	5832	5833	5834	5835	5837	5838	5843	5844	5857
5858	5859	5860	5861	5862	5863	5875	5876	5877	5878	5879	5880	5881
5892	5893	5894	5895	5896	5897	5898	5910	5911	5912	5913	5914	5915
5916	5925	5926	5927	5928	5929	5930	5931	5943	5944	5945	5946	5947
5948	5949	5958	5959	5960	5961	5962	5964	5965	5973	5974	6012	6013
6025	6026	6027	6028	6029	6030	6031	6038	6039	6040	6041	6042	6043
6044	6056	6057	6058	6059	6060	6061	6062	6073	6074	6075	6076	6077
6078	6079	6091	6092	6093	6094	6095	6096	6097	6109	6110	6111	6112

	6113	6115	6116	6118	6119	6126	6127	6128	6129	6130	6131	6132	6139
	6140	6141	6142	6143	6144	6145	6154	6155	6156	6157	6158	6159	6160
	6167	6168	6169	6170	6171	6172	6173	6183	6184	6185	6186	6187	6188
	6189	6197	6198	6199	6200	6201	6202	6203	6213	6214	6215	6216	6217
	6218	6219	6226	6227	6228	6229	6230	6231	6232	6240	6241	6242	6243
	6244	6245	6246	6256	6257	6258	6259	6260	6261	6262	6270	6271	6272
	6273	6274	6275	6276	6286	6287	6288	6289	6290	6294	6295	6302	6303
	6320	6321	6325	6326	6330	6331	6332	6333	6334	6335	6336	6340	6341
	6342	6343	6344	6345	6346	6354	6355	6356	6357	6358	6359	6360	6367
	6368	6369	6370	6371	6372	6373	6383	6384	6385	6386	6387	6388	6389
	6398	6399	6400	6401	6402	6404	6405	6409	6410	6411	6412	6413	6414
	6415	6416	6417	6418	6419	6423	6424	6425	6436	6437	6438	6442	6443
	6444	6445	6446	6447	6448	6454	6455	6456	6457	6458	6460	6461	6463
	6464	6467	6468	6473	6474	6478	6479	6480	6481	6482	6484	6485	6489
	6490	6491	6492	6493	6494	6495	6496	6497	6498	6499	6503	6504	6505
	6516	6517	6518	6522	6523	6524	6525	6526	6527	6528	6534	6535	6536
	6537	6538	6540	6541	6543	6544	6546	6547	6564	6565	6575	6576	6577
	6578	6579	6580	6581	6582	6583	6584	6585	6589	6590	6615	6616	6622
	6623	6629	6630	6631	6632	6637	6638	6639					
SVCSUB- 000000	1093#	2402	2403	2446	2447	2554	2555	2595	2596	3184	3185	3244	3245
	6319	6320	6466	6467									
SVCTAG- 000000	1093#	1254	1255	1272	1273	1685	1686	1708	1709	1728	1729	1741	1742
	1754	1755	1767	1768	1779	1780	1791	1792	1803	1804	1965	1966	1982
	1983	2000	2001	2016	2017	2060	2061	2099	2100	2168	2169	2256	2257
	2273	2274	2299	2300	2321	2322	2344	2345	2382	2383	2420	2421	2438
	2439	2442	2443	2466	2467	2484	2485	2488	2489	2492	2493	2522	2523
	2533	2534	2571	2572	2588	2589	2591	2592	2614	2615	2631	2632	2634
	2635	2637	2638	2665	2666	2671	2672	2700	2701	2716	2717	2719	2720
	2746	2747	2762	2763	2765	2766	2788	2789	2794	2795	2817	2818	2822
	2823	2871	2872	2874	2875	2959	2960	2968	2969	3156	3157	3165	3166
	3233	3234	3241	3242	3301	3302	3321	3322	3329	3330	3332	3333	3608
	3609	3611	3612	3888	3889	3891	3892	4097	4098	4100	4101	4187	4188
	4197	4198	4272	4273	4282	4283	4446	4447	4449	4450	4517	4518	4572
	4573	4592	4593	4643	4644	4687	4688	4702	4703	4817	4818	4884	4885
	4902	4903	5056	5057	5117	5118	5124	5125	5249	5250	5302	5303	5355
	5356	5410	5411	5451	5452	5457	5458	5577	5578	5630	5631	5686	5687
	5694	5695	5836	5837	5963	5964	5972	5973	6114	6115	6293	6294	6301
	6302	6403	6404	6459	6460	6462	6463	6483	6484	6539	6540	6542	6543
	6545	6546	6590	6591	6623	6624	6639	6640	6643	6644			
SVCTST= 000000	1093#	2375	2376	2398	2399	2505	2506	2549	2550	2650	2651	2684	2685
	2730	2731	2775	2776	2804	2805	2838	2839	2895	2896	3027	3028	3181
	3182	3346	3347	3626	3627	3905	3906	4118	4119	4217	4218	4298	4299
	4463	4464	4605	4606	4727	4728	4928	4929	5159	5160	5486	5487	5728
	5729	6006	6007	6315	6316								
S$LSYM- 010000	1093#	1255#	1273#	1686#	1709#	1729#	1742#	1755#	1768#	1780#	1792#	1804#	1926#
	1971#	1988#	2006#	2024#	2067#	2169#	2257#	2274#	2300#	2322#	2345#	2383#	2409#
	2427#	2443#	2455#	2473#	2489#	2493#	2509#	2534#	2561#	2578#	2592#	2604#	2621#
	2635#	2638#	2656#	2672#	2690#	2706#	2720#	2736#	2752#	2766#	2779#	2795#	2808#
	2823#	2841#	2875#	2901#	2969#	3033#	3166#	3189#	3242#	3254#	3305#	3330#	3333#
	3349#	3612#	3629#	3892#	3908#	4101#	4124#	4198#	4223#	4283#	4302#	4450#	4469#
	4527#	4593#	4611#	4655#	4703#	4733#	4823#	4903#	4936#	5073#	5125#	5165#	5255#
	5326#	5369#	5422#	5458#	5493#	5583#	5657#	5695#	5734#	5843#	5973#	6012#	6118#
	6302#	6325#	6409#	6463#	6473#	6489#	6543#	6546#	6591#	6624#			
T$ARG(000006	1111#	1112#	1113#	1114#	1115#	1116#	1663#	1667	1669#	1678	1691#	1695	1697#
	1706	1714#	1718	1720#	1726	1734#	1738	1747#	1751	1760#	1764	1773#	1777
	1785#	1789	1797#	1801	1807#	1811	1813#	1822	1824#	1828	1830#	1839	1841#

```
                    1845    1847#   1853    1855#   1859    1861#   1870
T$CODE= 001032      6575#   6580#
T$ERRN= 000001      1093#   1948#   1958#   1978#   1996#   2012#   2043#   2053#   2082#   2092#   2416#   2434#   2462#
                    2480#   2518#   2567#   2584#   2610#   2627#   2661#   2696#   2712#   2742#   2758#   2784#   2813#
                    2852#   2867#   2913#   2926#   2942#   2955#   3048#   3062#   3078#   3091#   3108#   3123#   3137#
                    3152#   3200#   3213#   3229#   3265#   3278#   3297#   3317#   3364#   3378#   3393#   3406#   3422#
                    3434#   3450#   3465#   3479#   3492#   3508#   3520#   3533#   3550#   3562#   3577#   3590#   3604#
                    3644#   3658#   3673#   3686#   3702#   3714#   3730#   3745#   3759#   3772#   3788#   3800#   3813#
                    3830#   3842#   3857#   3870#   3884#   3924#   3938#   3953#   3966#   3982#   3994#   4010#   4025#
                    4040#   4053#   4067#   4080#   4093#   4137#   4151#   4182#   4237#   4268#   4316#   4330#   4346#
                    4363#   4379#   4393#   4410#   4425#   4442#   4480#   4495#   4513#   4537#   4550#   4568#   4622#
                    4639#   4665#   4683#   4747#   4760#   4778#   4795#   4813#   4837#   4849#   4866#   4880#   4950#
                    4964#   4982#   4999#   5017#   5035#   5052#   5085#   5097#   5113#   5179#   5192#   5210#   5227#
                    5245#   5269#   5281#   5298#   5335#   5351#   5377#   5392#   5406#   5430#   5446#   5507#   5520#
                    5538#   5555#   5573#   5597#   5609#   5626#   5666#   5682#   5748#   5761#   5779#   5796#   5814#
                    5832#   5858#   5876#   5893#   5911#   5926#   5944#   5959#   6026#   6039#   6057#   6074#   6092#
                    6110#   6127#   6140#   6155#   6168#   6184#   6198#   6214#   6227#   6241#   6257#   6271#   6287#
                    6331#   6341#   6355#   6368#   6384#   6399#   6443#   6455#   6479#   6523#   6535#
T$EXCP= 000000      6575#   6579    6580#   6585
T$FLAG= 000040      2161#   2163    2249#   2292#   2314#   2316    2337#   2339    4579#   4695#
T$FREE= 021536      6630    6645#
I$GMAN= 000000      1093#
T$HILI= 000017      6575#   6578    6580#   6584
T$LAST= 000001      1093#   6630#   6636
T$LOLI= 000000      6575#   6577    6580#   6583
T$LSYM= 010000      1093#   1255    1273    1686    1709    1729    1742    1755    1768    1780    1792    1804    2169
                    2257    2274    2300    2322    2345    2383    2443    2489    2493    2534    2592    2635    2638
                    2672    2720    2766    2795    2823    2875    2969    3166    3242    3330    3333    3612    3892
                    4101    4198    4283    4450    4593    4703    4903    5125    5458    5695    5973    6302    6463
                    6543    6546    6591    6624
T$LTNO= 000034      6633#
T$NEST= 177777      1093#   1100#   1245#   1254#   1266#   1272#   1275#   1280#   1661#   1685#   1689#   1708#   1712#
                    1728#   1732#   1741#   1745#   1754#   1758#   1767#   1771#   1779#   1783#   1791#   1795#   1803#
                    1926#   1965#   1971#   1982#   1988#   2000#   2006#   2016#   2024#   2060#   2067#   2099#   2145#
                    2150#   2157#   2168#   2179#   2186#   2195#   2256#   2269#   2273#   2284#   2299#   2310#   2321#
                    2333#   2344#   2348#   2354#   2376#   2382#   2399#   2403#   2409#   2420#   2427#   2438#   2442#
                    2447#   2455#   2466#   2473#   2484#   2488#   2492#   2506#   2509#   2522#   2533#   2550#   2555#
                    2561#   2571#   2578#   2588#   2591#   2596#   2604#   2614#   2621#   2631#   2634#   2637#   2651#
                    2656#   2665#   2671#   2685#   2690#   2700#   2706#   2716#   2719#   2731#   2736#   2746#   2752#
                    2762#   2765#   2776#   2779#   2788#   2794#   2805#   2808#   2817#   2822#   2839#   2841#   2871#
                    2874#   2896#   2901#   2959#   2968#   3028#   3033#   3156#   3165#   3182#   3185#   3189#   3233#
                    3241#   3245#   3254#   3301#   3305#   3321#   3329#   3332#   3347#   3349#   3608#   3611#   3627#
                    3629#   3888#   3891#   3906#   3908#   4097#   4100#   4119#   4124#   4187#   4197#   4218#   4223#
                    4272#   4282#   4299#   4302#   4446#   4449#   4464#   4469#   4517#   4527#   4572#   4592#   4606#
                    4611#   4643#   4655#   4687#   4702#   4728#   4733#   4817#   4823#   4884#   4902#   4929#   4936#
                    5056#   5073#   5117#   5124#   5160#   5165#   5249#   5255#   5302#   5326#   5355#   5369#   5410#
                    5422#   5451#   5457#   5487#   5493#   5577#   5583#   5630#   5657#   5686#   5694#   5729#   5734#
                    5836#   5843#   5963#   5972#   6007#   6012#   6114#   6118#   6293#   6301#   6316#   6320#   6325#
                    6403#   6409#   6459#   6462#   6467#   6473#   6483#   6489#   6539#   6542#   6545#   6548#   6553#
                    6564#   6589#   6615#   6622#   6634#
T$NS0 = 000000      1100#   1275    1280#   2145    2150#   2348    2354#   6548    6553#   6634
T$NS1 = 000005      1245#   1254    1266#   1272    1661#   1685    1689#   1708    1712#   1728    1732#   1741    1745#
                    1754    1758#   1767    1771#   1779    1783#   1791    1795#   1803    1926#   1965    1971#   1982
                    1988#   2000    2006#   2016    2024#   2060    2067#   2099    2157#   2168    2179#   2186    2195#
                    2256    2269#   2273    2284#   2299    2310#   2321    2333#   2344    2376#   2382    2399#   2492
                    2506#   2533    2550#   2637    2651#   2671    2685#   2719    2731#   2765    2776#   2794    2805#
```

```
                    2822    2839#   2874    2896#   2968    3028#   3165    3182#   3332    3347#   3611    3627#   3891
                    3906#   4100    4119#   4197    4218#   4282    4299#   4449    4464#   4592    4[illegible]   4702    4728#
                    4902    4929#   5124    5160#   5457    5487#   5694    5729#   5972    6007#   6[illegible]   6316#   6545
                    6564#   6589    6615#   6622
T$NS2 = 000002      2403#   2442    2447#   2488    2509#   2522    2555#   2591    2596#   2634    2656#   2665    2690#
                    2700    2706#   2716    2736#   2746    2752#   2762    2779#   2788    2808#   2817    2841#   2871
                    2901#   2959    3033#   3156    3185#   3241    3245#   3329    3349#   3608    3629#   3888    3908#
                    4097    4124#   4187    4223#   4272    4302#   4446    4469#   4517    4527#   4572    4611#   4643
                    4655#   4687    4733#   4817    4823#   4884    4936#   5056    5073#   5117    5165#   5249    5255#
                    5302    5326#   5355    5369#   5410    5422#   5451    5493#   5577    5583#   5630    5657#   5686
                    5734#   5836    5843#   5963    6012#   6114    6118#   6293    6320#   6462    6467#   6542
T$NS3 = 000003      2409#   2420    2427#   2438    2455#   2466    2473#   2484    2561#   257     2578#   2588    2604#
                    2614    2621#   2631    3189#   3233    3254#   3301    3305#   3321    6325#   6403    6409#   6459
                    6473#   6483    6489#   6539
T$PCNT= 000000      6636#   6637#
T$PTAB= 010071      6637#   6640
T$PTHV= 000001      1124    6645#
T$PTNU= 000001      1093#   6640#   6645
T$SAVL= 177777      1093#
T$SEGL= 177777      1093#   1926#   1965#   1967    1971#   1982#   1984    1988#   2000#   2002    2006#   2016#   2018
                    2024#   2060#   2062    2067#   2099#   2101    2409#   2420#   2422    2427#   2438#   2440    2455#
                    2466#   2468    2473#   2484#   2486    2509#   2522#   2524    2561#   2571#   2573    2578#   2588#
                    2590    2604#   2614#   2616    2621#   2631#   2633    2656#   2665#   2667    2690#   2700#   2702
                    2706#   2716#   2718    2736#   2746#   2748    2752#   2762#   2764    2779#   2788#   2790    2808#
                    2817#   2819    2841#   2871#   2873    2901#   2959#   2961    3033#   3156#   3158    3189#   3233#
                    3235    3254#   3301#   3303    3305#   3321#   3323    3349#   3608#   3610    3629#   3888#   3890
                    3908#   4097#   4099    4124#   4187#   4189    4223#   4272#   4274    4302#   4446#   4448    4469#
                    4517#   4519    4527#   4572#   4574    4611#   4643#   4645    4655#   4687#   4689    4733#   4817#
                    4819    4823#   4884#   4886    4936#   5056#   5058    5073#   5117#   5119    5165#   5249#   5251
                    5255#   5302#   5304    5326#   5355#   5357    5369#   5410#   5412    5422#   5451#   5453    5493#
                    5577#   5579    5583#   5630#   5632    5657#   5686#   5688    5734#   5836#   5838    5843#   5963#
                    5965    6012#   6114#   6116    6118#   6293#   6295    6325#   6403#   6405    6409#   6459#   6461
                    6473#   6483#   6485    6489#   6539#   6541
T$SEKO= 010001      1926#   1965    1971#   1982    1988#   2000    2006#   2016    2024#   2060    2067#   2099    2409#
                    2420    2427#   2438    2455#   2466    2473#   2484    2509#   2522    2561#   2571    2578#   2588
                    2604#   2614    2621#   2631    2656#   2665    2690#   2700    2706#   2716    2736#   2746    2752#
                    2762    2779#   2788    2808#   2817    2841#   2871    2901#   2959    3033#   3156    3189#   3233
                    3254#   3301    3305#   3321    3349#   3608    3629#   3888    3908#   4097    4124#   4187    4223#
                    4272    4302#   4446    4469#   4517    4527#   4572    4611#   4643    4655#   4687    4733#   4817
                    4823#   4884    4936#   5056    5073#   5117    5165#   5249    5255#   5302    5326#   5355    5369#
                    5410    5422#   5451    5493#   5577    5583#   5630    5657#   5686    5734#   5836    5843#   5963
                    6012#   6114    6118#   6293    6325#   6403    6409#   6459    6473#   6483    6489#   6539
T$SIZE- 000004      6631    6645#
T$SUBN= 000002      1093#   2375#   2398#   2402#   2446#   2505#   2549#   2554#   2595#   2650#   2684#   2730#   2775#
                    2804#   2838#   2895#   3027#   3181#   3184#   3244#   3346#   3626#   3905#   4118#   4217#   4298#
                    4463#   4605#   4727#   4928#   5159#   5486#   5728#   6006#   6315#   6319#   6466#
T$TAGL= 177777      1093#
T$TAGN= 010073      1093#   1245#   1266#   1661#   1689#   1712#   1732#   1745#   1758#   1771#   1783#   1795#   2157#
                    2179#   2195#   2269#   2284#   2310#   2333#   2376#   2399#   2403#   2447#   2506#   2550#   2555#
                    2596#   2651#   2685#   2731#   2776#   2805#   2839#   2896#   3028#   3182#   3185#   3245#   3347#
                    3627#   3906#   4119#   4218#   4299#   4464#   4606#   4728#   4929#   5160#   5487#   5729#   6007#
                    6316#   6320#   6467#   6564#   6615#   6636#   6637#   6638#
T$TEMP  000000      1206#   1207#   1208#   1209#   1210#   1211#   1212#   1213#   1214#   1215#   1216#   1217#   1218#
                    1219#   1220#   1221#   1222#   1223#   1224#   1225#   1226#   1227#   1228#   1229#   1230#   1231#
                    1232#   1233#   1234#   1254#   1272#   1275#   1685#   1708#   1728#   1741#   1754#   1767#   1779#
                    1791#   1803#   1965#   1982#   2000#   2016#   2060#   2099#   2145#   2161#   2162    2168#   2186#
```

	2249#	2250	2256#	2273#	2292#	2293	2299#	2314#	2315	2321#	2337#	2338	2344#
	2348#	2382#	2420#	2438#	2442#	2466#	2484#	2488#	2492#	2522#	2533#	2571#	2588#
	2591#	2614#	2631#	2634#	2637#	2665#	2671#	2700#	2716#	2719#	2746#	2762#	2765#
	2788#	2794#	2817#	2822#	2871#	2874#	2959#	2968#	3156#	3165#	3233#	3241#	3301#
	3321#	3329#	3332#	3608#	3611#	3888#	3891#	4097#	4100#	4187#	4197#	4272#	4282#
	4446#	4449#	4517#	4572#	4579#	4580	4592#	4643#	4687#	4695#	4696	4702#	4817#
	4884#	4902#	5056#	5117#	5124#	5249#	5302#	5355#	5410#	5451#	5457#	5577#	5630#
	5686#	5694#	5836#	5963#	5972#	6114#	6293#	6301#	6403#	6459#	6462#	6483#	6539#
	6542#	6545#	6548#	6575#	6580#	6589#	6622#	6634#					
T$TEST= 000034	1093#	2375#	2398#	2402	2446	2505#	2549#	2554	2595	2650#	2684#	2730#	2775#
	2804#	2838#	2895#	3027#	3181#	3184	3244	3346#	3626#	3905#	4118#	4217#	4298#
	4463#	4605#	4727#	4928#	5159#	5486#	5728#	6006#	6315#	6319	6466	6633	
T$TSTM= 177777	1093#	1666	1677	1686	1694	1705	1709	1717	1725	1729	1737	1742	1750
	1755	1763	1768	1776	1780	1788	1792	1800	1804	1810	1821	1827	1838
	1844	1852	1858	1869	1926	1932	1947	1952	1957	1963	1966	1971	1977
	1983	1988	1995	2001	2006	2011	2017	2024	2030	2042	2047	2052	2058
	2061	2067	2073	2081	2086	2091	2097	2100	2169	2198	2203	2208	2212
	2215	2221	2227	2245	2249	2257	2274	2292	2300	2322	2345	2383	2403
	2409	2415	2421	2427	2433	2439	2443	2447	2455	2461	2467	2473	2479
	2485	2489	2493	2509	2517	2523	2534	2555	2561	2566	2572	2578	2583
	2589	2592	2596	2604	2609	2615	2621	2626	2632	2635	2638	2656	2660
	2666	2672	2690	2695	2701	2706	2711	2717	2720	2736	2741	2747	2752
	2757	2763	2766	2779	2783	2789	2795	2808	2812	2818	2823	2841	2851
	2856	2866	2872	2875	2901	2912	2917	2925	2930	2941	2946	2954	2960
	2969	3033	3047	3052	3061	3066	3077	3082	3090	3095	3107	3112	3122
	3127	3136	3141	3151	3157	3166	3185	3189	3199	3204	3212	3217	3228
	3234	3242	3245	3254	3264	3269	3277	3282	3296	3302	3305	3316	3322
	3330	3333	3349	3363	3368	3377	3382	3392	3397	3405	3410	3421	3426
	3433	3438	3449	3454	3464	3469	3478	3483	3491	3496	3507	3512	3519
	3524	3532	3537	3549	3554	3561	3566	3576	3581	3589	3594	3603	3609
	3612	3629	3643	3648	3657	3662	3672	3677	3685	3690	3701	3706	3713
	3718	3729	3734	3744	3749	3758	3763	3771	3776	3787	3792	3799	3804
	3812	3817	3829	3834	3841	3846	3856	3861	3869	3874	3883	3889	3892
	3908	3923	3928	3937	3942	3952	3957	3965	3970	3981	3986	3993	3998
	4009	4014	4024	4029	4039	4044	4052	4057	4066	4071	4079	4084	4092
	4098	4101	4124	4136	4141	4150	4155	4181	4188	4198	4223	4236	4241
	4267	4273	4283	4302	4315	4320	4329	4334	4345	4350	4362	4367	4378
	4383	4392	4397	4409	4414	4424	4429	4441	4447	4450	4469	4479	4484
	4494	4499	4512	4518	4527	4536	4541	4549	4554	4567	4573	4579	4593
	4611	4621	4626	4638	4644	4655	4664	4669	4682	4688	4695	4703	4733
	4746	4751	4759	4764	4777	4782	4794	4799	4812	4818	4823	4836	4841
	4848	4853	4865	4870	4879	4885	4903	4936	4949	4954	4963	4968	4981
	4986	4998	5003	5016	5021	5034	5039	5051	5057	5073	5084	5089	5096
	5101	5112	5118	5125	5165	5178	5183	5191	5196	5209	5214	5226	5231
	5244	5250	5255	5268	5273	5280	5285	5297	5303	5326	5334	5339	5350
	5356	5369	5376	5381	5391	5396	5405	5411	5422	5429	5434	5445	5452
	5458	5493	5506	5511	5519	5524	5537	5542	5554	5559	5572	5578	5583
	5596	5601	5608	5613	5625	5631	5657	5665	5670	5681	5687	5695	5734
	5747	5752	5760	5765	5778	5783	5795	5800	5813	5818	5831	5837	5843
	5857	5862	5875	5880	5892	5897	5910	5915	5925	5930	5943	5948	5958
	5964	5973	6012	6025	6030	6038	6043	6056	6061	6073	6078	6091	6096
	6109	6115	6118	6126	6131	6139	6144	6154	6159	6167	6172	6183	6188
	6197	6202	6213	6218	6226	6231	6240	6245	6256	6261	6270	6275	6286
	6294	6302	6320	6325	6330	6335	6340	6345	6354	6359	6367	6372	6383
	6388	6398	6404	6409	6411	6417	6424	6437	6442	6447	6454	6460	6463
	6467	6473	6478	6484	6489	6491	6497	6504	6517	6522	6527	6534	6540

```
                    6543    6546
T$TSTS= 000001      1093#   2376#   2399#   2506#   2550#   2651#   2685#   2731#   2776#   2805#   2839#   2896#   3028#
                    3182#   3347#   3627#   3906#   4119#   4218#   4299#   4464#   4606#   4728#   4929#   5160#   5487#
                    5729#   6007#   6316#
T$$AU = 010021      2333#   2337    2344
T$$AUT= 010016      2269#   2273
T$$CLE= 010017      2284#   2292    2299
T$$DAT= 010072      6638#   6643
T$$DU = 010020      2310#   2314    2321
T$$HAR= 010066      6564#   6590
T$$HW = 010000      1245#   1254
T$$INI= 010015      2195#   2249    2256
T$$MSG= 010012      1661#   1685    1689#   1708    1712#   1728    1732#   1741    1745#   1754    1758#   1767    1771#
                    1779    1783#   1791    1795#   1803
T$$PC = 000001      6636#   6645
T$$PRO= 010014      2179#
T$$PTA= 010071      6636#   6639    6640#
T$$RPT= 010013      2157#   2161    2168
T$$SEG= 010001      1926#   1965#   1971#   1982#   1988#   2000#   2006#   2016#   2024#   2060#   2067#   2099#   2409#
                    2420#   2427#   2438#   2455#   2466#   2473#   2484#   2509#   2522#   2561#   2571#   2578#   2588#
                    2604#   2614#   2621#   2631#   2656#   2665#   2690#   2700#   2706#   2716#   2736#   2746#   2752#
                    2762#   2779#   2788#   2808#   2817#   2841#   2871#   2901#   2959#   3033#   3156#   3189#   3233#
                    3254#   3301#   3305#   3321#   3349#   3608#   3629#   3888#   3908#   4097#   4124#   4187#   4223#
                    4272#   4302#   4446#   4469#   4517#   4527#   4572#   4611#   4643#   4655#   4687#   4733#   4817#
                    4823#   4884#   4936#   5056#   5073#   5117#   5165#   5249#   5255#   5302#   5326#   5355#   5369#
                    5410#   5422#   5451#   5493#   5577#   5583#   5630#   5657#   5686#   5734#   5836#   5843#   5963#
                    6012#   6114#   6118#   6293#   6325#   6403#   6409#   6459#   6473#   6483#   6489#   6539#
T$$SOF= 010067      6615#   6623
T$$SUB= 010065      2403#   2442    2447#   2488    2555#   2591    2596#   2634    3185#   3241    3245#   3329    6320#
                    6462    6467#   6542
T$$SW = 010001      1266#   1272
T$$TES= 010063      2376#   2382    2399#   2492    2506#   2533    2550#   2637    2651#   2671    2685#   2719    2731#
                    2765    2776#   2794    2805#   2822    2839#   2874    2876#   2968    3028#   3165    3182#   3332
                    3347#   3611    3627#   3891    3906#   4100    4119#   4217    4218#   4282    4299#   4449    4464#
                    4579    4592    4606#   4695    4702    4728#   4902    4929#   5124    5160#   5457    5487#   5694
                    5729#   5972    6007#   6301    6316#   6545
T1      006012 G    1206    2375#
T10     007066 G    1215    2838#
T11     007164 G    1216    2895#
T12     007364 G    1217    3027#
T13     007740 G    1218    3181#
T13.1   007744      3184#
T13.2   010114      3244#
T14     010330 G    1219    3346#
T15     011206 G    1220    3626#
T16     012064 G    1221    3905#
T17     012572 G    1222    4118#
T18     013010 G    1223    4217#
T19     013204 G    1224    4298#
T2      006020 G    1207    2398#
T2.1    006024      2402#
T2.2    006122      2446#
T20     013550 G    1225    4463#
T21     014060 G    1226    4605#
T22     014314 G    1227    4727#
T23     014704 G    1228    4928#
```

```
T24      015352 G          1229    5159#
T25      016234 G          1230    5486#
T26      016706 G          1231    5728#
T27      017650 G          1232    6006#
T28      020622 G          1233    6315#
T28.1    020626            6319#
T28.2    021222            6466#
T3       006224 G          1208    2505#
T4       006340 G          1209    2549#
T4.1     006352            2554#
T4.2     006434            2595#
T5       006522 G          1210    2650#
T6       006600 G          1211    2684#
T7       006664 G          1212    2730#
T8       006752 G          1213    2775#
T9       007022 G          1214    2804#
UAM   =  000200 G          1347#
UNITNB   002246 G          1461#   2218*   2224*   2226
WREH  =  000002 G          1433#   2937    3132    3311    3503    3696    3783
WRENH =  000100 G          1391#   3117    3146    3147    3444    3528    3724    3725    3767    3808    4004
WRVH  =  000040 G          1376#   1938    1991    2411    2429    2457    2475    2510    2513    2847    3043    3359    3459
                           3639    3919    4019    6393
X$ALWA=  000000            1093#
X$FALS=  000040            1093#
X$OFFS=  000400            1093#
X$TRUE=  000020            1093#
$PATCH   021502 G          6625#
.     -  021536            1097#   1499#   1510#   1647#   2162    2250    2293    2315    2338    4580    4696    6626#   6638
                           6645

. ABS.   021536      000

 ERROPS DETECTED:  0

 CVCDAA.BIC,CVCDAA/CRF:SYM/SOL/NL:TOC=SVC/ML,CVCDAA.P11
 RUN-TIME: 46 47 3 SECONDS
 RUN-TIME RATIO: 940/97=9.5
 CORE USED:  18K  (36 PAGES)
```