

MDE/T-11

MEM SIM DIAG
CVCDABO

AH-T002B-MC
FICHE 1 OF 1

SEP 1982
COPYRIGHT © 81-82
MADE IN USA



Grid of 10 columns and 20 rows of data, likely representing a memory simulation or diagnostic output. The data is extremely faint and illegible.

.REM 8

IDENTIFICATION

PRODUCT CODE: AC-T000B-MC
PRODUCT NAME: CVC DABC MDE/T-11 MEM SIM DIAG
PRODUCT DATE: OCTOBER 1982
MAINTAINER: DIAGNOSTIC ENGINEERING

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

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

COPYRIGHT (C) 1981,1982 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

REVISION HISTORY

REVISION

DATE

REASONS

A
B

SEPTEMBER 1981
OCTOBER 1982

FIRST RELEASE
DOCUMENTATION CHANGE

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	COMMANDS
2.2	SWITCHES
2.3	FLAGS
2.4	HARDWARE QUESTIONS
2.5	SOFTWARE QUESTIONS
2.6	EXTENDED P-TABLE DIALOGUE
2.7	QUICK STARTUP PROCEDURE
3.0	ERROR INFORMATION
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THE MDE/T-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. MDE/T-11 BACKPLANE AND CABLES
5. MEMORY SIMULATOR MODULE(S) (M8740)
6. MXV11 MODULE AND MDE/T-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 (CHOUS).

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 'DDDD'.

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:DDDD	EXECUTE DDDDD PASSES (DDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED IN SECTION 2.3.
/EOP:DDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDD PASSES ONLY. (DDDD = 1 TO 64000)
/UNITS:LIST	TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED IN THE LIST. LIST EXAMPLE - /UNITS:0:5:10-12 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

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

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

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

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

2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
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'

xxxxxx: BITS CLEARED
SIX OCTAL DIGITS INDICATING THE DATA 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 **

CVCD A 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.

CVCD A 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
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: xx, 'xx
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.

CVDA 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 **

CVDA 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 **

CVDA 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 THE 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 ZEROES. 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 REGISTER 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 ERROR 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 DO 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 SIMULATOR RAM

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.

```

1092          .TITLE PROGRAM HEADER AND TABLES
1093          .SBTTL PROGRAM HEADER
1094
1095          .ENABL ABS
1096          .ENABL AMA
1097          .DSABL GBL
1098          .      =      2000
1099          002000
1100 002000          BGNMOD
1101
1102          :++
1103          : THE PROGRAM HEADER IS THE INTERFACE BETWEEN
1104          : THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
1105          :--
1106
1107 002000          POINTER BGNSETUP
1108
1109
1110 002000          HEADER CVCDA,B,0,60.,0,PRI07
1111 002000          LSNAME::          :DIAGNOSTIC NAME
1112 002000          103          .ASCII /C/
1113 002001          125          .ASCII /V/
1114 002002          103          .ASCII /C/
1115 002003          104          .ASCII /D/
1116 002004          101          .ASCII /A/
1117 002005          000          .BYTE 0
1118 002006          000          .BYTE 0
1119 002007          000          .BYTE 0
1120 002010          LSREV::          :REVISION LEVEL
1121 002010          102          .ASCII /B/
1122 002011          LSDEPO::          :0
1123 002011          060          .ASCII /O/
1124 002012          LSUNIT::          :NUMBER OF UNITS
1125 002012          000001          .WORD TSPTHV
1126 002014          LSTIML::          :LONGEST TEST TIME
1127 002014          000074          .WORD 60.
1128 002016          LSHPCP::          :POINTER TO H.W. QUES.
1129 002016          021426          .WORD LSHARD
1130 002020          LSSPCP::          :POINTER TO S.W. QUES.
1131 002020          000000          .WORD n
1132 002022          LSHPTP::          :PTR. TO DEF. H.W. PTABLE
1133 002022          002216          .WORD LSHW
1134 002024          LSSPTP::          :PTR. TO S.W. PTABLE
1135 002024          000000          .WORD 0
1136 002026          LSLADP::          :DIAG. END ADDRESS
1137 002026          021530          .WORD L$LAST
1138 002030          LSSTA::          :RESERVED FOR APT STATS
1139 002030          000000          .WORD 0
1140 002032          LSCO::          :DIAGNOSTIC TYPE
1141 002032          000000          .WORD 0
1142 002034          LSDTYP::          :DIAGNOSTIC TYPE
1143 002034          000000          .WORD 0
1144 002036          LSAPT::          :APT EXPANSION
1145 002036          000000          .WORD 0
1146 002040          LSDTP::          :PTR. TO DISPATCH TABLE
1147 002040          002124          .WORD L$DISPATCH

```

1148	002042		LSPRIO::		:DIAGNOSTIC RUN PRIORITY
1149	002042	000340		.WORD	PRI07
1150	002044		LSENV1::		:FLAGS DESCRIBE HOW IT WAS SETUP
1151	002044	000000		.WORD	0
1152	002046		LSEXP1::		:EXPANSION WORD
1153	002046	000000		.WORD	0
1154	002050		LSMREV::		:SVC REV AND EDIT #
1155	002050	003		.BYTE	CSREVISION
1156	002051	003		.BYTE	CSEDIT
1157	002052		LSEF::		:DIAG. EVENT FLAGS
1158	002052	000000		.WORD	0
1159	002054	000000		.WORD	0
1160	002056		LSSPC::		
1161	002056	000000		.WORD	0
1162	002060		LSDEVP::		: POINTER TO DEVICE TYPE LIST
1163	002060	002314		.WORD	LSDVTYP
1164	002062		LSREPP::		:PTR. TO REPORT CODE
1165	002062	000000		.WORD	0
1166	002064		LSEXP4::		
1167	002064	000000		.WORD	0
1168	002066		LSEXP5::		
1169	002066	000000		.WORD	0
1170	002070		LSAUT::		:PTR. TO ADD UNIT CODE
1171	002070	000000		.WORD	0
1172	002072		LSDUT::		:PTR. TO DROP UNIT CODE
1173	002072	000000		.WORD	0
1174	002074		LSLUN::		:LUN FOR EXERCISERS TO FILL
1175	002074	000000		.WORD	0
1176	002076		LSDESP::		:POINTER TO DIAG. DESCRIPTION
1177	002076	002326		.WORD	LSDESC
1178	002100		LSLOAD::		:GENERATE SPECIAL AUTOLOAD EMT
1179	002100	104035		EMT	ESLOAD
1180	002102		LSETP::		:POINTER TO ERR TBL
1181	002102	000000		.WORD	0
1182	002104		LSICP::		:PTR. TO INIT CODE
1183	002104	005570		.WORD	LS\$INIT
1184	002106		LSCCP::		:PTR. TO CLEAN-UP CODE
1185	002106	005750		.WORD	LSCLEAN
1186	002110		LSACP::		:PTR. TO AUTO CODE
1187	002110	005746		.WORD	LSAUTO
1188	002112		LSPRT::		:PTR. TO PROTECT TABLE
1189	002112	005562		.WORD	LSPROT
1190	002114		LSTEST::		:TEST NUMBER
1191	002114	000000		.WORD	0
1192	002116		LSDLY::		:DELAY COUNT
1193	002116	000000		.WORD	0
1194	002120		LSHIME::		:PTR. TO HIGH MEM
1195	002120	000000		.WORD	0
1196					

1197
1198
1199
1200
1201
1202
1203
1204 002122
1205 002122 000034
1206 002124
1207 002124 006014
1208 002126 006022
1209 002130 006024
1210 002132 006342
1211 002134 006524
1212 002136 006602
1213 002140 006666
1214 002142 006754
1215 002144 007024
1216 002146 007070
1217 002150 007166
1218 002152 007366
1219 002154 007742
1220 002156 010332
1221 002160 011210
1222 002162 012066
1223 002164 012574
1224 002166 013012
1225 002170 013206
1226 002172 013552
1227 002174 014062
1228 002176 014316
1229 002200 014706
1230 002202 015354
1231 002204 016236
1232 002206 016710
1233 002210 017652
1234 002212 020624
1235

.SBTTL DISPATCH TABLE

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

DISPATCH 28.
.WORD 28
LSDISPATCH::
.WORD T1
.WORD T2
.WORD T3
.WORD T4
.WORD T5
.WORD T6
.WORD T7
.WORD T8
.WORD T9
.WORD T10
.WORD T11
.WORD T12
.WORD T13
.WORD T14
.WORD T15
.WORD T16
.WORD T17
.WORD T18
.WORD T19
.WORD T20
.WORD T21
.WORD T22
.WORD T23
.WORD T24
.WORD T25
.WORD T26
.WORD T27
.WORD T28


```
1236 .SBTTL DEFAULT HARDWARE P-TABLE
1237
1238 :++
1239 : THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
1240 : THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
1241 : IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
1242 : AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
1243 :--
1244
1245 002214          BGNHW  DFPTBL
1246 002214 000002  .WORD  L10000-L$HW/2
1247 002216          L$HW::
1248 002216          DFPTBL::
1249
1250 002216 163010  .WORD  163010          ;CSR ADDRESS
1251 002220 000000  .WORD  0              ;DEVICE SELECTION NUMBER
1252
1253
1254 002222          ENDPHW
1255 002222 L10000:
1256
1257 .SBTTL SOFTWARE P-TABLE
1258
1259 :++
1260 : THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
1261 : PROGRAM AS OPERATIONAL PARAMETERS. THESE PARAMETERS ARE
1262 : SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
1263 : AT RUN TIME.
1264 :--
1265
1266 002222          BGNSW  SFPTBL
1267 002222 000000  .WORD  L10001-L$SW/2
1268 002224          L$SW::
1269 002224          SFPTBL::
1270
1271
1272 002224          ENDSW
1273 J02224 L10001:
1274
1275 002224          ENDMOD
```

1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331

002224

002224

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

.TITLE GLOBAL AREAS
.SBTTL GLOBAL EQUATES SECTION

BGNMOD

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

EQUALS

:
: BIT DIFINITIONS

BIT15== 100000
BIT14== 40000
BIT13== 20000
BIT12== 10000
BIT11== 4000
BIT10== 2000
BIT09== 1000
BIT08== 400
BIT07== 200
BIT06== 100
BIT05== 40
BIT04== 20
BIT03== 10
BIT02== 4
BIT01== 2
BIT00== 1

BIT9== BIT09
BIT8== BIT08
BIT7== BIT07
BIT6== BIT06
BIT5== BIT05
BIT4== BIT04
BIT3== BIT03
BIT2== BIT02
BIT1== BIT01
BIT0== BIT00

:
: EVENT FLAG DEFINITIONS
: EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

EF.START== 32. : START COMMAND WAS ISSUED
EF.RESTART== 31. : RESTART COMMAND WAS ISSUED
EF.CONTINUE== 30. : CONTINUE COMMAND WAS ISSUED
EF.NEW== 29. : A NEW PASS HAS BEEN STARTED
EF.PWR== 28. : A POWER-FAIL/POWER-UP OCCURRED

:
: PRIORITY LEVEL DEFINITIONS

1332	000340	PRI07== 340
1333	000300	PRI06== 300
1334	000240	PRI05== 240
1335	000200	PRI04== 200
1336	000140	PRI03== 140
1337	000100	PRI02== 100
1338	000040	PRI01== 40
1339	000000	PRI00== 0

: OPERATOR FLAG BITS

1340		:
1341		:
1342		:
1343	000004	EVL== 4
1344	000010	LOT== 10
1345	000020	ADR== 20
1346	000040	IDU== 40
1347	000100	ISR== 100
1348	000200	UAM== 200
1349	000400	BOE== 400
1350	001000	PNT== 1000
1351	002000	PRI== 2000
1352	004000	IXE== 4000
1353	010000	IBE== 10000
1354	020000	IER== 20000
1355	040000	LOE== 40000
1356	100000	HOE== 100000

: MEMORY SIMULATOR CONTROL REGISTER 0

1357		:
1358		:
1359		:
1360		:
1361		:
1362	100000	IDH== BIT15
1363		:
1364		:
1365		:
1366		:
1367		:
1368		:
1369	004000	SIG11H==BIT11
1370	002000	SIG10H==BIT10
1371	001000	SIG9H== BIT9
1372	000400	SIG8H== BIT8

:BIT15=1 READ DEVICE TYPE IN 15-8
:BITS 14-12 WILL READ BACK AS 0
:MS DEVICE TYPE EQUALS 400 (BIT8=1)

:BIT15=0 READ DEVICE NUMBER INTO
:BITS 11-8

:BITS 11-8 ARE USED TO SELECT THE
:DEVICE NUMBER TO ASSERT THE SIGNAL
:DEVE L. WHEN SELECTING MS THESE BITS
:MUST = THE SETTING OF DEV 3 - DEV 0

:BIT 7 UNUSED (ALWAYS READ AS A 0)

1373		:
1374		:
1375		:
1376	000100	CKH== BIT6
1377	000040	WRVH== BIT5
1378	000020	RDVH== BIT4
1379	000010	BIT8H== BIT3
1380	000004	MPH== BIT2
1381	000002	CTSH== BIT1
1382		:
1383	000001	RSTH== BIT0

:C OCK HIGH
:WRITE VIOLATION (READ ONLY)
:READ VIOLATION (READ ONLY)
:SELECT MS AS 8 BIT MODE NOT 16
:MAP PROTECT SELECT
:BIT1=1 MEM ACCESS FROM LSI-11 BUS
:BIT1=0 MEM ACCESS FROM SYSTEM BUS
:RESET MEMORY SIMULATOR MODULE

: MEMORY SIMULATOR CONTROL REGISTER 2

1384
1385
1386
1387

```

1388
1389
1390                                ;BITS 15-8 ARE UNUSED BITS
1391      000200      MSBRKH==BIT7      ;MEMORY SIMULATOR BREAK (READ ONLY)
1392      000100      WRENH== BIT6      ;WRITE ENABLE (READ ONLY)
1393      000040      ESRH==  BITS      ;ENABLE SIMULATOR RAM
1394
1395                                ;BIT 4 IS UNUSED (ALWAYS READ AS A 0)
1396
1397      000010      MSEL1== BIT3      ;MEMORY SELECTS
1398      000004      MSEL0== BIT2      ;
1399
1400      :      MSEL1=0 MSEL0=0 - SELECT SIMULATOR MEMORY - SSM L
1401      :      MSEL1=0 MSEL0=1 - SELECT MODULE SELECT MEMORY 0 - SMDS0 L
1402      :      MSEL1=1 MSEL0=0 - SELECT MAP PROTECT MEMORY - SMPM L
1403      :      MSEL1=1 MSEL0=1 - SELECT MODULE SELECT MEMORY 1 - SMDS1 L
1404
1405      000002      MSAD17==BIT1      ;MS ADDRESS 17
1406      000001      MSAD16==BIT0      ;MS ADDRESS 16
1407
1408      :
1409      ;MEMORY SIMULATOR CONTROL REGISTER 4
1410      :
1411
1412      100000      MSAD15==BIT15      ;MS ADDRESS 15
1413      040000      MSAD14==BIT14      ;MS ADDRESS 14
1414      020000      MSAD13==BIT13      ;MS ADDRESS 13
1415      010000      MSAD12==BIT12      ;MS ADDRESS 12
1416      004000      MSAD11==BIT11      ;MS ADDRESS 11
1417      002000      MSAD10==BIT10      ;MS ADDRESS 10
1418      001000      MSAD9== BIT9      ;MS ADDRESS 9
1419      000400      MSAD8== BIT8      ;MS ADDRESS 8
1420      000200      MSAD7== BIT7      ;MS ADDRESS 7
1421      000100      MSAD6== BIT6      ;MS ADDRESS 6
1422      000040      MSAD5== BIT5      ;MS ADDRESS 5
1423      000020      MSAD4== BIT4      ;MS ADDRESS 4
1424      000010      MSAD3== BIT3      ;MS ADDRESS 3
1425      000004      MSAD2== BIT2      ;MS ADDRESS 2
1426      000002      MSAD1== BIT1      ;MS ADDRESS 1
1427      000001      MSAD0== BIT0      ;MS ADDRESS 0
1428
1429      :
1430      ;MEMORY SIMULATOR MAP PROTECT BITS - CONTROL REGISTER 6
1431      :
1432
1433      000001      MPINH== BIT0      ;MAPPED INTO MEMORY SIMULATOR
1434      000002      WRENH== BIT1      ;WRITE ENABLED SIMULATOR MEMORY
1435      000004      RDEH== BIT2      ;READ ENABLED SIMULATOR MEMORY
1436      000010      MUTBH== BIT3      ;

```

1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484

002224
002224
002224 000000
002226 000000
002230 000000
002232 000000

002234 163010
002236 163012
002240 163014
002242 163016

002244 000000
002246 000000
002250 000000

002252 000000
002254 000000
002256 000000
002260 000000
002262 000000

002264 000000
002266 000000
002270 000000
002272 000000
002274 000000

002276 000000
002300 000000

002302 000000
002304 000000
002306 000000
002310 000000
002312 000000

.SBTTL GLOBAL DATA SECTION

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

ERRTBL
L\$ERRTBL::
ERRTYP:: .WORD 0
ERRNBR:: .WORD 0
ERRMSG:: .WORD 0
ERRBLK:: .WORD 0

:GLOBAL DATA FOR MEMORY SIMULATOR

REG0:: .WORD 163010 :CONTROL REGISTER 0
REG2:: .WORD 163012 :CONTROL REGISTER 2
REG4:: .WORD 163014 :CONTROL REGISTER 4
REG6:: .WORD 163016 :CONTROL REGISTER 6

IDDEV:: .WORD 0 :MEMORY SIMULATOR DEVICE # (11-2)
UNITNB:: .WORD 0 :
IDTYPE:: .WORD 0 :MEMORY SIMULATOR DEVICE TYPE (15-8)

R0LOAD:: .WORD 0 :WORD LOADED INTO REGISTER 0
R0GOOD:: .WORD 0 :EXPECTED REG 0
R0MASK:: .WORD 0 :REGISTER 0 MASK WORD
R0READ:: .WORD 0 :ACTUAL REG 0 READ
R0BAD:: .WORD 0 :REG 0 READ MINUS R0MASK BITS

R2LOAD:: .WORD 0 :WORD LOADED INTO REGISTER 2
R2GOOD:: .WORD 0 :EXPECTED REGISTER 2
R2MASK:: .WORD 0 :REGISTER 2 MASK WORD
R2READ:: .WORD 0 :ACTUAL REG 2 READ
R2BAD:: .WORD 0 :REG 2 READ MINUS R2MASK BITS

R4LOAD:: .WORD 0 :WORD LOADED INTO REGISTER 4
R4READ:: .WORD 0 :WORD READ OUT OF REGISTER 4

R6LOAD:: .WORD 0 :WORD LOADED INTO REGISTER 6
R6GOOD:: .WORD 0 :EXPECTED REGISTER 6
R6MASK:: .WORD 0 :REGISTER 6 MASK WORD
R6READ:: .WORD 0 :ACTUAL REGISTER 6 READ
R6BAD:: .WORD 0 :REG 6 READ MINUS MASK WORD

```
1485 .SBTTL GLOBAL TEXT SECTION
1486
1487 :++
1488 : THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
1489 : MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
1490 : MORE THAN ONE TEST.
1491 :--
1492
1493 :
1494 : NAMES OF DEVICES SUPPORTED BY PROGRAM
1495 :
1496 002314          DEVTYP <MDE/T-11>
1497 002314          LSDVTYP::
1498 002314 042115 027505 026524 .ASCIZ %MDE/T-11%
1499 002322 030461 000          .EVEN
1500 002326
1501
1502 :
1503 : TEST DESCRIPTION
1504 :
1505 002326          DESCRIPT      <MEMORY SIMULATOR DIAG.>
1506 002326          L$DESC::
1507 002326 042515 047515 054522 .ASCIZ /MEMORY SIMULATOR DIAG./
1508 002334 051440 046511 046125
1509 002342 052101 051117 042040
1510 002350 040511 027107 000
1511 002356          .EVEN
1512
1513 :
1514 : ASCII MESSAGES USED BY ERROR CALLS
1515 :
1516 :
1517 :
1518 002356 040504 040524 042440 MSGMP:: .ASCIZ /DATA ERROR IN MAP PROTECTION RAM/
1519 002364 051122 051117 044440
1520 002372 020116 040515 020120
1521 002400 051120 052117 041505
1522 002406 044524 047117 051040
1523 002414 046501 000
1524 002417 010 042104 042522 MSGMPS:: .ASCIZ /ADDRESS SHORT IN MAP PROTECTION RAM/
1525 002424 051523 051440 047510
1526 002432 052122 044440 020116
1527 002440 040515 020120 051120
1528 002446 052117 041505 044524
1529 002454 047117 051040 046501
1530 002462 000
1531 002463 0115 050101 050040 MSGMPL:: .ASCIZ /MAP PROTECT LOGIC ERROR/
1532 002470 047522 042524 052103
1533 002476 046040 043517 041511
1534 002504 042440 051122 051117
1535 002512 000
1536 002513 0104 052101 020101 MSGMDO:: .ASCIZ /DATA ERROR IN MODULE SELECT RAM 0/
1537 002520 051105 047522 020122
1538 002526 047111 046440 042117
1539 002534 046125 020105 042523
1540 002542 042514 052103 051040
```


1541	002550	046501	030040	000	
1542	002555	104	052101	020101	MSGMD1::ASCIZ /DATA ERROR IN MODULE SELECT RAM 1/
1543	002562	051105	047522	020122	
1544	002570	047111	046440	042117	
1545	002576	046125	020105	042523	
1546	002604	042514	052103	051040	
1547	002612	046501	030440	000	
1548	002617	103	044510	020120	MSGMDC::ASCIZ /CHIP ENABLE ERROR - MODULE SELECT RAM'S/
1549	002524	047105	041101	042514	
1550	002632	042440	051122	051117	
1551	002640	026440	046440	042117	
1552	002646	046125	020105	042523	
1553	002654	042514	052103	051040	
1554	002662	046501	051447	000	
1555	002667	115	042117	046125	MSGMDA::ASCIZ /MODULE SELECT RAM ADDRESSING ERROR/
1556	002674	020105	042523	042514	
1557	002702	052103	051040	046501	
1558	002710	040440	042104	042522	
1559	002716	051523	047111	020107	
1560	002724	051105	047522	000122	
1561	002732	040504	040524	042440	MSGMSR::ASCIZ /DATA ERROR IN MEMORY SIMULATOR RAM/
1562	002740	051122	051117	044440	
1563	002746	020116	042515	047515	
1564	002754	054522	051440	046511	
1565	002762	046125	052101	051117	
1566	002770	051040	046501	000	
1567	002775	103	044510	020120	MSGMSC::ASCIZ /CHIP ENABLE ERROR - MEMORY SIMULATOR RAM/
1568	003002	047105	041101	042514	
1569	003010	042440	051122	051117	
1570	003016	026440	046440	046505	
1571	003024	051117	020131	044523	
1572	003032	052515	040514	047524	
1573	003040	020122	040522	000115	
1574					.EVEN
1575					
1576					
1577					:
1578					: FORMAT STATEMENTS USED IN PRINT CALLS
1579					:
1580					
1581	003046	040445	047503	052116	EMSGR0::ASCIZ /%ACONTROL REG 0 ERROR%/
1582	003054	047522	020114	042522	
1583	003062	020107	020060	051105	
1584	003070	047522	022522	000116	
1585	003076	040445	047503	052116	EMSGR2::ASCIZ /%ACONTROL REG 2 ERROR%/
1586	003104	047522	020114	042522	
1587	003112	020107	020062	051105	
1588	003120	047522	022522	000116	
1589	003126	040445	047503	052116	EMSGR4::ASCIZ /%ACONTROL REG 4 ERROR%/
1590	003134	047522	020114	042522	
1591	003142	020107	020064	051105	
1592	003150	047522	022522	000116	
1593	003156	040445	047503	052116	EMSGR6::ASCIZ /%ACONTROL REG 6 ERROR%/
1594	003164	047522	020114	042522	
1595	003172	020107	020066	051105	
1596	003200	047522	022522	000116	

1597	003206	040445	042522	030107	REG0EQ::ASCIZ	/%AREGG = /
1598	003214	036440	000040			
1599	003220	040445	042522	031107	REG2EQ::ASCIZ	/%AREG2 = /
1600	003226	036440	000040			
1601	003232	040445	042522	032107	REG4EQ::ASCIZ	/%AREG4 = /
1602	003240	036440	000040			
1603	003244	040445	042522	033107	REG6EQ::ASCIZ	/%AREG6 = /
1604	003252	036440	000040			
1605	003256	040445	047514	042101	FRMTR0::ASCIZ	/%ALOAD: %06%S1%AREAD: %06%S1%AMASK: %06%S1%AGOOD: %06%S1%ABAD: %06%N/
1606	003264	020072	047445	022466		
1607	003272	030523	040445	042522		
1608	003300	042101	020072	047445		
1609	003306	022466	030523	040445		
1610	003314	040515	045523	020072		
1611	003322	047445	022466	030523		
1612	003330	040445	047507	042117		
1613	003336	020072	047445	022466		
1614	003344	030523	040445	040502		
1615	003352	035104	022440	033117		
1616	003360	047045	000			
1617	003363	045	046101	040517	FRMTR4::ASCIZ	/%ALOAD: %06%S1%AREAD: %06%N/
1618	003370	035104	022440	033117		
1619	003376	051445	022461	051101		
1620	003404	040505	035104	022440		
1621	003412	033117	047045	000		
1622	003417	045	052101	046511	MSGTM0::ASCIZ	/%ATIME OUT ERROR ADDRESSING CONTROL REG 0%N/
1623	003424	020105	052517	020124		
1624	003432	051105	047522	020122		
1625	003440	042101	051104	051505		
1626	003446	044523	043516	041440		
1627	003454	047117	051124	046117		
1628	003462	051040	043505	030040		
1629	003470	047045	000			
1630	003473	045	052101	046511	MSGTM2::ASCIZ	/%ATIME OUT ERROR ADDRESSING CONTROL REG 2%N/
1631	003500	020105	052517	020124		
1632	003506	051105	047522	020122		
1633	003514	042101	051104	051505		
1634	003522	044523	043516	041440		
1635	003530	047117	051124	046117		
1636	003536	051040	043505	031040		
1637	003544	047045	000			
1638	003547	045	052101	046511	MSGTM4::ASCIZ	/%ATIME OUT ERROR ADDRESSING CONTROL REG 4%N/
1639	003554	020105	052517	020124		
1640	003562	051105	047522	020122		
1641	003570	042101	051104	051505		
1642	003576	044523	043516	041440		
1643	003604	047117	051124	046117		
1644	003612	051040	043505	032040		
1645	003620	047045	000			
1646						
1647						
1648	003624				.EVEN	
1649						
1650						
1651						

.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 SERVICE.
:--

1652			
1653			
1654			
1655			
1656			
1657			
1658			
1659			
1660			
1661	003624		
1662	003624		
1663	003624		
1664	003624	012746	003046
1665	003630	012746	000001
1666	003634	010600	
1667	003636	104414	
1668	003640	062706	000004
1669	003644		
1670	003644	013746	002262
1671	003650	013746	002254
1672	003654	013746	002256
1673	003660	013746	002260
1674	003664	013746	002252
1675	003670	012746	003256
1676	003674	012746	000006
1677	003700	010600	
1678	003702	104415	
1679	003704	062706	000016
1680			
1681			
1682			
1683			
1684			
1685	003710		
1686	003710		
1687	003710	104423	
1688			
1689	003712		
1690	003712		
1691	003712		
1692	003712	012746	003076
1693	003716	012746	000001
1694	003722	010600	
1695	003724	104414	
1696	003726	062706	000004
1697	003732		
1698	003732	013746	002274
1699	003736	013746	002266
1700	003742	013746	002270
1701	003746	013746	002272
1702	003752	013746	002264
1703	003756	012746	003256
1704	003762	012746	000006
1705	003766	010600	
1706	003770	104415	
1707	003772	062706	000016

```

BGNMSG ROEROR
ROEROR::
PRINTB #EMSGR0
MOV #EMSGR0,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
PRINTX #FRMTR0,ROLOAD,ROREAD,ROMASK,ROGOOD,ROBAD
MOV ROBAD,-(SP)
MOV ROGOOD,-(SP)
MOV ROMASK,-(SP)
MOV ROREAD,-(SP)
MOV ROLOAD,-(SP)
MOV #FRMTR0,-(SP)
MOV #6,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #16,SP

```

L10002: ENDMSG

TRAP C\$MSG

BGNMSG R2EROR

```

R2EROR::
PRINTB #EMSGR2
MOV #EMSGR2,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
PRINTX #FRMTR0,R2LOAD,R2READ,R2MASK,R2GOOD,R2BAD
MOV R2BAD,-(SP)
MOV R2GOOD,-(SP)
MOV R2MASK,-(SP)
MOV R2READ,-(SP)
MOV R2LOAD,-(SP)
MOV #FRMTR0,-(SP)
MOV #6,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #16,SP

```

1708	003776		
1709	003.76		
1710	003776	104423	
1711			
1712	004000		
1713	004000		
1714	004000		
1715	004000	012746	003126
1716	004004	012746	000001
1717	004010	010600	
1718	004012	104414	
1719	004014	062706	000004
1720	004020		
1721	004020	013746	002300
1722	004024	013746	002276
1723	004030	012746	003363
1724	004034	012746	000003
1725	004040	010600	
1726	004042	104415	
1727	004044	062706	000010
1728	004050		
1729	004050		
1730	004050	104423	
1731			
1732	004052		
1733	004052		
1734	004052		
1735	004052	012746	003156
1736	004056	012746	000001
1737	004062	010600	
1738	004064	104414	
1739	004066	062706	000004
1740	004072	004737	004242
1741	004076		
1742	004076		
1743	004076	104423	
1744			
1745	004100		
1746	004100		
1747	004100		
1748	004100	012746	003046
1749	004104	012746	000001
1750	004110	010600	
1751	004112	104414	
1752	004114	062706	000004
1753	004120	004737	004242
1754	004124		
1755	004124		
1756	004124	104423	
1757			
1758	004126		
1759	004126		
1760	004126		
1761	004126	012746	003076
1762	004132	012746	000001
1763	004136	010600	

```

ENDMSG
L10003: TRAP CSMSG

BGNMSG R4EROR
R4EROR:: PRINTB #EMSGR4
MOV #EMSGR4,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP CSPNTB
ADD #4,SP
PRINTX #FRMTR4,R4LOAD,R4READ
MOV R4READ,-(SP)
MOV R4LOAD,-(SP)
MOV #FRMTR4,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP CSPNTX
ADD #10,SP
ENDMSG
L10004: TRAP CSMSG

BGNMSG ALINFO
ALINFO:: PRINTB #EMSGR6
MOV #EMSGR6,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP CSPNTB
ADD #4,SP
JSR PC,PRNTAL
ENDMSG
L10005: TRAP CSMSG

BGNMSG ALROIN
ALROIN:: PRINTB #EMSGRO
MOV #EMSGRO,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP CSPNTB
ADD #4,SP
JSR PC,PRNTAL
ENDMSG
L10006: TRAP CSMSG

BGNMSG ALR2IN
ALR2IN:: PRINTB #EMSGR2
MOV #EMSGR2,-(SP)
MOV #1,-(SP)
MOV SP,R0

```

;GO PRINT ALL THE REGISTERS

1764	004140	104414	
1765	004142	062706	000004
1766	004146	004737	004242
1767	004152		
1768	004152		
1769	004152	104423	
1770			
1771	004154		
1772	004154		
1773	004154		
1774	004154	012746	003417
1775	004160	012746	000001
1776	004164	010600	
1777	004166	104414	
1778	004170	062706	000004
1779	004174		
1780	004174		
1781	004174	104423	
1782			
1783	004176		
1784	004176		
1785	004176		
1786	004176	012746	003473
1787	004202	012746	000001
1788	004206	010600	
1789	004210	104414	
1790	004212	062706	000004
1791	004216		
1792	004216		
1793	004216	104423	
1794			
1795	004220		
1796	004220		
1797	004220		
1798	004220	012746	003547
1799	004224	012746	000001
1800	004230	010600	
1801	004232	104414	
1802	004234	062706	000004
1803	004240		
1804	004240		
1805	004240	104423	
1806			
1807	004242		
1808	004242	012746	003206
1809	004246	012746	000001
1810	004252	010600	
1811	004254	104415	
1812	004256	062706	000004
1813	004262		
1814	004262	013746	002262
1815	004266	013746	002254
1816	004272	013746	002256
1817	004276	013746	002260
1818	004302	013746	002252
1819	004306	012746	003256

```

TRAP C$PNTB
ADD #4,SP
JSR PC,PRNTAL
ENDMSG
L10007: TRAP C$MSG
BGNMSG ROTM
ROTM:: PRINTB #MSGTMO
MOV #MSGTMO,-(SP)
MOV #1,-(SP)
MOV SP,RO
TRAP C$PNTB
ADD #4,SP
ENDMSG
L10010: TRAP C$MSG
BGNMSG R2TM
R2TM:: PRINTB #MSGTM2
MOV #MSGTM2,-(SP)
MOV #1,-(SP)
MOV SP,RO
TRAP C$PNTB
ADD #4,SP
ENDMSG
L10011: TRAP C$MSG
BGNMSG R4TM
R4TM:: PRINTB #MSGTM4
MOV #MSGTM4,-(SP)
MOV #1,-(SP)
MOV SP,RO
TRAP C$PNTB
ADD #4,SP
ENDMSG
L10012: TRAP C$MSG
PRNTAL::PRINTX #REGOEQ
MOV #REGOEQ,-(SP)
MOV #1,-(SP)
MOV SP,RO
TRAP C$PNTX
ADD #4,SP
PRINTX #FRMTR0,ROLOAD,ROREAD,ROMASK,ROGOOD,ROBAD
MOV ROBAD,-(SP)
MOV ROGOOD,-(SP)
MOV ROMASK,-(SP)
MOV ROREAD,-(SP)
MOV ROLOAD,-(SP)
MOV #FRMTR0,-(SP)

```

:GO PRINT ALL THE REGISTERS

1820	004312	012746	000006	MOV	#6,-(SP)
1821	004316	010600		MOV	SP,R0
1822	004320	104415		TRAP	C\$PNTX
1823	004322	062706	000016	ADD	#16,SP
1824	004326			PRINTX	#REG2EQ
1825	004326	012746	003220	MOV	#REG2EQ,-(SP)
1826	004332	012746	000001	MOV	#1,-(SP)
1827	004336	010600		MOV	SP,R0
1828	004340	104415		TRAP	C\$PNTX
1829	004342	062706	000004	ADD	#4,SP
1830	004346			PRINTX	#FRMTR0,R2LOAD,R2READ,R2MASK,R2GOOD,R2BAD
1831	004346	013746	002274	MOV	R2BAD,-(SP)
1832	004352	013746	002266	MOV	R2GOOD,-(SP)
1833	004356	013746	002270	MOV	R2MASK,-(SP)
1834	004362	013746	002272	MOV	R2READ,-(SP)
1835	004366	013746	002264	MOV	R2LOAD,-(SP)
1836	004372	012746	003256	MOV	#FRMTR0,-(SP)
1837	004376	012746	000006	MOV	#6,-(SP)
1838	004402	010600		MOV	SP,R0
1839	004404	104415		TRAP	C\$PNTX
1840	004406	062706	000016	ADD	#16,SP
1841	004412			PRINTX	#REG4EQ
1842	004412	012746	003232	MOV	#REG4EQ,-(SP)
1843	004416	012746	000001	MOV	#1,-(SP)
1844	004422	010600		MOV	SP,R0
1845	004424	104415		TRAP	C\$PNTX
1846	004426	062706	000004	ADD	#4,SP
1847	004432			PRINTX	#FRMTR4,R4LOAD,R4READ
1848	004432	013746	002300	MOV	R4READ,-(SP)
1849	004436	013746	002276	MOV	R4LOAD,-(SP)
1850	004442	012746	003363	MOV	#FRMTR4,-(SP)
1851	004446	012746	000003	MOV	#3,-(SP)
1852	004452	010600		MOV	SP,R0
1853	004454	104415		TRAP	C\$PNTX
1854	004456	062706	000010	ADD	#10,SP
1855	004462			PRINTX	#REG6EQ
1856	004462	012746	003244	MOV	#REG6EQ,-(SP)
1857	004466	012746	000001	MOV	#1,-(SP)
1858	004472	010600		MOV	SP,R0
1859	004474	104415		TRAP	C\$PNTX
1860	004476	062706	000004	ADD	#4,SP
1861	004502			PRINTX	#FRMTR0,R6LOAD,R6READ,R6MASK,R6GOOD,R6BAD
1862	004502	013746	002312	MOV	R6BAD,-(SP)
1863	004506	013746	002304	MOV	R6GOOD,-(SP)
1864	004512	013746	002306	MOV	R6MASK,-(SP)
1865	004516	013746	002310	MOV	R6READ,-(SP)
1866	004522	013746	002302	MOV	R6LOAD,-(SP)
1867	004526	012746	003256	MOV	#FRMTR0,-(SP)
1868	004532	012746	000006	MOV	#6,-(SP)
1869	004536	010600		MOV	SP,R0
1870	004540	104415		TRAP	C\$PNTX
1871	004542	062706	000016	ADD	#16,SP
1872	004546	000207		RTS	PC
1873					

1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929

004550
004550 104404
004552
004552 012746 000340

```
.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:
: ROLOAD CONTAINS USER DEFINED UNIT NUMBER IN BITS 11-8
: ROMASK 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
: LDRDR1 ROUTINE TO LOAD, READ AND COMPARE REGISTER 1 (USED FOR DEVICE TYPE)
: LDRDR2 ROUTINE TO LOAD, READ AND COMPARE REGISTER 2
: LDRDR4 ROUTINE TO LOAD, READ AND COMPARE 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

:--

INITMS::BGNSEG                                ;ROUTINE TO INIT MS MODULE
        TRAP          CSBSEG
        SETVEC        #4,#18,#PRI07           ;SETUP VECTOR
        MOV           #PRI07,-(SP)
```

```

1930 004556 012746 004676      MOV    #1$,-(SP)
1931 004562 012746 000004      MOV    #4,-(SP)
1932 004566 012746 000003      MOV    #3,-(SP)
1933 004572 104437                TRAP   C$SVEC
1934 004574 062706 000010      ADD    #10,SP
1935
1936                                ;LOAD DEVICE NUMBER INTO REGISTER 0 AND CHECK IT
1937
1938 004600 013737 002244 002252      MOV    IDDEV,ROLOAD                ;GET USER DEFINED DEVICE NUMBER
1939 004506 012737 000060 002256      MOV    #RDVH!WRVH,ROMASK          ;SETUP TO MASK READ ONLY BITS
1940 004614 013737 002252 002254      MOV    ROLOAD,ROGOOD              ;PUT DATA LOADED INTO EXPECTED
1941 004622 013777 002252 175404      MOV    ROLOAD,@REGO               ;WRITE WORD TO REG 0
1942 004630 017737 175400 002260      MOV    @REGO,ROROAD               ;READ REGISTER CONTENTS BACK
1943 004636 013737 002260 002262      MOV    ROROAD,ROBAD               ;COPY REG 0 READ TO ALLOW MASKING
1944 004644 043737 002256 002262      BIC    ROMASK,ROBAD               ;CLEAR UNWANTED BITS OF REG 0
1945 004652 023737 002254 002262      CMP    ROGOOD,ROBAD               ;COMPARE EXPECTED WITH THAT READ
1946 004660 001414                BEQ    2$                          ;IF COMPARE WAS GOOD THEN CONT
1947 004662                                ERRDF  1,,ROEROR                    ;DEVICE # OR LB NOT = EXPECTED
1948 004662 104455                TRAP   C$ERDF
1949 004664 000001                .WORD  1
1950 004666 000000                .WORD  0
1951 004670 003624                .WORD  ROEROR
1952 004672                                CKLOOP
1953 004672 104406                TRAP   C$CLP1
1954 004674 000406                BR     2$                          ;BRANCH AROUND TIME OUT ERROR
1955 004676 005726                1$:  TST    (SP)+                      ;CLEAN UP STACK
1956 004700 005726                TST    (SP)+                      ;CLEAN UP STACK
1957 004702                                ERRDF  1,,ROTM                      ;TIME OUT ERROR REG 0
1958 004702 104455                TRAP   C$ERDF
1959 004704 000001                .WORD  1
1960 004706 000000                .WORD  0
1961 004710 004154                .WORD  ROTM
1962 004712                2$:  CLRVEC #4                          ;CLEAR VECTOR
1963 004712 012700 000004      MOV    #4,RO
1964 004716 104436                TRAP   C$CVEC
1965 004720                                ENDSEG
1966 004720                10000$:
1967 004720 104405                TRAP   C$ESEG
1968
1969                                ;READ DEVICE TYPE IN REGISTER 0
1970
1971 004722                                BGNSEG
1972 004722 104404                TRAP   C$BSEG
1973 004724 052737 100000 002252      BIS    #IDH,ROLOAD                ;SETUP TO READ DEVICE TYPE
1974 004732 013737 002250 002254      MOV    IDTYPE,ROGOOD              ;SETUP EXPECTED DATA
1975 004740 004737 005354                JSR    PC,LDRDOR                   ;LOAD, READ AND COMPARE REG 0
1976 004744 001404                BEQ    3$                          ;IF EQUAL THEN DEVICE TYPE COMPARED
1977 004746                                ERRDF  1,,ROEROR                    ;DEVICE TYPE NOT EQUAL EXPECTED
1978 004746 104455                TRAP   C$ERDF
1979 004750 000001                .WORD  1
1980 004752 000000                .WORD  0
1981 004754 003624                .WORD  ROEROR
1982 004756                3$:  ENDSEG
1983 004756                10001$:
1984 004756 104405                TRAP   C$ESEG
1985

```



```

2042 005164          ERRDF 2,,R2EROR          ;REGISTER 2 NOT EQUAL TO 0
2043 005164 104455  TRAP  C$ERDF
2044 005166 000002  .WORD 2
2045 005170 000000  .WORD 0
2046 005172 003712  .WORD R2EROR
2047 005174          CKLOOP
2048 005174 104406  TRAP  C$CLP1
2049 005176 000406  BR 7$          ;BRANCH AROUND TIME OUT ERROR
2050 005200 005726 6$: TST (SP)+    ;CLEAN UP STACK
2051 005202 005726  .ST (SP)+    ;CLEAN UP STACK
2052 005204          ERRDF 2,,R2TM          ;TIME OUT ERROR REG 2
2053 005204 104455  TRAP  C$ERDF
2054 005206 000002  .WORD 2
2055 005210 000000  .WORD 0
2056 005212 004176  .WORD R2TM
2057 005214          CLRVEC #4          ;CLEAR VECTOR
2058 005214 012700 000004  MOV #4,R0
2059 005220 104436  TRAP  C$CVEC
2060 005222          ENDSEG
2061 005222          10004$:
2062 005222 104405  TRAP  C$ESEG
2063
2064          ;CLEAR MSAD BITS 15:0 IN CONTROL REGISTER 4 AND CHECK THAT THESE BITS
2065          ;ARE CLEAR BY READING BACK CONTROL REGISTER 4.
2066
2067 005224          BGNSEG
2068 005224 104404  TRAP  C$BSEG
2069 005226          SETVEC #4,#8$,#PRI07          ;SETUP VECTOR
2070 005226 012746 000340  MOV #PRI07,-(SP)
2071 005232 012746 005320  MOV #8$,-(SP)
2072 005236 012746 000004  MOV #4,-(SP)
2073 005242 012746 000003  MOV #3,-(SP)
2074 005246 104437  TRAP  C$SVEC
2075 005250 062706 000010  ADD #10,SP
2076 005254 005037 002276  CLR R4LOAD          ;SETUP TO CLEAR ALL MSAD BITS
2077 005260 013777 002276 174752  MOV R4LOAD,@REG4    ;WRITE WORD INTO REGISTER 4
2078 005266 017737 174746 002300  MOV @REG4,R4READ    ;READ WORD BACK FROM REGISTER 4
2079 005274 023737 002276 002300  CMP R4LOAD,R4READ   ;COMPARE WORD LOADED WITH READ
2080 005302 001414          BEQ 9$          ;IF LOADED OK THEN CONTINUE
2081 005304          ERRDF 3,,R4EROR          ;REGISTER 4 NOT EQUAL TO ZERO
2082 005304 104455  TRAP  C$ERDF
2083 005306 000003  .WORD 3
2084 005310 000000  .WORD 0
2085 005312 004000  .WORD R4EROR
2086 005314          CKLOOP
2087 005314 104406  TRAP  C$CLP1
2088 005316 000406  BR 9$          ;BRANCH AROUND TIME OUT ERROR
2089 005320 005726 8$: TST (SP)+    ;CLEAN UP STACK
2090 005322 005726  .TST (SP)+    ;CLEAN UP STACK
2091 005324          ERRDF 3,,R4TM          ;TIME OUT ERROR REG 4
2092 005324 104455  TRAP  C$ERDF
2093 005326 000003  .WORD 3
2094 005330 000000  .WORD 0
2095 005332 004220  .WORD R4TM
2096 005334          CLRVEC #4          ;CLEAR VECTOR
2097 005334 012700 000004  MOV #4,R0

```

```

2098 005340 104436          TRAP   C$CVEC
2099 005342          ENDSEG
2100 005342          10005$:
2101 005342 104405          TRAP   C$ESEG
2102 005344 000207          RTS    PC                ;RETURN BACK TO TEST
2103
2104          ;ROUTINE TO LOAD, READ, AND COMPARE CONTENTS OF REGISTER 0
2105          ;CONDITION CODES ARE SET ON EXIT AS RESULT OF THE "CMP" INSTRUCTION.
2106
2107 005346 013737 002252 002254 LDRDR0:;MOV    R0LOAD,R0GOOD          ;PUT DATA LOADED INTO EXPECTED
2108 005354 013777 002252 174652 LDRDR0:;MOV    R0LOAD,@REG0        ;WRITE WORD TO REGISTER 0
2109 005362 017737 174646 002260 READR0:;MOV    @REG0,R0READ        ;READ REGISTER CONTENTS BACK
2110 005370 013737 002260 002262          MOV    R0READ,R0BAD        ;COPY REG 0 READ TO ALLOW MASKING
2111 005376 043737 002256 002262          BIC    R0MASK,R0BAD        ;CLEAR UNWANTED BITS OF REG 0
2112 005404 023737 002254 002262          CMP    R0GOOD,R0BAD        ;COMPARE EXPECTED WITH THAT READ
2113 005412 000207          RTS    PC                ;EXIT WITH CONDITION CODES SET
2114
2115          ;ROUTINE TO LOAD, READ, AND COMPARE CONTENTS OF REGISTER 2.
2116          ;CONDITION CODES ARE SET ON EXIT AS RESULT OF "CMP" INSTRUCTION
2117
2118 005414 013737 002264 002266 LDRDR2:;MOV    R2LOAD,R2GOOD        ;PUT DATA TO BE LOADED INTO EXPECTED
2119 005422 013777 002264 174606 LDRDR2:;MOV    R2LOAD,@REG2        ;WRITE BITS INTO REGISTER 2
2120 005430 017737 174602 002272 READR2:;MOV    @REG2,R2READ        ;READ REGISTER 2 BACK
2121 005436 013737 002272 002274          MOV    R2READ,R2BAD        ;COPY DATA READ
2122 005444 043737 002270 002274          BIC    R2MASK,R2BAD        ;CLEAR UNWANTED BITS IN REG 2
2123 005452 023737 002266 002274          CMP    R2GOOD,R2BAD        ;CHECK IF EXP EQUALS ACTUAL
2124 005460 000207          RTS    PC                ;EXIT WITH CONDITION CODES SET
2125
2126          ;ROUTINE TO LOAD, READ AND COMPARE CONTENTS OF REGISTER 4.
2127          ;CONDITION CODES ARE SET ON EXET AS RESULT OF "CMP" INSTRUCTION.
2128
2129 005462 013777 002276 174550 LDRDR4:;MOV    R4LOAD,@REG4        ;WRITE WORD INTO REGISTER 4
2130 005470 017737 174544 002300 READR4:;MOV    @REG4,R4READ        ;READ WORD BACK FROM REGISTER 4
2131 005476 023737 002276 002300          CMP    R4LOAD,R4READ        ;COMPARE WORD LOADED WITH READ
2132 005504 000207          RTS    PC                ;RETURN WITH CONDITION CODES SET
2133
2134          ;ROUTINE TO LOAD, READ AND COMPARE CONTENTS OF CONTROL REGISTER 6
2135          ;CONDITION CODES ARE SET ON EXIT AS RESULT OF "CMP" INSTRUCTION.
2136
2137 005506 013737 002302 002304 LDRDR6:;MOV    R6LOAD,R6GOOD        ;COPY DATA TO BE LOADED
2138 005514 013777 002302 174520 LDRDR6:;MOV    R6LOAD,@REG6        ;WRITE WORD INTO REGISTER 6
2139 005522 017737 174514 002310 READR6:;MOV    @REG6,R6READ        ;READ THE WORD BACK
2140 005530 013737 002310 002312          MOV    R6READ,R6BAD        ;COPY DATA READ
2141 005536 043737 002306 002312          BIC    R6MASK,R6BAD        ;MASK OUT UNWANTED BITS
2142 005544 023737 002304 002312          CMP    R6GOOD,R6BAD        ;COMPARE DATA LOADED WITH DATA READ
2143 005552 000207          RTS    PC                ;EXIT WITH CONDITON CODES SET
2144
2145 005554          ENDMOD
2146

```

```
2147      .TITLE MISCELLANEOUS SECTIONS
2148      .SBTTL REPORT CODING SECTION
2149
2150 005554      BGNMOD
2151
2152      :++
2153      : THE REPORT CODING SECTION CONTAINS THE
2154      : 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
2155      :--
2156
2157 005554      BGNRPT
2158 005554      L$RPT::
2159
2160
2161 005554      EXIT      RPT
2162 005554 000167      .WORD      JSJMP
2163 005556 000000      .WORD      L10013-2-.
2164
2165
2166      .EVEN
2167
2168 005560      ENDRPT
2169 005560      L10013:
2170 005560 104425      TRAP      C$RPT
2171
2172      .SBTTL PROTECTION TABLE
2173
2174      :++
2175      : THIS TABLE IS USED BY THE RUNTIME SERVICES
2176      : TO PROTECT THE LOAD MEDIA.
2177      :--
2178
2179 005562      BGNPROT
2180 005562      L$PROT::
2181
2182 005562 177777      -1          :OFFSET INTO P-TABLE FOR CSR ADDRESS
2183 005564 177777      -1          :OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
2184 005566 177777      -1          :OFFSET INTO P-TABLE FOR DRIVE NUMBER
2185
2186 005570      ENDPROT
2187
```

2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243

005570
005570
005570
005570 012700 000040
005574 104447
005576
005576 103410
005600
005600 012700 000037
005604 104447
005606
005606 103404
005610
005610 012700 000034
005614 104447
005616
005616 103001
005620
005620 104433
005622
005622 012700 000035
005626 104447
005630
005630 103003
005632 012737 177777 002246
005640
005640 012700 000036
005644 104447
005646
005646 103431
005650 005237 002246
005654
005654 013700 002246
005660 104442
005662 010005
005664
005664 103371
005666 012701 002234
005672 005002
005674 011511
005676 060221
005700 005202
005702 005202
005704 022702 000010
005710 001371
005712 005725
005714 005037 002244
005720 111537 002245
005724 012737 100400 002250

```

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

LSINIT: BGNINIT
        READEF #EF.START           ;SEE IF A START COMMAND
        MOV     #EF.START,R0
        TRAP   CSREFG
        BCOMPLETE 1$              ;BRANCH IF START COMMAND
        BCS    1$
        READEF #EF.RESTART        ;SEE IF A RESTART COMMAND
        MOV     #EF.RESTART,R0
        TRAP   CSREFG
        BCOMPLETE 1$              ;BRANCH IF RESTART
        BCS    1$
        READEF #EF.PWR            ;SEE IF RECOVERING FROM A POWER FAIL
        MOV     #EF.PWR,R0
        TRAP   CSREFG
        BNCOMPLETE 2$             ;IF NOT CHECK IN CONTINUE
        BCC    2$
1$:     BRESET                     ;INITIALIZE THE SYSTEM TO A KNOWN STATE
        TRAP   CSRESET
2$:     READEF #EF.NEW            ;SEE IF A NEW PASS
        MOV     #EF.NEW,R0
        TRAP   CSREFG
        BNCOMPLETE 3$             ;IF NOT GO CHECK IF CONTINUE
        BCC    3$
3$:     MOV     #-1,UNITNB         ;SETUP TO INIT UNIT NUMBER
        READEF #EF.CONTINUE       ;CHECK IF CONTINUE
        MOV     #EF.CONTINUE,R0
        TRAP   CSREFG
        BCOMPLETE 6$             ;IF YES THEN EXIT
        BCS    6$
4$:     INC     UNITNB            ;INC TO NEW UNIT NUMBER
        GPHARD UNITNB,R5          ;GET DEVICE INFORMATION
        MOV     UNITNB,R0
        TRAP   CS$GPHRD
        MOV     R0,R5
        BNCOMPLETE 4$            ;GO TRY ANOTHER UNIT
        BCC    4$
5$:     MOV     #REGO,R1          ;ADDRESS OF MS DEVICE ADDRESS TABLE
        CLR     R2                ;CLEAR OFFSET TO ADD TO TABLE ADDRESS
        MOV     (R5),(R1)         ;GET ADDRESS AND SAVE
        ADD     R2,(R1)+          ;ADD OFFSET TO ADDRESS
        INC     R2                ;UPDATE OFFSET BY 2
        INC     R2
        CMP     #10,R2            ;CHECK IF DONE LOADING TABLE
        BNE     5$               ;GO UPDATE NEXT ADDRESS
        TST     (R5)+             ;UPDATE THE POINTER
        CLR     IDDEV            ;CLEAR OUT DEVICE NUMBER
        MOVB   (R5),IDDEV+1       ;GET THE MS DEVICE NUMBER
        MOV     #IDH!SIG8H,IDTYPE ;SETUP MS DEVICE TYPE
    
```



```

2244 005732          6$:  SETPRI #PRI07          ;RAISE PROCESSOR PRIORITY
2245 005732 012700 000340  MOV      #PRI07,RO
2246 005736 104441      TRAP      C$SPRI
2247
2248
2249 005740          EXIT      INIT
2250 005740 104432      TRAP      C$EXIT
2251 005742 000002      .WORD    L10015-.
2252
2253
2254          .EVEN
2255
2256 005744          ENDINIT
2257 005744          L10015:
2258 005744 104411      TRAP      C$INIT
2259
2260          .SBTTL  AUTODROP SECTION
2261
2262          :++
2263          : THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
2264          : THE 'ADR' FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
2265          : SEE IF THEY WILL RESPOND.  THOSE THAT DON'T ARE IMMEDIATELY
2266          : DROPPED FROM TESTING.
2267          :--
2268
2269 005746          BGNAUTO
2270 005746          L$AUTO::
2271
2272
2273 005746          ENDAUTO
2274 005746          L10016:
2275 005746 104461      TRAP      C$AUTO
2276
2277          .SBTTL  CLEANUP CODING SECTION
2278
2279          :++
2280          : THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
2281          : AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
2282          :--
2283
2284 005750          BGNCLN
2285 005750          L$CLEAN::
2286 005750 013777 002244 174256  MOV      IDDEV,@REG0          ;CLEAR CONTROL REGISTER 0 EXCEPT
2287          ;FOR DEVICE NUMBER
2288 005756 012777 000000 174252  MOV      #0,@REG2          ;CLEAR REGISTER 2
2289 005764 012777 000000 174246  MOV      #0,@REG4          ;CLEAR REGISTER 4
2290
2291
2292 005772          EXIT      CLN
2293 005772 104432      TRAP      C$EXIT
2294 005774 000002      .WORD    L10017-.
2295
2296
2297          .EVEN
2298
2299 005776          ENDCLN
  
```

```

2300 005776          L10017:
2301 005776 104412   TRAP   C$CLEAN
2302
2303                 .SBTTL  DROP UNIT SECTION
2304
2305                 :++
2306                 : THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
2307                 : TO NO LONGER BE TESTED.
2308                 :--
2309
2310 006000          L$DU::  BGNDU
2311 006000
2312
2313
2314 006000          EXIT   DU
2315 006000 000167   .WORD  JSJMP
2316 006002 000000   .WORD  L10020-2-.
2317
2318
2319                 .EVEN
2320
2321 006004          ENDDU
2322 006004          L10020:
2323 006004 104453   TRAP   C$DU
2324
2325                 .SBTTL  ADD UNIT SECTION
2326
2327                 :++
2328                 : THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
2329                 : TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
2330                 : TO THE TEST CYCLE.
2331                 :--
2332
2333 006006          L$AU::  BGNAU
2334 006006
2335
2336
2337 006006          EXIT   AU
2338 006006 000167   .WORD  JSJMP
2339 006010 000000   .WORD  L10021-2-.
2340
2341
2342                 .EVEN
2343
2344 006012          ENDAU
2345 006012          L10021:
2346 006012 104452   TRAP   C$AU
2347
2348 006014          ENDMOD
2349
  
```

2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385

006014

006014

006014

006014 004737 004550

006020

006020

006020 104401

.TITLE HARDWARE TESTS

.SBTTL TEST 1: SELECT AND INITIALIZE MEMORY SIMULATOR

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.
:--

T1:: BGNST

JSR PC,INITMS ;INITIALIZE THE MEMORY SIMULATOR

L10022: ENDTST

TRAP C\$ETST

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

.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
:--
  
```

2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441

006022
006022
006022 004737 004550
006026
006026 104402
006030
006030 104404
006032 052737 000117 002252
006040 042737 000060 002256
006046 004737 005346
006052 001404
006054
006054 104455
006056 000001
006060 000000
006062 003624
006064
006064
006064 104405
006066
006066 104404
006070 105037 002252
006074 052737 000060 002256
006102 004737 005346
006106 001404
006110
006110 104455
006112 000001
006114 000000
006116 003624
006120
006120
006120 104405

```

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

T2.1: BGNSUB
      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

      BGNSEG
      TRAP   C$BSEG
      BIS    #RSTH!CTSH!MPH!BIT8H!CKH,ROLOAD ;SET ALL R/W BITS TO 1
      BIC    #RDVH!WRVH,ROMASK             ;CLEAR MASK BITS FOR RDV AND WRV
      JSR    PC,LDRDRO                     ;GO LOAD, READ AND COMPARE REG 0
      BEQ    1$                             ;IF LOAD EQUALS READ - CONTINUE
      ERRDF  1,ROEROR                       ;R/W BITS NOT ALL SET
      TRAP   C$ERDF
      .WORD  1
      .WORD  0
      .WORD  ROEROR
1$:   ENDSEG
10000$: TRAP   C$ESEG

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

      BGNSEG
      TRAP   C$BSEG
      CLRB   ROLOAD
      BIS    #RDVH!WRVH,ROMASK             ;CLEAR ALL R/W BITS TO BE LOADED
      JSR    PC,LDRDRO                     ;SETUP TO IGNORE RDV AND WRV BITS
      BEQ    2$                             ;GO LOAD, READ AND COMPARE REG 0
      ERRDF  1,ROEROR                       ;IF ALL R/W BITS ARE 0 - CONT
      TRAP   C$ERDF
      .WORD  1
      .WORD  0
      .WORD  ROEROR
2$:   ENDSEG
10001$: TRAP   C$ESEG
  
```

```

2442 006122          ENDSUB
2443 006122          L10024: TRAP    C$ESUB
2444 006122 104403
2445
2446 006124          BGNSUB
2447 006124          T2.2: TRAP    C$BSUB
2448 006124 104402
2449
2450          ;CHECK THAT ALTERNATING BITS IN REG 0 CAN BE SET AND CLEARED
2451          ;
2452          ;CHECK THAT CTS H AND 8 BIT H ARE 0 WHEN RST H, MP H AND CK H ARE 1
2453          ;RDV H AND WRV H ARE CHECKED TO BE 0
2454
2455 006126          BGNSEG
2456 006126 104404   TRAP    C$BSEG
2457 006130 112737   000105 002252   MOVB   #RSTH!MPH!CKH,ROLOAD ;SETUP BITS TO LOAD
2458 006136 042737   000060 002256   BIC   #RDVH!WRVH,ROMASK ;SETUP TO CHECK RDV AND WRV
2459 006144 004737   005346          JSR   PC,LDRDRO ;GO LOAD, READ AND COMPARE REG 0
2460 006150 001404          BEQ   3$ ;IF OK THEN CONTINUE
2461 006152          ERRDF 1,,ROEROR ;A BIT IS SHORTED TO ANOTHER
2462 006152 104455   TRAP   C$ERDF
2463 006154 000001   .WORD 1
2464 006156 000000   .WORD 0
2465 006160 003624   .WORD ROEROR
2466 006162          3$: ENDSEG
2467 006162          10000$:
2468 006162 104405   TRAP   C$ESEG
2469
2470          ;CHECK THAT RST H, MP H, AND CK H ARE 0 WHEN CTS H AND 8 BIT H ARE 1
2471          ;RDV H AND WRV H ARE IGNORED
2472
2473 006164          BGNSEG
2474 006164 104404   TRAP   C$BSEG
2475 006166 112737   000C12 002252   MOVB   #CTSH!BIT8H,ROLOAD ;SETUP BITS TO LOAD
2476 006174 052737   000060 002256   BIS   #RDVH!WRVH,ROMASK ;SETUP TO IGNORE RDV AND WRV
2477 006202 004737   005346          JSR   PC,LDRDRO ;GO LOAD, READ AND COMPARE REG 0
2478 006206 001404          BEQ   4$ ;IF OK THEN CONTINUE
2479 006210          ERRDF 1,,ROEROR ;A BIT IS SHORTED TO ANOTHER
2480 006210 104455   TRAP   C$ERDF
2481 006212 000001   .WORD 1
2482 006214 000000   .WORD 0
2483 006216 003624   .WORD ROEROR
2484 006220          4$: ENDSEG
2485 006220          10001$:
2486 006220 104405   TRAP   C$ESEG
2487
2488 006222          ENDSUB
2489 006222          L10025:
2490 006222 104403   TRAP   C$ESUB
2491
2492 006224          ENDTST
2493 006224          L10023:
2494 006224 104401   TRAP   C$ETST
2495

```

2496
 2497
 2498
 2499
 2500
 2501
 2502
 2503
 2504
 2505
 2506
 2507
 2508
 2509
 2510
 2511
 2512
 2513
 2514
 2515
 2516
 2517
 2518
 2519
 2520
 2521
 2522
 2523
 2524
 2525
 2526
 2527
 2528
 2529
 2530
 2531
 2532
 2533
 2534
 2535
 2536

006226
 006226
 006226 004737 004550
 006232 105037 002252
 006236 104404
 006240 052737 000060 002256
 006246 032737 000001 002252
 006254 001403
 006256 042737 000060 002256
 006264 004737 005346
 006270 001404
 006272
 006272 104455
 006274 000001
 006276 000000
 006300 003624
 006302
 006302
 006302 104405
 006304 005237 002252
 006310 032737 ^0020 002252
 006316 001747
 006320 032737 000100 002252
 006326 001004
 006330 112737 000100 002252
 006336 000737
 006340
 006340
 006340 104401

.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.
:--
  
```

```

T3:: BGNTST
      JSR    PC,INITMS      ;SELECT MS AND INITIALIZE IT
      CLRB   ROLOAD        ;SET LOW BYTE OF REG 0 TO 0
1$:   BGNSEG
      TRAP   C$BSEG
      BIS    #RDVH!WRVH,ROMASK ;SETUP TO IGNORE RDV H AND WRV H
      BIT    #RSTH,ROLOAD    ;CHECK IF SIGNAL RST H WILL BE SET
      BEQ    2$             ;IF NOT THEN IGNORE RDV H AND WRV H
2$:   BIC    #RDVH!WRVH,ROMASK ;SETUP TO CHECK RDV H AND WRV H
      JSR    PC,LDRDRO      ;GO LOAD, READ AND CHECK REGISTER 0
      BEQ    3$             ;IF LOADED OK THEN CONTINUE
      ERDF   1,ROEROR      ;REGISTER 0 NOT EQUAL EXPECTED
      TRAP   C$ERDF
      .WORD  1
      .WORD  0
      .WORD  ROEROR
3$:   ENDSEG
10000$: TRAP   C$ESEG
      INC    ROLOAD        ;UPDATE TEST PATTERN BY 1
      BIT    #BIT4,ROLOAD  ;CHECK IF LOW R/W BITS DONE
      BEQ    1$            ;IF NOT LOAD NEXT BIT
      BIT    #CKH,ROLOAD   ;CHECK IF 2ND PORTION DONE
      BNE    4$            ;IF YES THEN END OF TEST
      MOVB   #CKH,ROLOAD   ;SETUP TO DO 2ND PORTION OF TEST
      BR     1$            ;DO BINARY COUNT AGAIN WITH CK H SET
4$:   ENDTST
L10026: TRAP   C$ETST
  
```

2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592

006342
006342
006342 004737 004550
006346 012737 177740 002270

006354
006354
006354 104402

006356
006356 104404
006360 012737 000017 002264
006366 004737 005414
006372 001404
006374
006374 104455
006376 000002
006400 000000
006402 003712
006404
006404
006404 104405

006406
006406 104404
006410 005037 002264
006414 004737 005414
006420 001404
006422
006422 104455
006424 000002
006426 000000
006430 003712
006432
006432
006432 104405
006434
006434

```
.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
:--

14::      BGNSTST
          JSR      PC,INITMS      ;SELECT AND INIT MEM SIM
          MOV      #177740,R2MASK ;SETUP REG 2 MASK WORD

T4.1:     BGNSUB
          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

          BGNSEG
          TRAP     C$BSEG
          MOV      #MSAD16!MSAD17!MSEL0!MSEL1,R2LOAD ;SETUP BITS TO LOAD
          JSR      PC,LDRDR2      ;GO LOAD, READ AND COMPARE REG 2
          BEQ      1$             ;IF ALL ONES THEN CONTINUE
          ERRDF   2,,R2EROR      ;A BIT(S) FAILED TO SET
          TRAP     C$ERDF
          .WORD   2
          .WORD   0
          .WORD   R2EROR
1$:       ENDSEG
10000$:   TRAP     C$ESEG

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

          BGNSEG
          TRAP     C$BSEG
          CLR      R2LOAD
          JSR      PC,LDRDR2      ;SETUP BITS TO LOAD TO 0
          BEQ      2$             ;GO LOAD, READ AND COMPARE REG 2
          ERRDF   2,,R2EROR      ;IF ALL ZERO THEN CONT
          TRAP     C$ERDF
          .WORD   2
          .WORD   0
          .WORD   R2EROR
2$:       ENDSEG
10001$:   TRAP     C$ESEG

L10030:   ENDSUB
```

```

2593 006434 104403          TRAP  C$ESUB
2594
2595 006436
2596 006436
2597 006436 104402          T4.2: TRAP  C$BSUB
2598
2599                                ;CHECK THAT ALTERNATING BITS IN REG 2 CAN BE SET AND CLEARED.
2600                                ;
2601                                ;CHECK THAT MSAD17 H AND MSEL1 H ARE 0 WHEN MSAD16 H AND MSEL0 H
2602                                ;ARE SET TO A ONE.
2603
2604 006440          BGNSEG
2605 006440 104404          TRAP  C$BSEG
2606 006442 012737 000005 002264 MOV   #MSAD16!MSEL0,R2LOAD  ;SETUP BITS TO LOAD INTO REG 2
2607 006450 004737 005414 JSR   PC,LDRDR2             ;GO LOAD, READ, AND COMPARE REG 2
2608 006454 001404          BEQ   1$                    ;IF OK THEN CONTINUE
2609 006456          ERRDF  2,,R2EROR  ;ALTERNATING 1'S AND 0'S FAILED
2610 006456 104455          TRAP  C$ERDF
2611 006460 000002          .WORD  2
2612 006462 000000          .WORD  0
2613 006464 003712          .WORD  R2EROR
2614 006466          1$:
2615 006466          10000$:
2616 006466 104405          TRAP  C$ESEG
2617
2618                                ;CHECK THAT MSAD16 H AND MSEL0 H ARE 0 WHEN MSAD17 H AND MSEL1 H
2619                                ;ARE SET TO A ONE
2620
2621 006470          BGNSEG
2622 006470 104404          TRAP  C$BSEG
2623 006472 012737 000012 002264 MOV   #MSAD17!MSEL1,R2LOAD  ;SETUP BITS TO LOAD
2624 006500 004737 005414 JSR   PC,LDRDR2             ;GO LOAD, READ, AND COMPARE REG 2
2625 006504 001404          BEQ   2$                    ;IF EQUAL THEN CONT
2626 006506          ERRDF  2,,R2EROR  ;ALTERNATING 0'S AND 1'S
2627 006506 104455          TRAP  C$ERDF
2628 006510 000002          .WORD  2
2629 006512 000000          .WORD  0
2630 006514 003712          .WORD  R2EROR
2631 006516          2$:
2632 006516          10001$:
2633 006516 104405          TRAP  C$ESEG
2634 006520          ENDSUB
2635 006520          L10031:
2636 006520 104403          TRAP  C$ESUB
2637 006522          ENDTST
2638 006522          L10027:
2639 006522 104401          TRAP  C$ETST
2640

```


2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674

006524
006524
006524 004737 004550
006530 012701 000020
006534 005037 002264
006540 012737 177740 002270
006546
006546 104404
006550 004737 005414
006554 001404
006556 104455
006560 000002
006562 000000
006564 003712
006566
006566 104405
006570 005237 002264
006574 005301
006576 001363
006600
006600
006600 104401

.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.
:--

TS:: BGNTST
JSR PC,INITMS ;SELECT AND INIT MEMORY SIM
MOV #20,R1 ;SETUP TEST COUNTER
CLR R2LOAD ;SET PATTERN INITIALLY TO 0
MOV #177740,R2MASK ;SETUP REG 2 MASK WORD
1\$: BGNSEG
TRAP C\$BSEG
JSR PC,LDRDR2 ;GO LOAD, READ AND COMPARE REG 2
BEQ 2\$;IF COMPARED OK THEN CONT
ERRDF 2,,R2EROR ;DATA LOADED NOT EQ. EXPECTED
TRAP C\$ERDF
.WORD 2
.WORD 0
.WORD R2EROR
2\$: ENDSEG
10000\$: TRAP C\$ESEG
INC R2LOAD ;UPDATE THE TEST PATTERN
DEC R1 ;DECREMENT THE TEST COUNTER
BNE 1\$;IF NOT 0 THEN DO NEXT PATTERN
ENDTST
L10032: TRAP C\$ETST

2675
 2676
 2677
 2678
 2679
 2680
 2681
 2682
 2683
 2684
 2685
 2686
 2687
 2688
 2689
 2690
 2691
 2692
 2693
 2694
 2695
 2696
 2697
 2698
 2699
 2700
 2701
 2702
 2703
 2704
 2705
 2706
 2707
 2708
 2709
 2710
 2711
 2712
 2713
 2714
 2715
 2716
 2717
 2718
 2719
 2720
 2721
 2722

006602
 006602
 006602 004737 004550
 006606
 006606 104404
 006610 012737 177777 002276
 006616 004737 005462
 006622 001404
 006624 104455
 006624 000003
 006626 000000
 006630 004000
 006634
 006634 104405
 006634
 006636 104404
 006636 005037 002276
 006640 004737 005462
 006650 001404
 006652 104455
 006652 000003
 006654 000000
 006660 004000
 006662
 006662 104405
 006664
 006664 104401

```

.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.
:--
BGNTST
T6:: JSR PC,INITMS ;SELECT AND INIT MEM SIMULATOR
;SET MSAD BITS 15 THROUGH 0 TO A 1
BGNSEG
TRAP C$BSEG
MOV #1,R4LOAD ;SETUP TO LOAD ALL ONES INTO REG 4
JSR PC,LDRDR4 ;LOAD, READ AND COMPARE REG 4
BEQ 1$ ;IF ALL ONES THEN CONT
ERRDF 3,R4EROR ;FAILED TO LOAD ALL ONES
TRAP C$ERDF
.WORD 3
.WORD 0
.WORD R4EROR
1$: ENDSEG
10000$: TRAP C$ESEG
;SET MSAD BITS 15 THROUGH 0 TO A 0
BGNSEG
TRAP C$BSEG
CLR R4LOAD ;SETUP TO LOAD ALL ZEROES INTO REG 4
JSR PC,LDRDR4 ;LOAD, READ AND COMPARE REG 4
BEQ 2$ ;IF ALL ZEROES THEN CONT
ERRDF 3,R4EROR ;REGISTER 4 NOT ALL 0'S
TRAP C$ERDF
.WORD 3
.WORD 0
.WORD R4EROR
2$: ENDSEG
10001$: TRAP C$ESEG
ENDTGT
L10033: TRAP C$ETST
  
```

```

2723      .SBTTL TEST 7: CONTROL REG 4 TEST (1'S + 0'S, AND 0'S + 1'S)
2724
2725      :++
2726      : TEST TO CHECK CONTROL REGISTER 4 BY LOADING ALTERNATING ONES AND ZEROES AND
2727      : THEN ZEROES AND ONES. CHECKS THAT ADJACENT BITS ARE NOT SHORED TOGETHER.
2728      :--
2729
2730      006666      BGNTST
2731      006666
2732      006666      004737      004550      T7::      JSR      PC,INITMS      ;SELECT AND INIT MS
2733
2734      ;LOAD REGISTER 4 WITH 125252 PATTERN
2735
2736      006672      BGNSEG
2737      006672      104404      TRAP      C$BSEG
2738      006674      012737      125252      002276      MOV      #125252,R4LOAD      ;SETUP PATTERN TO LOAD
2739      006702      004737      005462      JSR      PC,LDRDR4      ;LOAD, READ AND COMPARE REGISTER 4
2740      006706      001404      BEQ      1$      ;IF PATTERN OK THEN CONTINUE
2741      006710      ERRDF      3,,R4EROR      ;PATTERN READ NEQ 125252
2742      006710      104455      TRAP      C$ERDF
2743      006712      000003      .WORD      3
2744      006714      000000      .WORD      0
2745      006716      004000      .WORD      R4EROR
2746      006720      1$:      ENDSEG
2747      006720      10000$:
2748      006720      104405      TRAP      C$ESEG
2749
2750      ;LOAD REGISTER 4 WITH 052525 PATTERN
2751
2752      006722      BGNSEG
2753      006722      104404      TRAP      C$BSEG
2754      006724      012737      052525      002276      MOV      #052525,R4LOAD      ;SETUP PATTERN TO LOAD
2755      006732      004737      005462      JSR      PC,LDRDR4      ;LOAD, READ AND COMPARE REGISTER 4
2756      006736      001404      BEQ      2$      ;IF PATTERN OK THEN CON
2757      006740      ERRDF      3,,R4EROR      ;PATTERN READ NEQ 05252
2758      006740      104455      TRAP      C$ERDF
2759      006742      000003      .WORD      3
2760      006744      000000      .WORD      0
2761      006746      004000      .WORD      R4EROR
2762      006750      2$:      ENDSEG
2763      006750      10001$:
2764      006750      104405      TRAP      C$ESEG
2765      006752      ENDTST
2766      006752      L10034:
2767      006752      104401      TRAP      C$ETST
2768

```

2769
 2770
 2771
 2772
 2773
 2774
 2775 006754
 2776 006754
 2777 006754 004737 004550
 2778 006760 005037 002276
 2779 006764
 2780 006764 104404
 2781 006766 004737 005462
 2782 006772 001404
 2783 006774
 2784 006774 104455
 2785 006776 000003
 2786 007000 000000
 2787 007002 004000
 2788 007004
 2789 007004
 2790 007004 104405
 2791 007006 005237 002276
 2792 007012 032737 000400 002276
 2793 007020 001761
 2794 007022
 2795 007022
 2796 007022 104401
 2797

```
.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
:--

T8::      BGNTST
          JSR      PC,INITMS      ;SELECT AND INIT MEM SIM
          CLR      R4LOAD        ;SET PATTERN INITIALLY TO 0
1$:       BGNSEG
          TRAP     C$BSEG
          JSR      PC,LDRDR4     ;LOAD, READ AND COMPARE PATTERN
          BEQ      2$           ;IF PATTERN OK THEN CONT
          ERRDF    3,R4EROR     ;PATTERN LOADED NEG PATTERN READ
          TRAP     C$ERDF
          .WORD    3
          .WORD    0
          .WORD    R4EROR
2$:       ENDSEG
10000$:   TRAP     C$ESEG
          INC      R4LOAD
          BIT      #BIT8,R4LOAD ;UPDATE THE PATTERN BY 1
          BEQ      1$           ;CHECK IF DONE
          ENDTST                ;IF NOT THEN LOAD NEXT PATTERN
L10035:   TRAP     C$ETST
```

2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825

007024
007024
007024 004737 004550
007030 005037 002276
007034
007034 104404
007036 004737 005462
007042 001404
007044
007044 104455
007046 000003
007050 000000
007052 004000
007054
007054
007054 104405
007056 062737 000400 002276
007064 001363
007066
007066
007066 104401

```
.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
:--

          BGNTST
T9::
          JSR    PC,INITMS      ;SELECT AND INIT MEM SIM
          CLR    R4LOAD        ;SET PATTERN INITIALLY TO 0
1$:
          BGNSEG
          TRAP   C$BSEG
          JSR    PC,LDRDR4     ;LOAD, READ AND COMPARE PATTERN
          BEQ    2$           ;IF PATTERN OK THEN CONT
          ERRDF  3,,R4EROR    ;PATTERN LOADED NEG PATTERN READ
          TRAP   C$ERDF
          .WORD  3
          .WORD  0
          .WORD  R4EROR
2$:
          ENDSEG
10000$:
          TRAP   C$ESEG
          ADD    #BIT8,R4LOAD  ;UPDATE THE DATA PATTERN
          BNE    1$           ;IF NOT 0 THEN GO LOAD DATA PATTERN
          ENDTST
L10036:
          TRAP   C$ETST
```

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

SEQ 0057

2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877

.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.
:--
  
```

```

T10:: BGNTST
      JSR    PC,INITMS          ;SELECT AND INIT THE MEMORY SIMULATOR
      BGNSEG
      TRAP   CSBSEG

      ;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.

      BIS    #RSTH!MPH,ROLOAD   ;SETUP TO SET RST H AND MP H TO 1
      BIC    #RDVH!WRVH,ROMASK  ;SETUP TO CHECK RDV AND WRV BITS
      JSR    PC,LDRDRO          ;GO LOAD, READ AND CHECK REG 0
      BEQ    1$                ;IF OK THEN CONTINUE
      ERRDF  1,,ROEROR         ;REGISTER 0 NOT EQUALA EXPECTED
      TRAP   C$ERDF

      .WORD  1
      .WORD  0
      .WORD  ROEROR
      CKLOOP
      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.

1$:   CLR    R2LOAD             ;SETUP TO LOAD ALL ZEROES
      MOV    #177540,R2MASK    ;SETUP REG 2 MASK WORD
      JSR    PC,LDRDR2        ;GO LOAD, READ AND CHECK REG 2
      BEQ    2$                ;IF REG 2 OK THEN CONT
      ERRDF  2,,R2EROR        ;REG 2 NOT EQUAL EXPECTED
      TRAP   C$ERDF

      .WORD  2
      .WORD  0
      .WORD  R2EROR
2$:   ENDSEG
10000$: TRAP   C$ESEG
      ENDTST
L10037: TRAP   C$ETST
  
```

```

007070
007070
007070 004737 004550
007074
007074 104404
007076 052737 000005 002252
007104 042737 000060 002256
007112 004737 005346
007116 001405
007120
007120 104455
007122 000001
007124 000000
007126 003624
007130
007130 104406
007132 005037 002264
007136 012737 177540 002270
007144 004737 005414
007150 001404
007152
007152 104455
007154 000002
007156 000000
007160 003712
007162
007162 104405
007164
007164
007164 104401
  
```

2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933

007166
007166
007166 004737 004550
007172 005001
007174 005002

007176
007176 104404

007200 012737 000010 002264
007206 050237 002264
007212 012737 177740 002270
007220 004737 005414
007224 001405
007226 104455
007230 000002
007232 000000
007234 003712
007236 104406

007240 010137 002276
007244 004737 005462
007250 001405
007252 104455
007254 000003
007256 000000
007260 004000
007262 104406

```
.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.
:--
T11:: BGNTST
      JSR    PC,INITMS      ;SELECT AND INIT MEMORY SIMULATOR
      CLR    R1             ;CLEAR BITS 15-8 MSAD ADDRESS COUNTER
      CLR    R2             ;CLEAR BITS 17-16 MSAD ADDRESS COUNTER

1$:   BGNSEG
      TRAP   C$BSEG

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

      MOV    #MSEL1,R2LOAD  ;SETUP TO SET MSEL1 BIT TO A 1
      BIS    R2,R2LOAD      ;SETUP MSAD BITS 17 AND 16
      MOV    #177740,R2MASK ;SETUP TO IGNORE REG 2 BITS 15-5
      JSR    PC,LDRDR2     ;GO LOAD, READ AND COMPARE REG 2
      BEQ    2$            ;IF LOADED OK THEN CONTINUE
      ERDF   2,,R2EROR    ;REGISTER 2 FAILED TO LOAD CORRECTLY
      TRAP   C$ERDF
      .WORD  2
      .WORD  0
      .WORD  R2EROR
      CKLOOP
      TRAP   C$CLP1

      ;LOAD MSAD BITS 15-8 INTO CONTROL REGISTER 4

2$:   MOV    R1,R4LOAD      ;SETUP BITS TO LOAD
      JSR    PC,LDRDR4     ;GO LOAD, READ AND COMPARE REG 4
      BEQ    3$            ;IF REGISTER 4 CORRECT THEN CONT
      ERDF   3,,R4EROR    ;REGISTER 4 NOT EQUAL EXPECTED
      TRAP   C$ERDF
      .WORD  3
      .WORD  0
      .WORD  R4EROR
      CKLOOP
      TRAP   C$CLP1

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

```

2934                                     ;BY CONTROL REGISTER 2 AND 4. THE SIGNAL SMPM L WILL BE ASSERTED
2935                                     ;ON THE WRITE/READ TO REGISTER 6
2936
2937 007264 012737 177760 002306 3$: MOV #177760,R6MASK ;SETUP REG 6 MASK WORD
2938 007272 012737 000017 002302 MOV #MUTBH!MPINH!WREH!RDEH,R6LOAD
2939 007300 004737 005506 JSR PC,LDRDR6 ;GO LOAD, READ AND COMPARE REGISTER 6
2940 007304 001405 BEQ 4$ ;IF DATA OK THEN CONTINUE
2941 007306 ERRDF 4,MSGMP,ALINFO ;MAP PROTECTION RAM DATA ERROR
2942 007306 104455 TRAP C$ERDF
2943 007310 000004 .WORD 4
2944 007312 002356 .WORD MSGMP
2945 007314 004052 .WORD ALINFO
2946 007316 CKLOOP
2947 007316 104406 TRAP C$CLP1
2948
2949                                     ;WRITE THE SAME LOCATION WITH ALL THE BITS SET TO A 0
2950
2951 007320 005037 002302 4$: CLR R6LOAD ;SETUP ALL BITS TO BE LOADED AS A 0
2952 007324 004737 005506 JSR PC,LDRDR6 ;GO LOAD,READ AND COMPARE REG 6
2953 007330 001404 BEQ 5$ ;IF OK THEN CONTINUE
2954 007332 ERRDF 4,MSGMP,ALINFO ;DATA ERROR MAP PROTECTION RAM
2955 007332 104455 TRAP C$ERDF
2956 007334 000004 .WORD 4
2957 007336 002356 .WORD MSGMP
2958 007340 004052 .WORD ALINFO
2959 007342 5$: ENDSEG
2960 007342 10000$: TRAP C$ESEG
2961 007342 104405
2962
2963 007344 062701 000400 ADD #MSAD8,R1 ;UPDATE MSAD BITS 15-8 BY 1
2964 007350 001312 BNE 1$ ;IF NOT 0 THEN DO NEXT ADDRESS
2965 007352 062702 000001 ADD #MSAD16,R2 ;UPDATE MSAD BITS 16 AND 17
2966 007356 032702 000004 BIT #MSELO,R2 ;CHECK IF DONE
2967 007362 001705 BEQ 1$ ;IF NOT DONE DO RANGE OF ADDRESSES
2968 007364 ENDTST
2969 007364 L10040: TRAP C$ETST
2970 007364 104401
2971
    
```


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

SEQ 0060

2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027

.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 ADDRESS 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.

..--

007366

BGNTST

```

3028 007366
3029 007366 004737 004550
3030 007372 005001
3031 007374 005002
3032
3033 007376
3034 007376 104404
3035
3036
3037
3038
3039
3040
3041
3042
3043 007400 052737 000005 002252
3044 007406 042737 000060 002256
3045 007414 004737 005346
3046 007420 001405
3047 007422
3048 007422 104455
3049 007424 000001
3050 007426 000000
3051 007430 003624
3052 007432
3053 007432 104406
3054
3055
3056
3057
3058 007434 042737 000001 002252 2$:
3059 007442 004737 005346
3060 007446 001405
3061 007450
3062 007450 104455
3063 007452 000001
3064 007454 000000
3065 007456 003624
3066 007460
3067 007460 104406
3068
3069
3070
3071
3072 007462 012737 000010 002264 3$:
3073 007470 050237 002264
3074 007474 012737 177540 002270
3075 007502 004737 005414
3076 007506 001405
3077 007510
3078 007510 104455
3079 007512 000002
3080 007514 000000
3081 007516 003712
3082 007520
3083 007520 104406

T12::
JSR PC,INITMS ;SELECT AND INIT MEMORY SIMULATOR
CLR R1 ;CLEAR BITS 15-8 MSAD ADDRESS COUNTER
CLR R2 ;CLEAR BITS 17-16 MSAD ADDRESS COUNTER

1$:
BGNSEG
TRAP CSBSEG

;SET SIGNALS RST H AND MP H IN CONTROL REGISTER 0. THE SIGNAL
;RST H WILL PRESET THE RDV AND WRV FLIP-FLOPS. THE SIGNAL MP H
;WILL ALLOW THE MAP PROTECTION RAM BITS MPIN H AND WRE H TO BE
;READ INTO CONTROL REGISTER 2 AS ESR H AND WREN H RESPECTIVELY.
;THESE BITS WILL NOT BE CHECKED UNTIL THE RAM LOCATION HAS BEEN
;WRITTEN AND TESTED.

BIS #RSTH,MPH,ROLOAD ;SETUP TO SET RST H AND MP H
BIC #RDVH,WRVH,ROMASK ;SETUP TO CHECK RDV AND WRV FOR A 0
JSR PC,LDRDRO ;GO LOAD, READ AND CHECK REG 0
BEQ 2$ ;IF OK THEN GO CLEAR SIGNAL RST H
ERRDF 1,,ROEROR ;REGISTER 0 NOT EQUAL EXPECTED
TRAP CSERDF
.WORD 1
.WORD 0
.WORD ROEROR
CKLOOP
TRAP CSCLP1

;CLEAR BIT RST H IN CONTROL REGISTER 0. RDV AND WRV SHOULD NOT
;CHANGE DURING THIS TEST.

BIC #RSTH,ROLOAD ;SETUP TO CLEAR BIT RST H
JSR PC,LDRDRO ;GO LOAD, READ AND CHECK REG 0
BEQ 3$ ;IF OK THEN CONTINUE
ERRDF 1,,ROEROR ;REGISTER 0 NOT EQUAL EXPECTED
TRAP CSERDF
.WORD 1
.WORD 0
.WORD ROEROR
CKLOOP
TRAP CSCLP1

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

MOV #MSEL1,R2LOAD ;SETUP TO SET MSEL1 BIT TO A 1
BIS R2,R2LOAD ;SETUP MSAD BITS 17 AND 16
MOV #177540,R2MASK ;SETUP TO IGNORE REG 2 BITS 15-8 6-5
JSR PC,LDRDR2 ;GO LOAD, READ AND COMPARE REG 2
BEQ 4$ ;IF LOADED OK THEN CONTINUE
ERRDF 2,,R2EROR ;REGISTER 2 FAILED TO LOAD CORRECTLY
TRAP CSERDF
.WORD 2
.WORD 0
.WORD R2EROR
CKLOOP
TRAP CSCLP1

```

```

3084
3085
3086
3087 007522 010137 002276 4$: MOV R1,R4LOAD ;LOAD MSAD BITS 15-8 INTO CONTROL REGISTER 4
3088 007526 004737 005462 JSR PC,LDRDR4 ;SETUP BITS TO LOAD
3089 007532 001405 BEQ 5$ ;GO LOAD, READ AND COMPARE REG 4
3090 007534 ERRDF 3,R4EROR ;IF REGISTER 4 CORRECT THEN CONT
3091 007534 10' ' ' S TRAP C$ERDF ;REGISTER 4 NOT EQUAL EXPECTED
3092 007536 00C003 .WORD 3
3093 007540 000000 .WORD 0
3094 007542 004000 .WORD R4EROR
3095 007544 CKLOOP
3096 007544 104406 TRAP C$CLP1
3097
3098 ;SET BITS MPIN H AND RDE H TO A 1 - SET BITS WRE H AND MUTB H TO A 0.
3099 ;THESE BITS ARE SET IN LOCATION ADDRESSED BY CONTROL REGISTER 2 AND 4.
3100 ;THE SIGNAL SMPM L WILL BE ASSERTED ON A WRITE OR READ OF CONTROL
3101 ;REGISTER 6.
3102
3103 007546 012737 177760 002306 5$: MOV #177760,R6MASK ;SETUP REG 6 MASK WORD
3104 007554 012737 000005 002302 MOV #MPINH!RDEH,R6LOAD
3105 007562 004737 005506 JSR PC,LDRDR6 ;GO LOAD, READ AND COMPARE REGISTER 6
3106 007566 001405 BEQ 6$ ;IF DATA OK THEN CONTINUE
3107 007570 ERRDF 4,MSGMP,ALINFO ;MAP PROTECTION RAM DATA ERROR
3108 007570 104455 TRAP C$ERDF
3109 007572 000004 .WORD 4
3110 007574 002356 .WORD MSGMP
3111 007576 004052 .WORD ALINFO
3112 007600 CKLOUP
3113 007600 104406 TRAP C$CLP1
3114
3115 ;SETUP TO READ REGISTER 2 WITH REGISTER 0 BIT MP H SET TO A 1 TO
3116 ;ALLOW MPIN H AND WRE H TO BE READ AS BITS ESR H AND WREN H
3117
3118 007602 042737 000140 002270 6$: BIC #ESRH!WRENH,R2MASK ;SETUP TO CHECK ESR H AND WREN H
3119 007610 052737 000040 002266 BIS #ESRH,R2GOOD ;SETUP EXPECTED BIT TO BE SET (MPIN H)
3120 007616 004737 005430 JSR PC,READR2 ;GO READ AND CHECK REGISTER 2
3121 007622 001405 BEQ 7$ ;IF BIT ESR H SET THEN CONTINUE
3122 007624 ERRDF 2,MSGMPL,ALR2IN ;REG 2 NOT EQUAL EXPECTED
3123 007624 104455 TRAP C$ERDF
3124 007626 000002 .WORD 2
3125 007630 002463 .WORD MSGMPL
3126 007632 004126 .WORD ALR2IN
3127 007634 CKLOOP
3128 007634 104406 TRAP C$CLP1
3129
3130 ;WRITE THE SAME LOCATION WITH WRE H AND MUTB H SET TO A 1 AND
3131 ;MPIN H AND RED H SSET TO A 0
3132
3133 007636 012737 000012 002302 7$: MOV #WREH!MUTBH,R6LOAD ;SET WRE H AND MUTB H TO A 1
3134 007644 004737 005506 JSR PC,LDRDR6 ;GO LOAD,READ AND COMPARE REG 6
3135 007650 001405 BEQ 8$ ;IF OK THEN CONTINUE
3136 007652 ERRDF 4,MSGMP,ALINFO ;DATA ERROR MAP PROTECTION RAM
3137 007652 104455 TRAP C$ERDF
3138 007654 000004 .WORD 4
3139 007656 002356 .WORD MSGMP

```

```
3140 007660 004052          .WORD      .WFO
3141 007662                CKLOUP
3142 007662 104406          TRAP      C$CLP1
3143
3144                          ;SETUP TO CHECK THAT MAP PROTECTION BIT WREN H WILL SET THE SIGNAL
3145                          ;WREN H IN CONTROL REGISTER 2 WHEN MP H BIT IS SET.
3146
3147 007664 042737 000140 002266 8$:  BIC      #ESRH,WRENH,R2GOOD      ;CLEAR BOTH EXPECTED BITS
3148 007672 052737 000100 002266      BIS      #WRENH,R2GOOD      ;SETUP TO EXPECT WREN H BIT TO BE SET
3149 007700 004737 005430              JSR      PC,READR2          ;GO READ AND CHECK REGISTER 2
3150 007704 001404              BEQ      9$                  ;IF OK THEN CONTINUE
3151 007706              ERRDF  2 MSGMPL,ALR2IN      ;REGISTER 2 NOT EQUAL EXPECTED
3152 007706 104455          TRAP      'SERDF
3153 007710 000002          .WORD      2
3154 007712 002463          .WORD      MSGMPL
3155 007714 004126          .WORD      ALR2IN
3156 007716              9$:  ENDSEG
3157 007716              10000$:
3158 007716 104405          TRAP      C$ESEG
3159
3160 007720 062701 000400          ADD      #MSAD8,R1          ;UPDATE MSAD BITS 15-8 BY 1
3161 007724 001224          BNE      1$                  ;IF NOT 0 THEN DO NEXT ADDRESS
3162 007726 062702 000001          ADD      #MSAD16,R2        ;UPDATE MSAD BITS 16 AND 17
3163 007732 032702 000004          BIT      #MSELO,R2        ;CHECK IF DONE
3164 007736 001617          BEQ      1$                  ;IF NOT DO NEXT RANGE OF ADDRESSES
3165 007740          ENDTST
3166 007740              L10041:
3167 007740 104401          TRAP      C$ETST
3168
```

3169
 3170
 3171
 3172
 3173
 3174
 3175
 3176
 3177
 3178
 3179
 3180
 3181 007742
 3182 007742
 3183 007742 004737 004550
 3184 007746
 3185 007746
 3186 007746 104402
 3187 007750 005001
 3188 007752 005002
 3189 007754
 3190 007754 104404
 3191
 3192
 3193
 3194 007756 012737 000010 002264
 3195 007764 050237 002264
 3196 007770 012737 177740 002270
 3197 007776 004737 005414
 3198 010002 001405
 3199 010004
 3200 010004 104455
 3201 010006 000002
 3202 010010 000000
 3203 010012 003712
 3204 010014
 3205 010014 104406
 3206
 3207
 3208
 3209 010016 010137 002276
 3210 010022 004737 005462
 3211 010026 001405
 3212 010030
 3213 010030 104455
 3214 010032 000003
 3215 010034 000000
 3216 010036 004000
 3217 010040
 3218 010040 104406
 3219
 3220
 3221
 3222
 3223
 3224 010042 012737 177760 002306

```
.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.
:--

T13:: BGNTST
      JSR    PC,INITMS          ;SELECT AND INIT THE MEMORY SIMULATOR
      BGNSUB

T13.1: TRAP   C$BSUB
      CLR   R1                  ;CLEAR MSAD ADDRESS BITS 15-0
      CLR   R2                  ;CLEAR MSAD ADDRESS BITS 17-16

1$:   BGNSEG
      TRAP  C$BSEG
      ;SET SIGNAL MSEL1 H TO A 1 AND MSAD BITS 17 AND/OR 16 TO A 1 OR 0 IN
      ;CONTROL REGISTER 2

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

      ;LOAD MSAD BITS 15-8 INTO CONTROL REGISTER 2

2$:   MOV   R1,R4LOAD           ;SETUP ADDRESS BITS TO LGAD
      JSR   PC,LDRDR4          ;GO LOAD, READ AND COMPARE REG 4
      BEQ   3$                 ;IF LOADED OK THEN CONT
      ERRDF 3,,R4EROR          ;REGISTER 4 FAILED TO LOAD CORRECTLY
      TRAP  C$ERDF
      .WORD 3
      .WORD 0
      .WORD R4EROR
      CKLOOP
      TRAP  C$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.

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

```

3225 010050 005037 002302          CLR      R6LOAD          ;SET WORD TO LOAD TO 0
3226 010054 004737 005506          JSR      PC,LDRDR6      ;GO LOAD, READ AND COMPARE REG 6
3227 010060 001404                   BEQ      4$              ;IF ALL 0'S THEN CONTINUE
3228 010062                   ERRDF   4,MSGMP,ALINFO  ;MAP PROTECTION RAM DATA ERROR
3229 010062 104455          TRAP    C$ERDF
3230 010064 000004          .WORD   4
3231 010066 002356          .WORD   MSGMP
3232 010070 004052          .WORD   ALINFO
3233 010072                   4$: ENDSEG
3234 010072                   10000$:
3235 010072 104405          TRAP    C$ESEG
3236 010074 062701 000400          ADD     #MSAD8,R1      ;UPDATE MSAD BITS 15-8 TO NEXT ADDRESS
3237 010100 001325          BNE     1$              ;IF NOT 0 THEN CONTINUE
3238 010102 062702 000001          ADD     #MSAD16,R2     ;UPDATE ADDRESS BITS 16 AND 17
3239 010106 032702 000004          BIT     #MSEL0,R2     ;CHECK IF ALL ADDRESSES DONE
3240 010112 001720          BEQ     1$              ;IF NOT DO NEXT ADDRESS RANGE
3241 010114
3242 010114                   L10043:
3243 010114 104403          TRAP    C$ESUB
3244 010116                   BGNSUB
3245 010116                   T13.2:
3246 010116 104402          TRAP    C$BSUB
3247 010120 005001          CLR     R1              ;RESET THE ADDRESS POINTERS
3248 010122 005002          CLR     R2
3249
3250
3251
3252
3253
3254 010124                   1$:
3255 010124 104404          BGNSEG
3256
3257
3258
3259 010126 012737 000010 002264          MOV     #MSEL1,R2LOAD  ;SETUP TO SET MSEL1 H TO A 1
3260 010134 050237 002264          BIS     R2,R2LOAD      ;SETUP BITS MSAD 17-16
3261 010140 012737 177740 002270          MOV     #177740,R2MASK ;SETUP REGISTER 2 MASK WORD
3262 010146 004737 005414          JSR     PC,LDRDR2      ;GO LOAD, READ AND COMPARE REG 2
3263 010152 001405          BEQ     2$              ;IF LOADED OK THEN CONTINUE
3264 010154                   ERRDF   2,,R2EROR      ;REGISTER 2 FAILED TO LOAD CORRECTLY
3265 010154 104455          TRAP    C$ERDF
3266 010156 000002          .WORD   2
3267 010160 000000          .WORD   0
3268 010162 003712          .WORD   R2EROR
3269 010164
3270 010164 104406          CK OOP
3271
3272
3273
3274 010166 010137 002276                   2$:
3275 010172 004737 005462          MOV     R1,R4LOAD      ;SETUP ADDRESS BITS 15-8
3276 010176 001405          JSR     PC,LDRDR4      ;GO LOAD, READ AND COMPARE REGISTER 4
3277 010200                   BEQ     3$              ;IF OK THEN CONTINUE
3278 010200 104455          ERRDF   3,,R4EROR      ;REGISTER 4 NOT EQUAL EXPECTED
3279 010202 000003          TRAP    C$ERDF
3280 010204 000000          .WORD   3
                   .WORD   0

```

```

3281 010206 004000          .WORD  R4EROR
3282 010210          CKLOOP
3283 010210 104406          TRAP   C$CLP1
3284
3285          :SETUP TO READ MAP PROTECTION RAM FOR ZEROES WHICH WERE WRITTEN IN
3286          :THE PREVIOUS SUBST. IF AN ERROR OCCURS, THEN WRITING A PREVIOUS
3287          :LOCATION CHANGED THE LOCATION BEING READ. THIS ERROR IS PROBABLY
3288          :INTERNAL TO RAM CHIP. THE SIGNAL SMPM L WILL BE ASSFRTED DURING
3289          :A READ OF REGISTER 6
3290
3291 010212 012737 177760 002306 3$: MOV    #177760,R6MASK          ;SETUP REG 6 MASK WORD
3292 010220 005037 002302          CLR    R6LOAD              ;SETUP DATA PATTERN THAT WAS LOADED
3293 010224 005037 002304          CLR    R6GOOD             ;SETUP EXPECTED DATA PATTERN
3294 010230 004737 005522          JSR    PC,READR6          ;GO READ REG 6 AND CHECK DATA FOR 0'S
3295 010234 001404          BEQ    4$                  ;IF ALL ZEROES THEN CONTINUE
3296 010236          ERRDF  4,MSGMPS,ALINFO      ;ADDRESS SHORT IN MAP PROTECTION RAM
3297 010236 104455          TRAP   C$ERDF
3298 010240 000004          .WORD  4
3299 010242 002417          .WORD  MSGMPS
3300 010244 004052          .WORD  ALINFO
3301 010246          4$: ENDSEG
3302 010246          10000$:
3303 010246 104405          TRAP   C$ESEG
3304
3305 010250          BGNSEG
3306 010250 104404          TRAP   C$BSEG
3307
3308          :THE FOLLOING SECTION OF CODE WILL WRITE THE LOCATION JUST READ WITH
3309          :ONES AND THEN READ AND CHECK THE LOCATION FOR ONFS. IF THERE IS
3310          :AN INTERNAL ADDRESS SHORT IN THE RAM THIS WILL CAUSE ANOTHER LOCATION
3311          :TO BE WRITTEN AT THE SAME TIME.
3312 010252 012737 000017 002302 MOV    #MUTBH!MPINH!WREH!RDEH,R6LOAD ;SET ALL BITS TO ONES
3313 010260 012737 177760 002306 MOV    #177760,R6MASK          ;SETUP MASK WORD
3314 010266 004737 005506          JSR    PC,LDRDR6          ;GO LOAD, READ AND COMPARE REG 6
3315 010272 001404          BEQ    5$                  ;IF ALL ONES THEN CGNT
3316 010274          ERRDF  4,MSGMP,ALINFO      ;DATA ERROR IN MAP PROTECTION RAM
3317 010274 104455          TRAP   C$ERDF
3318 010276 000004          .WORD  4
3319 010300 002356          .WORD  MSGMP
3320 010302 004052          .WORD  ALINFO
3321 010304          5$: ENDSEG
3322 010304          10001$:
3323 010304 104405          TRAP   C$ESEG
3324 010306 062701 000400          ADD    #MSAD8,R1          ;UPDATE MSAD ADDRESS 15-8
3325 010312 001304          BNE    1$                  ;IF NOT 0 THEN DO NEXT ADDRESS
3326 010314 062702 000001          ADD    #MSAD16,R2         ;UPDATE ADDRESS BITS 17-16
3327 010320 032702 000004          BIT    #MSEL0,R2         ;CHECK IF DONE
3328 010324 001677          BEQ    1$                  ;IF DOT DO NEXT HIGHER RANGE
3329 010326          ENDSUB
3330
3331 010326          L10044:
3332 010326 104403          TRAP   C$ESUB
3333 010330          ENDTST
3334 010330          L10042:
3334 010330 104401          TRAP   C$ETST

```



```

3391 010446 001405      BEQ      3$                ;IF OK THEN CONTINUE
3392 010450              ERRDF    2,,R2EROR        ;REG 2 NOT EQUAL EXPECTED
3393 010450 104455      TRAP    C$ERDF
3394 010452 000002      .WORD   2
3395 010454 000000      .WORD   0
3396 010456 003712      .WORD  R2EROR
3397 010460              CKLOOP
3398 010460 104406      TRAP    C$CLP1
3399
3400              ;LOAD MSAD BIT 15 THROUGH 8 TO A 0 TO SELECT ADDRESS 0
3401
3402 010462 005037 002276 3$: CLR      R4LOAD          ;SETUP TO SET ALL BITS TO 0
3403 010466 004737 005462 JSR     PC,LDRDR4        ;GO LOAD,READ AND CHECK REG 4
3404 010472 001405      BEQ     4$                ;IF NO ERRORS THEN CONTINUE
3405 010474              ERRDF    3,,R4EROR        ;REGISTER 4 NOT ALL ZEROES
3406 010474 104455      TRAP    C$ERDF
3407 010476 000003      .WORD   3
3408 010500 000000      .WORD   0
3409 010502 004000      .WORD  R4EROR
3410 010504              CKLOOP
3411 010504 104406      TRAP    C$CLP1
3412
3413              ;SET BITS MPIN H, RDE H AND MUTB H TO A ONE AND BIT WRE H TO A 0.
3414              ;WRE H ON A 0 WILL SET THE WRV FLIP-FLOP WHEN BIT CK H IS TOGGLED
3415              ;IN CONTROL REGISTER 0.
3416
3417 010506 012737 000015 002302 4$: MOV     #MPINH!RDEH!MUTBH,R6LOAD ;SETUP TO LOAD THE MAP PROTECT RAM
3418 010514 012737 177760 002306 MOV     #177760,R6MASK    ;SETUP REGISTER 6 MASK WORD
3419 010522 004737 005506 JSR     PC,LDRDR6        ;GO LOAD,READ AND CHECK REGISTER 6
3420 010526 001405      BEQ     5$                ;OF COMPARED OK THEN CONTINUE
3421 010530              ERRDF    4,MSGMP,ALINFO    ;MAP PROTECT RAM DATA ERROR
3422 010530 104455      TRAP    C$ERDF
3423 010532 000004      .WORD   4
3424 010534 002356      .WORD  MSGMP
3425 010536 004052      .WORD  ALINFO
3426 010540              CKLOOP
3427 010540 104406      TRAP    C$CLP1
3428
3429              ;CHECK CONTROL REGISTER 0 TO MAKE SURE THAT NO CHANGES OCCURED.
3430
3431 010542 004737 005362 5$: JSR     PC,READR0        ;READ AND CHECK REGISTER 0
3432 010546 001405      BEQ     6$                ;IF OK THEN CONTINUE
3433 010550              ERRDF    1,MSGMPL,ALROIN    ;REGISTER 0 CHANGED STATES
3434 010550 104455      TRAP    C$ERDF
3435 010552 000001      .WORD   1
3436 010554 002463      .WORD  MSGMPL
3437 010556 004100      .WORD  ALROIN
3438 010560              CKLOOP
3439 010560 104406      TRAP    C$CLP1
3440
3441              ;SETUP TO READ CONTROL REGISTER 2 WITH MP H SET TO A 1 IN CONTROL
3442              ;REGISTER 0 TO ALLOW MPIN H, WRE H AND BRK L TO BE READ INTO
3443              ;CONTROL REGISTER 2 AS BITS ESR H, WREN H, AND MSBRK H.
3444
3445 010562 042737 000140 002270 6$: BIC     #ESRH!WRENH,R2MASK ;CLEAR BITS TO BE CHECKED
3446 010570 052737 000040 002266 BIS     #ESRH,R2GOOD     ;SETUP EXPECTED MAP PROTECT BIT

```

```

3447 010576 004737 005430 JSR PC,READR2 ;GO READ REGISTER 2 AND CHECK IT
3448 010602 001405 BEQ 7$ ;IF OK THEN CONTINUE
3449 010604 ERRDF 2,MSGMPL,ALR2IN ;REGISTER 2 NOT EQUAL EXPECTED
3450 010604 104455 TRAP C$ERDF
3451 010606 000002 .WORD 2
3452 010610 002463 .WORD MSGMPL
3453 010612 004126 .WORD ALR2IN
3454 010614 CKLOOP
3455 010614 104406 TRAP C$CLP1
3456
3457 ;SET SIGNAL CK H IN CONTROL REGISTER 0 TO CLOCK THE WRV AND RDV
3458 ;F/F'S. THE WRV F/F SHOULD BE SET (0) AND THE RDV F/F SHOULD BE CLEARED (1).
3459
3460 010616 052737 000140 002254 7$: BIS #CKH!WRVH,ROGOOD ;SETUP EXPECTED DATA
3461 010624 052737 000100 002252 BIS #CKI',ROLOAD ;SETUP BIT TO BE LOADED
3462 010632 004737 005354 JSR PC,LDRDOR ;GO LOAD, READ AND CHECK REG 0
3463 010636 001405 BEQ 8$ ;IF OK THEN CONTINUE
3464 010640 ERRDF 1,MSGMPL,ALROIN ;WRV F/F PROBABLY NOT SET (0)
3465 010640 104455 TRAP C$ERDF
3466 010642 000001 .WORD 1
3467 010644 002463 .WORD MSGMPL
3468 010646 004100 .WORD ALROIN
3469 010650 CKLOOP
3470 010650 104406 TRAP C$CLP1
3471
3472 ;CLEAR THE SIGNAL CK H IN CONTROL REGISTER 0. NO OTHER BITS SHOULD CHANGE.
3473
3474 010652 042737 000100 002254 8$: BIC #CKH,ROGOOD ;CLEAR CKH IN EXPECTED BITS
3475 010660 042737 000100 002252 BIC #CKH,ROLOAD ;SETUP TO CLEAR BIT CK H IN REG 0
3476 010666 004737 005354 JSR PC,LDRDOR ;GO CLEAR, LOAD AND CHECK REG 0
3477 010672 001405 BEQ 9$ ;IF OK THEN CONTINUE
3478 010674 ERRDF 1,MSGMPL,ALROIN ;REG 0 NOT EQUAL EXPECTED
3479 010674 104455 TRAP C$ERDF
3480 010676 000001 .WORD 1
3481 010700 002463 .WORD MSGMPL
3482 010702 004100 .WORD ALROIN
3483 010704 CKLOOP
3484 010704 104406 TRAP C$CLP1
3485
3486 ;CHECK THAT SIGNAL MSBRK H IS SET TO A 1 IN CONTROL REGISTER 2
3487
3488 010706 052737 000240 002266 9$: BIS #MSBRKH!ESRH,R2GOOD ;SETUP EXPECTED BITS IN REGISTER 2
3489 010714 004737 005430 JSR PC,READR2 ;GO READ REGISTER 2 AND CHECK IT
3490 010720 001405 BEQ 10$ ;IF OK THEN CONTINUE
3491 010722 ERRDF 2,MSGMPL,ALR2IN ;MSBRK H PROBABLY NOT SET VIA BRK L BEING LOW
3492 010722 104455 TRAP C$ERDF
3493 010724 000002 .WORD 2
3494 010726 002463 .WORD MSGMPL
3495 010730 004126 .WORD ALR2IN
3496 010732 CKLOOP
3497 010732 104406 TRAP C$CLP1
3498
3499 ;THE FOLLOWING SECTION WILL CHECK THAT THE WRV FLIP-FLOP CAN NOT
3500 ;BE CLEARED (1) BY TRYING TO CLOCK A ONE INTO IT. ONCE THE WRV FLIP-
3501 ;FLOP IS SET (0), IT IS LATCHED TO THAT STATE UNTIL A PULSE IS ISSUED
3502 ;ON THE SIGNALS RST H OR INIT H.

```

```

3503
3504 010734 012737 000017 002302 10$: MOV #MPINH!WREH!RDEH!MUTBH,R6LOAD ;SETUP TO SET ALL RAM BITS TO 1
3505 010742 004737 005506 JSR PC,LDRDR6 ;GO LOAD, READ AND CHECK RAM
3506 010746 001405 BEQ 11$ ;IF ALL ONES THEN CONTINUE
3507 010750 ERRDF 4,MSGMP,ALINFO ;MAP PROTECT RAM DATA ERROR
3508 010750 104455 TRAP C$ERDF
3509 010752 000004 .WORD 4
3510 010754 002356 .WORD MSGMP
3511 010756 004052 .WORD ALINFO
3512 010760 CKLOOP
3513 010760 104406 TRAP C$CLP1
3514
3515 ;CHECK THAT CONTROL REGISTER 0 DID NOT CHANGE STATE.
3516
3517 010762 004737 005362 11$: JSR PC,READR0 ;GO READ AND CHECK REGISTER 0
3518 010766 001405 BEQ 12$ ;IF NO CHANGES THEN CONTINUE
3519 010770 ERRDF 1,MSGMPL,ALROIN ;REGISTER 0 CHANGED STATE
3520 010770 104455 TRAP C$ERDF
3521 010772 000001 .WORD 1
3522 010774 002463 .WORD MSGMPL
3523 010776 004100 .WORD ALROIN
3524 011000 CKLOOP
3525 011000 104406 TRAP C$CLP1
3526
3527 ;CHECK CONTROL REGISTER 2 TO HAVE ESR H, WREN H AND MSBRK H SET TO 1
3528
3529 011002 052737 000340 002266 12$: BIS #ESRH!WRENH!MSBRKH,R2GOOD ;ADD WREN H TO EXPECTED DATA
3530 011010 004737 005430 JSR PC,READR2 ;GO READ AND CHECK REGISTER 2
3531 011014 001405 BEQ 13$ ;IF NO ERRORS THEN CONTINUE
3532 011016 ERRDF 2,MSGMPL,ALR2IN ;REGISTER 2 NOT EQUAL EXPECTED
3533 011016 104455 TRAP C$ERDF
3534 011020 000002 .WORD 2
3535 011022 002463 .WORD MSGMPL
3536 011024 004126 .WORD ALR2IN
3537 011026 CKLOOP
3538 011026 104406 TRAP C$CLP1
3539
3540 ;SET THE SIGNAL CK H TO THE HIGH STATE TO CLOCK THE WRV AND RDV FLIP-
3541 ;FLOPS. THE WRV FLIP-FLOP SHOULD REMAIN SET (0) AS A RESULT OF IT BEING
3542 ;SET ALREADY. THE WRV F.F. FLIP-FLOP, ONCE SET, CAN NOT BE CLOCKED AGAIN
3543 ;UNLESS IT HAS BEEN PRESET BY A PULSE ON THE SIGNALS RST H OR INIT H.
3544
3545 011030 052737 000100 002252 13$: BIS #CKH,ROLOAD ;SETUP TO SET SIGNAL CK H TO HIGH STATE
3546 011036 052737 000100 002254 BIS #CKH,ROGOOD ;EXPECT CK H TO BE SET IN CONTROL REG 0
3547 011044 004737 005354 JSR PC,LDRDOR ;GO LOAD, READ AND CHECK CONTROL REG 0
3548 011050 001405 BEQ 14$ ;IF OK THEN CONTINUE
3549 011052 ERRDF 1,MSGMPL,ALROIN ;WRV F/F CLEARED (1) AFTER BEING LATCHED
3550 011052 104455 TRAP C$ERDF
3551 011054 000001 .WORD 1
3552 011056 002463 .WORD MSGMPL
3553 011060 004100 .WORD ALROIN
3554 011062 CKLOOP
3555 011062 104406 TRAP C$CLP1
3556
3557 ;READ CONTROL REG 2 TO CHECK THAT ESR H, WREN H AND MSBRK H ARE STILL SET (1).
3558

```

```

3559 011064 004737 005430      14$: JSR      PC,READR2      ;READ AND CHECK CONTROL REGISTER 2
3560 011070 001405              BEQ      15$             ;IF OK THEN CONTINUE
3561 011072              ERRDF    2,MSGMPL,ALR2IN ;REGISTER 2 NOT EQUAL EXPECTED
3562 011072 104455              TRAP    C$ERDF
3563 011074 000002              .WORD   2
3564 011076 002463              .WORD   MSGMPL
3565 011100 004126              .WORD   ALR2IN
3566 011102              CKLOOP
3567 011102 104406              TRAP    C$CLP1
3568
3569              ;CLEAR THE SIGNAL CK H AND SET THE SIGNAL RST H IN CONTROL REGISTER 0.
3570              ;SETTING THE SIGNAL RST H TO A ONE WILL CAUSE THE WRV FLIP-FLOP TO BE
3571              ;SET IN ITS PRESET STATE (1).
3572
3573 011104 112737 000005 002252 15$: MOVB    #RSTH!MPH,ROLOAD ;SET RST H AND CLEAR CK H
3574 011112 004737 005346              JSR      PC,LDRDR0      ;GO LOAD, READ AND CHECK CONTROL REG, 0
3575 011116 001405              BEQ      16$             ;IF LOADED OK THEN CONTINUE
3576 011120              ERRDF    1,MSGMPL,ALROIN ;WRV F/F PROBABLY NOT CLEARED BY RST H
3577 011120 104455              TRAP    C$ERDF
3578 011122 000001              .WORD   1
3579 011124 002463              .WORD   MSGMPL
3580 011126 004100              .WORD   ALROIN
3581 011130              CKLOOP
3582 011130 104406              TRAP    C$CLP1
3583
3584              ;CLEAR THE SIGNAL RST H IN CONTROL REGISTER 0
3585
3586 011132 042737 000001 002252 16$: BIC      #RSTH,ROLOAD    ;SETUP TO CLEAR RST H
3587 011140 004737 005346              JSR      PC,LDRDR0      ;GO LOAD, READ AND CHECK CONTROL REG 0
3588 011144 001405              BEQ      17$             ;IF LOADED OK THEN CONTINUE
3589 011146              ERRDF    1,MSGMPL,ALROIN ;CONTROL REG 0 NOT EQUAL EXPECTED
3590 011146 104455              TRAP    C$ERDF
3591 011150 000001              .WORD   1
3592 011152 002463              .WORD   MSGMPL
3593 011154 004100              .WORD   ALROIN
3594 011156              CKLOOP
3595 011156 104406              TRAP    C$CLP1
3596
3597              ;CHECK CONTROL REGISTER 2 TO MAKE SURE THAT THE SIGNAL MSBRK H WENT TO
3598              ;A ZERO AS A RESULT OF THE WRV AND RDV FLIP-FLOPS BEING PRESET BY RST H.
3599
3600 011160 042737 000200 002266 17$: BIC      #MSBRKH,R2GOOD  ;CLEAR MSBRK H IN EXPECTED DATA
3601 011166 004737 005430              JSR      PC,READR2      ;READ AND CHECK CONTROL REGISTER 2
3602 011172 001404              BEQ      18$             ;IF MSBRK H EQUALS A 0 THEN CONTINUE
3603 011174              ERRDF    2,MSGMPL,ALR2IN ;CONTROL REGISTER 2 NOT EQUAL EXPECTED
3604 011174 104455              TRAP    C$ERDF
3605 011176 000002              .WORD   2
3606 011200 002463              .WORD   MSGMPL
3607 011202 004126              .WORD   ALR2IN
3608 011204              18$: ENDSEG
3609 011204              10000$:
3610 011204 104405              TRAP    C$ESEG
3611 011206              ENDTST
3612 011206              L10045:
3613 011206 104401              TRAP    C$ETST
3614

```

3615
 3616
 3617
 3618
 3619
 3620
 3621
 3622
 3623
 3624
 3625
 3626
 3627
 3628
 3629
 3630
 3631
 3632
 3633
 3634
 3635
 3636
 3637
 3638
 3639
 3640
 3641
 3642
 3643
 3644
 3645
 3646
 3647
 3648
 3649
 3650
 3651
 3652
 3653
 3654
 3655
 3656
 3657
 3658
 3659
 3660
 3661
 3662
 3663
 3664
 3665
 3666
 3667
 3668
 3669
 3670

011210
 011210
 011210 004737 004550
 011214
 011214 104404
 011216 105037 002252
 011222 052737 000005 002252
 011230 042737 000060 002256
 011236 004737 005346
 011242 001405
 011244
 011244 104455
 011246 000001
 011250 000000
 011252 003624
 011254
 011254 104406
 011256 042737 000001 002252 1\$:
 011264 004737 005346
 011270 001405
 011272
 011272 104455
 011274 000001
 011276 000000
 011300 003624
 011302
 011302 104406
 011304 012737 000010 002264 2\$:
 011312 012737 177540 002270
 011320 004737 005414

```

.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.
:--

T15:: BGNTST
JSR PC,INITMS ;SELECT AND INIT THE MEMORY SIMULATOR
BGNSEG
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 CSR H, WREN H AND MSBRK H
;IN CONTROL REGISTER 2.

CLRB ROLOAD ;CLEAR LOWER BYTE OF REG 0 FOR LOOPING
BIS #RSTH,MPH,ROLOAD ;SETUP TO SET RST H AND MP H
BIC #WRVH,RDVH,ROMASK ;SETUP TO CHECK WRV AND RDV F/F
JSR PC,LDRDRO ;GO LOAD, READ AND CHECK REG 0
BEQ 1$ ;IF OK THEN CONTINUE
ERRDF 1,ROEROR ;REGISTER 0 NOT EQUAL EXPECTED
TRAP C$ERDF
.WORD 1
.WORD 0
.WORD ROEROR
CKLOOP
TRAP C$CLP1

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

BIC #RSTH,ROLOAD ;SETUP TO CLEAR RST H IN REG 0
JSR PC,LDRDRO ;GO LOAD, READ AND CHECK REG 0
BEQ 2$ ;IF EQUAL THEN CONTINUE
ERRDF 1,ROEROR ;REGISTER NOT EQUAL EXPECTED
TRAP C$ERDF
.WORD 1
.WORD 0
.WORD ROEROR
CKLOOP
TRAP C$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

MOV #MSEL1,R2LOAD ;SETUP TO SET MSEL1 H TO A 1
MOV #177540,R2MASK ;SETUP TO READ MSBRK H AND LOWER 5 BITS
JSR PC,LDRDR2 ;GO LOAD, READ AND CHECK REG 2
  
```

3671	011324	001405				BEQ	3\$:IF OK THEN CONTINUE
3672	011326					ERRDF	2,R2EROR		:REG 2 NOT EQUAL EXPECTED
3673	011326	104455				TRAP	C\$ERDF		
3674	011330	000002				.WORD	2		
3675	011332	000000				.WORD	0		
3676	011334	003712				.WORD	R2EROR		
3677	011336					CKLOOP			
3678	011336	104406				TRAP	C\$CLP1		
3679									
3680									:LOAD MSAD BIT 15 THROUGH 8 TO A 0 TO SELECT ADDRESS 0
3681									
3682	011340	005037	002276		3\$:	CLR	R4LOAD		:SETUP TO SET ALL BITS TO 0
3683	011344	004737	005462			JSR	PC,LDRDR4		:GO LOAD,READ AND CHECK REG 4
3684	011350	001405				BEQ	4\$:IF NO ERRORS THEN CONTINUE
3685	011352					ERRDF	3,R4EROR		:REGISTER 4 NOT ALL ZEROES
3686	011352	104455				TRAP	C\$ERDF		
3687	011354	000003				.WORD	3		
3688	011356	000000				.WORD	0		
3689	011360	004000				.WORD	R4EROR		
3690	011362					CKLOOP			
3691	011362	104406				TRAP	C\$CLP1		
3692									
3693									:SET BITS MPIN H, WRE H AND MUTB H TO A ONE AND BIT RDE H TO A 0.
3694									:RDE H ON A 0 WILL SET THE RDV FLIP-FLOP WHEN BIT CK H IS TOGGLED
3695									:IN CONTROL REGISTER 0.
3696									
3697	011364	012737	000013	002302	4\$:	MOV	#MPINH!WREH!MUTBH,R6LOAD		:SETUP TO LOAD THE MAP PROTECT RAM
3698	011372	012737	177760	002306		MOV	#177760,R6MASK		:SETUP REGISTER 6 MASK WORD
3699	011400	004737	005506			JSR	PC,LDRDR6		:GO LOAD,READ AND CHECK REGISTER 6
3700	011404	001405				BEQ	5\$:OF COMPARED OK THEN CONTINUE
3701	011406					ERRDF	4,MSGMP,ALINFO		:MAP PROTECT RAM DATA ERROR
3702	011406	104455				TRAP	C\$ERDF		
3703	011410	000004				.WORD	4		
3704	011412	002356				.WORD	MSGMP		
3705	011414	004052				.WORD	ALINFO		
3706	011416					CKLOOP			
3707	011416	104406				TRAP	C\$CLP1		
3708									
3709									:CHECK CONTROL REGISTER 0 TO MAKE SURE THAT NO CHANGES OCCUR.
3710									
3711	011420	004737	005362		5\$:	JSR	PC,READR0		:READ AND CHECK REGISTER 0
3712	011424	001405				BEQ	6\$:IF OK THEN CONTINUE
3713	011426					ERRDF	1,MSGMPL,ALROIN		:REGISTER 0 CHANGED STATES
3714	011426	104455				TRAP	C\$ERDF		
3715	011430	000001				.WORD	1		
3716	011432	002463				.WORD	MSGMPL		
3717	011434	004100				.WORD	ALROIN		
3718	011436					CKLOOP			
3719	011436	104406				TRAP	C\$CLP1		
3720									
3721									:SETUP TO READ CONTROL REGISTER 2 WITH MP H SET TO A 1 IN CONTROL
3722									:REGISTER 0 TO ALLOW MPIN H, WRE H AND BRK L TO BE READ INTO
3723									:CONTROL REGISTER 2 AS BITS ESR H, WREN H, AND MSBRK H.
3724									
3725	011440	042737	000140	002270	6\$:	BIC	#ESRH!WRENH,R2MASK		:CLEAR BITS TO BE CHECKED
3726	011446	052737	000140	002266		BIS	#ESRH!WRENH,R2GOOD		:SETUP EXPECTED MAP PROTECT BIT

```

3727 011454 004737 005430 JSR PC,READR2 ;GO READ REGISTER 2 AND CHECK IT
3728 011460 001405 BEQ 7$ ;IF OK THEN CONTINUE
3729 011462 ERRDF 2,MSGMPL,ALR2IN ;REGISTER 2 NOT EQUAL EXPECTED
3730 011462 104455 TRAP C$ERDF
3731 011464 000002 .WORD 2
3732 011466 002463 .WORD MSGMPL
3733 011470 004126 .WORD ALR2IN
3734 011472 CKLOOP
3735 011472 104406 TRAP C$CLP1
3736
3737 ;SET SIGNAL CK H IN CONTROL REGISTER 0 TO CLOCK THE WRV AND RDV
3738 ;F/F'S. THE RDV F/F SHOULD BE SET (0) AND THE WRV F/F SHOULD BE CLEARED (1).
3739
3740 011474 052737 000120 002254 7$: BIS #CKH!RDVH,ROGOOD ;SETUP EXPECTED DATA
3741 011502 052737 000100 002252 BIS #CKH,ROLOAD ;SETUP BIT TO BE LOADED
3742 011510 004737 005354 JSR PC,LDRDOR ;GO LOAD, READ AND CHECK REG 0
3743 011514 001405 BEQ 8$ ;IF OK THEN CONTINUE
3744 011516 ERRDF 1,MSGMPL,ALROIN ;RDV F/F PROBABLY NOT SET (0)
3745 011516 104455 TRAP C$ERDF
3746 011520 000001 .WORD 1
3747 011522 002463 .WORD MSGMPL
3748 011524 004100 .WORD ALROIN
3749 011526 CKLOOP
3750 011526 104406 TRAP C$CLP1
3751
3752 ;CLEAR THE SIGNAL CK H IN CONTROL REG 0. NO OTHER BITS SHOULD CHANGE STATE.
3753
3754 011530 042737 000100 002254 8$: BIC #CKH,ROGOOD ;CLEAR CKH IN EXPECTED BITS
3755 011536 042737 000100 002252 BIC #CKH,ROLOAD ;SETUP TO CLEAR BIT CK H IN REG 0
3756 011544 004737 005354 JSR PC,LDRDOR ;GO CLEAR, LOAD AND CHECK REG 0
3757 011550 001405 BEQ 9$ ;IF OK THEN CONTINUE
3758 011552 ERRDF 1,MSGMPL,ALROIN ;REG 0 NOT EQUAL EXPECTED
3759 011552 104455 TRAP C$ERDF
3760 011554 000001 .WORD 1
3761 011556 002463 .WORD MSGMPL
3762 011560 004100 .WORD ALROIN
3763 011562 CKLOOP
3764 011562 104406 TRAP C$CLP1
3765
3766 ;CHECK THAT SIGNAL MSBRK H IS SET TO A 1 IN CONTROL REGISTER 2
3767
3768 011564 052737 000340 002266 9$: BIS #MSBRKH!WRENH!ESRH,R2GOOD ;SETUP EXPECTED BITS IN REGISTER 2
3769 011572 004737 005430 JSR PC,READR2 ;GO READ REGISTER 2 AND CHECK IT
3770 011576 001405 BEQ 10$ ;IF OK THEN CONTINUE
3771 011600 ERRDF 2,MSGMPL,ALR2IN ;MSBRK H PROBABLY NOT SET VIA BRK L BEING LOW
3772 011600 104455 TRAP C$ERDF
3773 011602 000002 .WORD 2
3774 011604 002463 .WORD MSGMPL
3775 011606 004126 .WORD ALR2IN
3776 011610 CKLOOP
3777 011610 104406 TRAP C$CLP1
3778
3779 ;THE FOLLOWING SECTION WILL CHECK THAT THE RDV FLIP-FLOP CAN NOT BE
3780 ;CLEARED (1) BY TRYING TO CLOCK A ONE INTO IT. ONCE THE RDV FLIP-FLOP
3781 ;IS SET (0), IT IS LATCHED TO THAT STATE UNTIL A PULSE IS ISSUED ON THE
3782 ;SIGNAL RST H OR INIT H.
    
```

```

3783
3784 011612 012737 000017 002302 10$: MOV #MPINH!WREN!RDEH!MUTBH,R6LOAD ;SETUP TO SET ALL RAM BITS TO 1
3785 011620 004737 005506 JSR PC,LDRDR6 ;GO LOAD, READ AND CHECK RAM
3786 011624 001405 BEQ 11$ ;IF ALL ONES THEN CONTINUE
3787 011626 ERRDF 4,MSGMP,ALINFO ;MAP PROTECT RAM DATA ERROR
3788 011626 104455 TRAP C$ERDF
3789 011630 000004 .WORD 4
3790 011632 002356 .WORD MSGMP
3791 011634 004052 .WORD ALINFO
3792 011636 CKLOOP
3793 011636 104406 TRAP C$CLP1
3794
3795 ;CHECK THAT CONTROL REGISTER 0 DID NOT CHANGE STATE.
3796
3797 011640 004737 005362 11$: JSR PC,READR0 ;GO READ AND CHECK REGISTER 0
3798 011644 001405 BEQ 12$ ;IF NO CHANGES THEN CONTINUE
3799 011646 ERRDF 1,MSGMPL,ALROIN ;REGISTER 0 CHANGED STATE
3800 011646 104455 TRAP C$ERDF
3801 011650 000001 .WORD 1
3802 011652 002463 .WORD MSGMPL
3803 011654 004100 .WORD ALROIN
3804 011656 CKLOOP
3805 011656 104406 TRAP C$CLP1
3806
3807 ;CHECK CONTROL REGISTER 2 TO HAVE ESR H, WREN H AND MSBRK H SET TO 1
3808
3809 011660 052737 000340 002266 12$: BIS #ESRH!WRENH!MSBRKH,R2GOOD ;SETUP EXPECTED DATA
3810 011666 004737 005430 JSR PC,READR2 ;GO READ AND CHECK REGISTER 2
3811 011672 001405 BEQ 13$ ;IF NO ERRORS THEN CONTINUE
3812 011674 ERRDF 2,MSGMPL,ALR2IN ;REGISTER 2 NOT EQUAL EXPECTED
3813 011674 104455 TRAP C$ERDF
3814 011676 000002 .WORD 2
3815 011700 002463 .WORD MSGMPL
3816 011702 004126 .WORD ALR2IN
3817 011704 CKLOOP
3818 011704 104406 TRAP C$CLP1
3819
3820 ;SET THE SIGNAL CK H TO THE HIGH STATE TO CLOCK THE WRV AND RDV FLIP-
3821 ;FLOPS. THE RDV FLIP-FLOP SHOULD REMAIN SET (0) AS A RESULT OF IT BEING
3822 ;SET ALREADY. THE RDV FLIP-FLOP, ONCE SET, CAN NOT BE CLOCKED AGAIN
3823 ;UNLESS IT HAS BEEN PRESET BY A PULSE ON THE SIGNALS RST H OR INIT H.
3824
3825 011706 052737 000100 002252 13$: BIS #CKH,ROLOAD ;SETUP TO SET SIGNAL CK H TO HIGH STATE
3826 011714 052737 000100 002254 BIS #CKH,ROGOOD ;EXPECT CK H TO BE SET IN CONTROL REG 0
3827 011722 004737 005354 JSR PC,LDRDOR ;GO LOAD, READ AND CHECK CONTROL REG 0
3828 011726 001405 BEQ 14$ ;IF OK THEN CONTINUE
3829 011730 ERRDF 1,MSGMPL,ALROIN ;RDV F/F CLEARED (1) AFTER BEING LATCHED
3830 011730 104455 TRAP C$ERDF
3831 011732 000001 .WORD 1
3832 011734 002463 .WORD MSGMPL
3833 011736 004100 .WORD ALROIN
3834 011740 CKLOOP
3835 011740 104406 TRAP C$CLP1
3836
3837 ;READ CONTROL REG 2 TO CHECK THAT ESR H, WREN H AND MSBRK H ARE STILL SET TO 1'S
3838

```



```

3839 011742 004737 005430      14$: JSR      PC,READR2      ;READ AND CHECK CONTROL REGISTER 2
3840 011746 001405              BEQ      15$            ;IF OK THEN CONTINUE
3841 011750              ERRDF    2,MSGMPL,ALR2IN ;REGISTER 2 NOT EQUAL EXPECTED
3842 011750 104455              TRAP    C$ERDF
3843 011752 000002              .WORD   2
3844 011754 002463              .WORD   MSGMPL
3845 011756 004126              .WORD   ALR2IN
3846 011760              CKLOOP
3847 011760 104406              TRAP    C$CLP1
3848
3849              ;CLEAR THE SIGNAL CK H AND SET THE SIGNAL RST H IN CONTROL REGISTER 0.
3850              ;SETTING THE SIGNAL RST H TO A ONE WILL CAUSE THE RDV FLIP-FLOP TO BE
3851              ;SET IN ITS PRESET STATE (1).
3852
3853 011762 112737 000005 002252 15$: MOVB    #RSTH!MPH,ROLOAD ;SET RST H AND CLEAR CK H
3854 011770 004737 005346              JSR      PC,LDRDRO     ;GO LOAD, READ AND CHECK CONTROL REG 0
3855 011774 001405              BEQ      16$            ;IF LOADED OK THEN CONTINUE
3856 011776              ERRDF    1,MSGMPL,ALROIN ;RDV F/F PROBABLY NOT CLEARED BY RST H
3857 011776 104455              TRAP    C$ERDF
3858 012000 000001              .WORD   1
3859 012002 002463              .WORD   MSGMPL
3860 012004 004100              .WORD   ALROIN
3861 012006              CKLOOP
3862 012006 104406              TRAP    C$CLP1
3863
3864              ;CLEAR THE SIGNAL RST H IN CONTROL REGISTER 0
3865
3866 012010 042737 000001 002252 16$: BIC      #RSTH,ROLOAD   ;SETUP TO CLEAR RST H
3867 012016 004737 005346              JSR      PC,LDRDRO     ;GO LOAD, READ AND CHECK CONTROL REG 0
3868 012022 001405              BEQ      17$            ;IF LOADED OK THEN CONTINUE
3869 012024              ERRDF    1,MSGMPL,ALROIN ;CONTROL REG 0 NOT EQUAL EXPECTED
3870 012024 104455              TRAP    C$ERDF
3871 012026 000001              .WORD   1
3872 012030 002463              .WORD   MSGMPL
3873 012032 004100              .WORD   ALROIN
3874 012034              CKLOOP
3875 012034 104406              TRAP    C$CLP1
3876
3877              ;CHECK CONTROL REGISTER 2 TO MAKE SURE THAT THE SIGNAL MSBRK H WENT TO
3878              ;A ZERO AS A RESULT OF THE WRV AND RDV FLIP-FLOPS BEING PRESET BY RST H.
3879
3880 012036 042737 000200 002266 17$: BIC      #MSBRKH,R2GOOD ;CLEAR MSBRK H IN EXPECTED DATA
3881 012044 004737 005430              JSR      PC,READR2     ;READ AND CHECK CONTROL REGISTER 2
3882 012050 001404              BEQ      18$            ;IF MSBRK H EQUALS A 0 THEN CONTINUE
3883 012052              ERRDF    2,MSGMPL,ALR2IN ;CONTROL REGISTER 2 NOT EQUAL EXPECTED
3884 012052 104455              TRAP    C$ERDF
3885 012054 000002              .WORD   2
3886 012056 002463              .WORD   MSGMPL
3887 012060 004126              .WORD   ALR2IN
3888 012062              18$: ENDSEG
3889 012062              10000$:
3890 012062 104405              TRAP    C$ESEG
3891 012064              ENDTST
3892 012064              L10046:
3893 012064 104401              TRAP    C$ETST
3894

```

3895
 3896
 3897
 3898
 3899
 3900
 3901
 3902
 3903
 3904
 3905 012066
 3906 012066
 3907 012066 004737 004550
 3908 012072
 3909 012072 104404
 3910
 3911
 3912
 3913
 3914
 3915
 3916
 3917
 3918 012074 105037 002252
 3919 012100 052737 000005 002252
 3920 012106 042737 000060 002256
 3921 012114 004737 005346
 3922 012120 001405
 3923 012122
 3924 012122 104455
 3925 012124 000001
 3926 012126 000000
 3927 012130 003624
 3928 012132
 3929 012132 104406
 3930
 3931
 3932
 3933
 3934 012134 042737 000001 002252 1\$:
 3935 012142 004737 005346
 3936 012146 001405
 3937 012150
 3938 012150 104455
 3939 012152 000001
 3940 012154 000000
 3941 012156 003624
 3942 012160
 3943 012160 104406
 3944
 3945
 3946
 3947
 3948 012162 012737 000010 002264 2\$:
 3949 012170 012737 177540 002270
 3950 012176 004737 005414

.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.
:--
  
```

```

T16:: BGNTST
      JSR   PC,INITMS           ;SELECT AND INIT THE MEMORY SIMULATOR
      BGNSEG
      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-FLOPS 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.
  
```

```

      CLRB  R0LOAD              ;CLEAR LOWER BYTE OF REG 0 FOR LOOPING
      BIS   #RSTH!MPH,R0LOAD    ;SETUP TO SET RST H AND MP H
      BIC   #WRVH!RDVH,R0MASK   ;SETUP TO CHECK WRV AND RDV F/F
      JSR   PC,LDRDRO           ;GO LOAD, READ AND CHECK REG 0
      BEQ   1$                  ;IF OK THEN CONTINUE
      ERDF  1,,ROEROR          ;REGISTER 0 NOT EQUAL EXPECTED
      TRAP  C$ERDF
      .WORD 1
      .WORD 0
      .WORD ROEROR
      CKLOOP
      TRAP  C$CLP1
  
```

```

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

```

      BIC   #RSTH,R0LOAD        ;SETUP TO CLEAR RST H IN REG 0
      JSR   PC,LDRDRO           ;GO LOAD, READ AND CHECK REG 0
      BEQ   2$                  ;IF EQUAL THEN CONTINUE
      ERDF  1,,ROEROR          ;REGISTER NOT EQUAL EXPECTED
      TRAP  C$ERDF
      .WORD 1
      .WORD 0
      .WORD ROEROR
      CKLOOP
      TRAP  C$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
  
```

```

      MOV   #MSEL1,R2LOAD       ;SETUP TO SET MSEL1 H TO A 1
      MOV   #177540,R2MASK      ;SETUP TO READ MSBRK H AND LOWER 5 BITS
      JSR   PC,LDRDR2           ;GO LOAD, READ AND CHECK REG 2
  
```

```

3951 012202 001405          BEQ      3$                ;IF OK THEN CONTINUE
3952 012204                ERRDF    2,,R2EROR        ;REG 2 NOT EQUAL EXPECTED
3953 012204 104455          TRAP    C$ERDF
3954 012206 000002          .WORD   2
3955 012210 000000          .WORD   0
3956 012212 003712          .WORD   R2EROR
3957 012214                CKLOOP
3958 012214 104406          TRAP    C$CLP1
3959
3960                ;LOAD MSAD BIT 15 THROUGH 8 TO A 0 TO SELECT ADDRESS 0
3961
3962 012216 005.37 002276      3$:   CLR      R4LOAD          ;SETUP TO SET ALL BITS TO 0
3963 012222 004737 005462      JSR     PC,LDRDR4        ;GO LOAD,READ AND CHECK REG 4
3964 012226 001405          BEQ     4$                ;IF NO ERRORS THEN CONTINUE
3965 012230                ERRDF    3,,R4EROR        ;REGISTER 4 NOT ALL ZEROES
3966 012230 104455          TRAP    C$ERDF
3967 012232 000003          .WORD   3
3968 012234 000000          .WORD   0
3969 012236 004000          .WORD   R4EROR
3970 012240                CKLOOP
3971 012240 104406          TRAP    C$CLP1
3972
3973                ;SET BITS MPIN H AND MUTB H TO A ONE AND BITS WRE H AND RDE H TO
3974 012242                ;A 0. RDE H AND WRE H ON A 0 WILL SET THE RDV AND WRV FLIP-FLOPS WHEN
3975 012244                ;THE SIGNAL CK H IS TOGGLED IN CONTROL REGISTER 0.
3976
3977 012242 012737 000011 002302 4$:   MOV     #MPINH!MUTBH,R6LOAD ;SETUP TO LOAD THE MAP PROTECT RAM
3978 012250 012737 177760 002306      MOV     #177760,R6MASK    ;SETUP REGISTER 6 MASK WORD
3979 012256 004.37 005506      JSR     PC,LDRDR6        ;GO LOAD,READ AND CHECK REGISTER 6
3980 012262 001405          BEQ     5$                ;IF COMPARED OK THEN CONTINUE
3981 012264                ERRDF    4,MSGMP,ALINFO    ;MAP PROTECT RAM DATA ERROR
3982 012264 104455          TRAP    C$ERDF
3983 012266 000004          .WORD   4
3984 012270 002356          .WORD   MSGMP
3985 012272 004052          .WORD   ALINFO
3986 012274                CKLOOP
3987 012274 104406          TRAP    C$CLP1
3988
3989                ;CHECK CONTROL REGISTER 0 TO MAKE SURE THAT NO CHANGES OCCURED.
3990
3991 012276 004737 005362      5$:   JSR     PC,READR0        ;READ AND CHECK REGISTER 0
3992 012302 001405          BEQ     6$                ;IF OK THEN CONTINUE
3993 012304                ERRDF    1,MSGMPL,ALROIN    ;REGISTER 0 CHANGED STATES
3994 012304 104455          TRAP    C$ERDF
3995 012306 000001          .WORD   1
3996 012310 002463          .WORD   MSGMPL
3997 012312 004100          .WORD   ALROIN
3998 012314                CKLOOP
3999 012314 104406          TRAP    C$CLP
4000
4001                ;SETUP TO READ CONTROL REGISTER 2 WITH MP H SET TO A 1 IN CONTROL
4002                ;REGISTER 0 TO ALLOW MPIN H, WRE H AND BRK L TO BE READ INTO
4003                ;CONTROL REGISTER 2 AS BITS ESR H, WRFV H, AND MSBRK H.
4004
4005 012316 042737 000140 002270 6$:   BIC     #ESRH!WRENH,R2MASK ;CLEAR BITS TO BE CHECKED
4006 012324 052737 000040 002266      BIS     #ESRH,R2GOOD     ;SETUP EXPECTED MAP PROTECT BIT
  
```

```

4007 012332 004737 005430 JSR PC,READR2 ;GO READ REGISTER 2 AND CHECK IT
4008 012336 001405 BEQ 7$ ;IF OK THEN CONTINUE
4009 012340 ERRDF 2,MSGMPL,ALR2IN ;REGISTER 2 NOT EQUAL EXPECTED
4010 012340 104455 TRAP C$ERDF
4011 012342 000002 .WORD 2
4012 012344 002463 .WORD MSGMPL
4013 012346 004176 .WORD ALR2IN
4014 012350 CKLOOP
4015 012350 104406 TRAP C$CLP1
4016
4017
4018 ;SET SIGNAL CK H IN CONTROL REGISTER 0 TO CLOCK THE WRV AND RDV
4019 ;FLIP-FLOPS. THE RDV AND WRV FLIP-FLOPS SHOULD BE SET TO A 1.
4020 012352 052737 000160 002254 7$: BIS #CKH!RDVH!WRVH,ROGOOD ;SETUP EXPECTED DATA
4021 012360 052737 000100 002252 BIS #CKH,ROLOAD ;SETUP BIT TO BE LOADED
4022 012366 004737 005354 JSR PC,LDRDOR ;GO LOAD, READ AND CHECK REG 0
4023 012372 001405 BEQ 8$ ;IF OK THEN CONTINUE
4024 012374 ERRDF 1,MSGMPL,ALROIN ;REGISTER 0 NOT EQUAL EXPECTED
4025 012374 104455 TRAP C$ERDF
4026 012376 000001 .WORD 1
4027 012400 002463 .WORD MSGMPL
4028 012402 004100 .WORD ALROIN
4029 012404 CKLOOP
4030 012404 104406 TRAP C$CLP1
4031
4032 ;CLEAR THE SIGNAL CK H IN CONTROL REGISTER 0. NO OTHER BITS SHOULD
4033 ;CHANGE STATE.
4034
4035 012406 042737 000100 002254 8$: BIC #CKH,ROGOOD ;CLEAR CKH IN EXPECTED BITS
4036 012414 042737 000100 002252 BIC #CKH,ROLOAD ;SETUP TO CLEAR BIT CK H IN REG 0
4037 012422 004737 005354 JSR PC,LDRDOR ;GO CLEAR, LOAD AND CHECK REG 0
4038 012426 001405 BEQ 9$ ;IF OK THEN CONTINUE
4039 012430 ERRDF 1,MSGMPL,ALROIN ;REG 0 NOT EQUAL EXPECTED
4040 012430 104455 TRAP C$ERDF
4041 012432 000001 .WORD 1
4042 012434 002463 .WORD MSGMPL
4043 012436 004100 .WORD ALROIN
4044 012440 CKLOOP
4045 012440 104406 TRAP C$CLP1
4046
4047 ;CHECK THAT SIGNAL MSBRK H IS SET TO A 1 IN CONTROL REGISTER 2
4048
4049 012442 052737 000240 002266 9$: BIS #MSBRKH!ESRH,R2GOOD ;SETUP EXPECTED BIT IN REGISTER 2
4050 012450 004737 005430 JSR PC,READR2 ;GO READ REGISTER 2 AND CHECK IT
4051 012454 001405 BEQ 10$ ;IF OK THEN CONTINUE
4052 012456 ERRDF 2,MSGMPL,ALR2IN ;REGISTER 2 NOT EQUAL EXPECTED
4053 012456 104455 TRAP C$ERDF
4054 012460 000002 .WORD 2
4055 012462 002463 .WORD MSGMPL
4056 012464 004126 .WORD ALR2IN
4057 012466 CKLOOP
4058 012466 104406 TRAP C$CLP1
4059
4060 ;THE FOLLOWING SECTION WILL CHECK THAT THE SIGNAL RST H WILL CLEAR
4061 ;THE RDV AND WRV FLIP-FLOPS.
4062

```

```

4063 012470 052737 000005 002252 10$: BIS #RSTH!MPH,ROLOAD ;SETUP TO SET THE SIGNAL RST H
4064 012476 004737 005346 JSR PC,LDRDRO ;GO LOAD, READ AND CHECK REG 0
4065 012502 001405 BEQ 11$ ;IF REG 0 OK THEN CONTINUE
4066 012504 ERRDF 1,MSGMPL,ALROIN ;RDV AND WRV PROBABLY NOT CLEARED
4067 012504 104455 TRAP C$ERDF
4068 012506 000001 .WORD 1
4069 012510 002463 .WORD MSGMPL
4070 012512 004100 .WORD ALROIN
4071 012514 CKLOOP
4072 012514 104406 TRAP C$CLP1
4073
4074 ;CLEAR THE SIGNAL RST H IN CONTROL REGISTER 0
4075
4076 012516 042737 000001 002252 11$: BIC #RSTH,ROLOAD ;SETUP TO CLEAR SIGNAL RST H
4077 012524 004737 005346 JSR PC,LDRDRO ;GO LOAD,READ AND CHECK REG 0
4078 012530 001405 BEQ 12$ ;IF OK THEN CONTINUE
4079 012532 ERRDF 1,MSGMPL,ALROIN ;REGISTER 0 NET EQUAL EXPECTED
4080 012532 104455 TRAP C$ERDF
4081 012534 000001 .WORD 1
4082 012536 002463 .WORD MSGMPL
4083 012540 004100 .WORD ALROIN
4084 012542 CKLOOP
4085 012542 104406 TRAP C$CLP1
4086
4087 ;READ REGISTER 2 AND CHECK THAT THE SIGNAL MSBRK H WENT TO A 0
4088
4089 012544 042737 000200 002266 12$: BIC #MSBRKH,R2GOOD ;CLEAR EXPECTED BIT IN REG 2
4090 012552 004737 005430 JSR PC,READR2 ;GO READ AND CHECK REGISTER 2
4091 012556 001404 BEQ 13$ ;IF OK THEN CONTINUE
4092 012560 ERRDF 2,MSGMPL,ALR2IN ;REGISTER 2 NOT EQUAL EXPECTED
4093 012560 104455 TRAP C$ERDF
4094 012562 000002 .WORD 2
4095 012564 002463 .WORD MSGMPL
4096 012566 004126 .WORD ALR2IN
4097 012570 13$: ENDSEG
4098 012570 10000$:
4099 012570 104405 TRAP C$ESEG
4100 012572 ENDTST
4101 012572 L10047:
4102 012572 104401 TRAP C$ETST
4103

```

4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159

012574
012574
012574 004737 004550
012600 005001
012602 005002
012604
012604 104404
012606 012737 000004 002264
012614 012737 177740 002270
012622 004737 005414
012626 001405
012630
012630 104455
012632 000002
012634 000000
012636 003712
012640
012640 104406
012642 010137 002276
012646 004737 005462
012652 001405
012654
012654 104455
012656 000003
012660 000000
012662 004000
012664
012664 104406

```
.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 SMDSO L. THE SIGNAL SMDSO L WILL SELECT MODULE SELECT RAM 0.
:--

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

1$:        BGNSEG
          TRAP     C$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 SMDSO L TO BE ASSERTED ON A WRITE OR READ TO REGISTER 6.
          ;THE SIGNAL SMDSO L WILL SELECT MODULE SELECT RAM 0.

          MOV      #MSEL0,R2LOAD   ;SETUP TO SET MSEL0 H TO A 1
          MOV      #177740,R2MASK  ;SETUP MASK WORD TO IGNORE BITS 15-5
          JSR      PC,LDRDR2      ;GO LOAD, READ AND CHECK REG 2
          BEQ      2$              ;IF LOADED OK THEN CONTINUE
          ERRDF   2,R2EROR        ;REG 2 NOT EQUAL EXPECTED DATA
          TRAP     C$ERDF

          .WORD   2
          .WORD   0
          .WORD   R2EROR
          CKLOOP
          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.

2$:        MOV      R1,R4LOAD      ;SETUP BITS TO BE LOADED
          JSR      PC,LDRDR4      ;GO LOAD, READ AND CHECK REGISTER 4
          BEQ      3$              ;IF LOADED OK THEN CONTINUE
          ERRDF   3,R4EROR        ;REGISTER 4 NOT EQUAL EXPECTED DATA
          TRAP     C$ERDF

          .WORD   3
          .WORD   0
          .WORD   R4EROR
          CKLOOP
          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
```

```

4160 ;PATTERN IS READ BACK. ON A WRITE OR READ COMMAND TO CONTROL
4161 ;REGISTER 6 WITH THE SIGNAL MSEL0 H A 1 AND SIGNAL MSEL1 H A 0, THE
4162 ;SIGNAL SMDS0 L WILL BE ASSERTED WHICH WILL SELECT MODULE SELECT RAM 0.
4163
4164 012666 012737 177760 002306 3$: MOV #177760,R6MASK ;SETUP REGISTER 6 MASK WORD
4165 012674 010237 002302 MOV R2,R6LOAD ;SETUP DATA PATTERN TO BE LOADED
4166 012700 010203 MOV R2,R3 ;COPY DATA TO BE LOADED
4167 012702 032703 000001 BIT #BIT0,R3 ;CHECK IF BIT 0 IS SET
4168 012706 001403 BEQ 4$ ;IF NOT GO CHECK BIT 1
4169 012710 042703 000016 BIC #BIT3!BIT2!BIT1,R3 ;CLEAR MOST SIGNIFICANT BITS
4170 012714 000413 BR 6$ ;GO DO THE TEST
4171 012716 032703 000002 4$: BIT #BIT1,R3 ;CHECK IF BIT 1 IS SET
4172 012722 001403 BEQ 5$ ;IF NOT THEN GO CHECK CHECK BIT 2
4173 012724 042703 000014 BIC #BIT3!BIT2,R3 ;CLEAR MORE SIGNIFICANT BITS
4174 012730 000405 BR 6$ ;GO START THE TEST
4175 012732 032703 000004 5$: BIT #BIT2,R3 ;CHECK IF BIT 2 IS SET
4176 012736 001402 BEQ 6$ ;IF NOT THEN GO DO THE TEST
4177 012740 042703 000010 BIC #BIT3,R3 ;CLEAR MOST SIGNIFICANT BIT
4178 012744 010337 002304 6$: MOV R3,R6GOOD ;SAVE EXPECTED PATTERN
4179 012750 004737 005514 JSR PC,LDRD6R ;GO LOAD, READ, AND CHECK REG 6
4180 012754 001404 BEQ 7$ ;IF DATA OK THEN CONTINUE
4181 012756 ERRDF 4,MSGMDO,ALINFO ;DATA ERROR IN MODULE SELECT RAM 0
4182 012756 104455 TRAP C$ERDF
4183 012760 000004 .WORD 4
4184 012762 002513 .WORD MSGMDO
4185 012764 004052 .WORD ALINFO
4186
4187 012766 7$: ENDSEG
4188 012766 10000$.
4189 012766 104405 TRAP C$ESEG
4190 012770 005202 INC R2 ;UPDATE DATA PATTERN BY 1
4191 012772 022702 000020 CMP #20,R2 ;CHECK IF DONE
4192 012776 001302 BNE 1$ ;IF NOT THEN DO NEXT PATTERN
4193 013000 005002 CLR R2 ;RESET PATTERN TO 0
4194
4195 013002 062701 020000 ADD #MSAD13,R1 ;UPDATE MODULE SELECT RAM ADDRESS BY 1
4196 013006 001276 BNE 1$ ;IF NOT DONE REPEAT THE TEST
4197 013010 ENDTST
4198 013010 L10050:
4199 013010 104401 TRAP C$ETST
4200

```

4201
 4202
 4203
 4204
 4205
 4206
 4207
 4208
 4209
 4210
 4211
 4212
 4213
 4214
 4215
 4216
 4217 013012
 4218 013012
 4219 013012 004737 004550
 4220 013016 012701 000014
 4221 013022 005002
 4222
 4223 013024
 4224 013024 104404
 4225
 4226
 4227
 4228
 4229
 4230
 4231
 4232 013026 010137 002264
 4233 013032 012737 177740 002270
 4234 013040 004737 005414
 4235 013044 001405
 4236 013046
 4237 013046 104455
 4238 013050 000002
 4239 013052 000000
 4240 013054 003712
 4241 013056
 4242 013056 104406
 4243
 4244
 4245
 4246
 4247
 4248
 4249
 4250 013060 012737 177760 002306 28:
 4251 013066 010237 002302
 4252 013072 010203
 4253 013074 032703 000001
 4254 013100 001403
 4255 013102 042703 000016
 4256 013106 000413

.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 SIGNIF!CANT 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.
:--
  
```

T18:: BGNST

```

JSR PC,INITMS ;SELECT AND INIT THE MEMORY SIMULATOR
MOV #MSEL0!MSEL1,R1 ;SETUP BITS FOR CONTROL REGISTER 2
CLR R2 ;SET DATA PATTERN INITIALLY TO 0
  
```

18: BGNSEG
 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.
  
```

```

MOV R1,R2LOAD ;SETUP ADDRESS + SET MSEL0 AND MSEL1 = 1
MOV #177740,R2MASK ;SETUP REGISTER 2 MASK WORD
JSR PC,LDRDR2 ;GO LOAD, READ AND CHECK REGISTER 2
BEQ 2$ ;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 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.
  
```

```

MOV #177760,R6MASK ;SETUP REGISTER 6 MASK WORD
MOV R2,R6LOAD ;SETUP PATTERN TO BE LOADED
MOV R2,R3 ;COPY DATA PATTERN TO BE LOADED
BIT #BIT0,R3 ;CHECK IF BIT 0 IS SET
BEQ 3$ ;IF NOT GO CHECK BIT 1
BIC #BIT3!BIT2!BIT1,R3 ;IF YES - 0 MOST SIGNIFICANT BITS
BR 5$ ;GO LOAD THE DATA PATTERN
  
```


4257	013110	032703	000002	3\$:	BIT	#BIT1,R3		:CHECK IF BIT 1 IS SET
4258	013114	001403			BEQ	4\$:IF NOT THEN CHECK BIT 2
4259	013116	042703	000014		BIC	#BIT3!BIT2,R3		:IF YES - 0 MOST SIGNIFICANT BITS
4260	013122	000405			BR	5\$:GO LOAD THE DATA PATTERN
4261	013124	032703	000004	4\$:	BIT	#BIT2,R3		:CHECK IF BIT 2 IS SET
4262	013130	001402			BEQ	5\$:IF NOT GO LOAD THE DATA PATTERN
4263	013132	042703	000010		BIC	#BIT3,R3		:CLEAR MOST SIGNIFICANT BITS
4264	013136	010337	002304	5\$:	MOV	R3,R6GOOD		:SAVE EXPECTED DATA PATTERN
4265	013142	004737	005514		JSR	PC,LDRD6R		:GO LOAD, READ AND CHECK DATA
4266	013146	001404			BEQ	6\$:IF DATA OK THEN CONTINUE
4267	013150				ERRDF	4,MSGMD1,ALINFO		:DATA ERROR IN MODULE SELECT RAM 1
4268	013150	104455			TRAP	C\$ERDF		
4269	013152	000004			.WORD	4		
4270	013154	002555			.WORD	MSGMD1		
4271	013156	004052			.WORD	ALINFO		
4272	013160			6\$:	ENDSEG			
4273	013160			10000\$:				
4274	013160	104405			TRAP	C\$ESEG		
4275	013162	005202			INC	R2		:UPDATE THE DATA PATTERN
4276	013164	022702	000020		CMP	#20,R2		:CHECK IF BINARY COUNT DONE
4277	013170	001315			BNE	1\$:IF NOT LOAD NEXT PATTERN
4278	013172	005002			CLR	R2		:OTHERWISE CLEAR DATA PATTERN
4279	013174	005201			INC	R1		:UPDATE MSAD BITS 17 AND 16
4280	013176	032701	000020		BIT	#BIT4,R1		:CHECK IF DONE
4281	013202	001710			BEQ	1\$:IF NOT THEN CHECK NEXT ADDRESS
4282	013204				ENDTST			
4283	013204			L10051:				
4284	013204	104401			TRAP	C\$ETST		
4285								

4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341

013206
013206
013206 004737 004550
013212
013212 104404
013214 012737 000004 002264
013222 012737 177740 002270
013230 004737 005414
013234 001405
013236
013236 104455
013240 000002
013242 000000
013244 003712
013246
013246 104406
013250 005037 002276
013254 004737 005462
013260 001405
013262
013262 104455
013264 000003
013266 000000
013270 004000
013272
013272 104406
013274 012737 177760 002306 2\$:

```
.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.
: --
T19:: BGNTST
JSR PC,INITMS ;SELECT AND INIT MEMORY SIMULATOR
BGNSEG
TRAP C$BSEG
;SET SIGNAL MSEL0 H TO A ONE AND SIGNALS MSEL1 H, MSAD17 H AND
;MSAD16 H TO A ZERO IN CONTROL REGISTER 2. THE SIGNAL MSEL0 H ON A
;ONE WILL CAUSE THE SIGNAL SMDS0 L TO BE ASSERTED ON A WRITE OR READ
;TO CONTROL REGISTER 6. THE SIGNAL SMDS0 L WILL SELECT MODULE SELECT
;RAM 0.
MOV #MSEL0,R2LOAD ;SETUP TO SET MSEL0 H TO A 1
MOV #177740,R2MASK ;SETUP REGISTER 2 MASK WORD
JSR PC,LDRDR2 ;GO LOAD, READ AND CHECK REGISTER 2
BEQ 1$ ;IF LOADED OK THEN CONTINUE
ERRDF 2,,R2EROR ;REGISTER 2 NOT EQUAL EXPECTED DATA
TRAP C$ERDF
.WORD 2
.WORD 0
.WORD R2EROR
CKLOOP
TRAP C$CLP1
;CLEAR CONTROL REGISTER 4 TO SET MODULE SELECT ADDRESSES TO 0. MSAD
;BITS 15-13 SELECT MODULE SELECT RAM 0 ADDRESSES.
1$: CLR R4LOAD ;SETUP TO CLEAR MSAD BITS 15-0
JSR PC,LDRDR4 ;GO LOAD, READ AND CHECK REGISTER 4
BEQ 2$ ;IF LOADED OK THEN CONTINUE
ERRDF 3,,R4EROR ;REGISTER 4 NOT EQUAL EXPECTED DATA
TRAP C$ERDF
.WORD 3
.WORD 0
.WORD R4EROR
CKLOOP
TRAP C$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.
2$: MOV #177760,R6MASK ;SETUP REGISTER 6 MASK WORD
```

4342	013302	012737	000002	002302	MOV	#BIT1,R6LOAD	:SETUP DATA PATTERN OF 1	
4343	013310	004737	005506		JSR	PC,LDRDR6	:GO LOAD, READ AND CHECK RAM	
4344	013314	001405			BEQ	3\$:IF DATA OK THEN CONTINUE	
4345	013316				ERRDF	4,MSGMDO,ALINFO	:DATA ERROR IN MODULE SELECT RAM 0	
4346	013316	104455			TRAP	C\$ERDF		
4347	013320	000004			.WORD	4		
4348	013322	002513			.WORD	MSGMDO		
4349	013324	004052			.WORD	ALINFO		
4350	013326				CKLOOP			
4351	013326	104406			TRAP	C\$CLP1		
4352								
4353								
4354								
4355								
4356								
4357								
4358								
4359	013330	012737	000014	002264	3\$:	MOV	#MSELO!MSEL1,R2LOAD	:SETUP MODULE SELECT BITS AND ADDRESS
4360	013336	004737	005414			JSR	PC,LDRDR2	:GO LOAD, READ AND CHECK REG 2
4361	013342	001405				BEQ	4\$:IF LOADED OK THEN CONT
4362	013344					ERRDF	2,,R2EROR	:REG 2 NOT EQUAL EXPECTED DATA
4363	013344	104455				TRAP	C\$ERDF	
4364	013346	000002				.WORD	2	
4365	013350	000000				.WORD	0	
4366	013352	003712				.WORD	R2EROR	
4367	013354					CKLOOP		
4368	013354	104406				TRAP	C\$CLP1	
4369								
4370								
4371								
4372								
4373								
4374								
4375	013356	012737	000010	002302	4\$:	MOV	#BIT3,R6LOAD	:SET DATA PATTERN TO 10
4376	013364	004737	005506			JSR	PC,LDRDR6	:GO LOAD, READ AND CHECK DATA
4377	013370	001405				BEQ	5\$:IF DATA OK THEN CONTINUE
4378	013372					ERRDF	4,MSGMD1,ALINFO	:DATA ERROR IN MODULE SELECT RAM 1
4379	013372	104455				TRAP	C\$ERDF	
4380	013374	000004				.WORD	4	
4381	013376	002555				.WORD	MSGMD1	
4382	013400	004052				.WORD	ALINFO	
4383	013402					CKLOOP		
4384	013402	104406				TRAP	C\$CLP1	
4385								
4386								
4387								
4388								
4389	013404	012737	000004	002264	5\$:	MOV	#MSELO,R2LOAD	:SETUP TO CLEAR BIT MSEL1 H
4390	013412	004737	005414			JSR	PC,LDRDR2	:GO LOAD, READ AND CHECK REG 2
4391	013416	001405				BEQ	6\$:IF LOADED OK THEN CONTINUE
4392	013420					ERRDF	2,,R2EROR	:REG 2 NOT EQUAL EXPECTED DATA
4393	013420	104455				TRAP	C\$ERDF	
4394	013422	000002				.WORD	2	
4395	013424	000000				.WORD	0	
4396	013426	003712				.WORD	R2E OR	
4397	013430					CKLOOP		

```

4398 013430 104406 TRAP C$CLP1
4399
4400 ;READ MODULE SELECT RAM 0 BY READING CONTROL REGISTER 6. ISSUING
4401 ;A READ COMMAND WHEN MSEL0 H IS SET AND MSEL1 H IS CLEARED WILL
4402 ;ASSERT THE SIGNAL SMDS0 L. THE SIGNAL SMDS0 L BEING ASSERTED WILL
4403 ;SELECT MODULE SELECT RAM 0.
4404
4405 013432 012737 000002 002302 6$: MOV #BIT1,R6LOAD ;SETUP PREVIOUSLY LOADED DATA
4406 013440 013737 002302 002304 MOV R6LOAD,R6GOOD ;SETUP EXPECTED DATA PATTERN
4407 013446 004737 005522 JSR PC,READR6 ;GO READ AND CHECK RAM LOCATION
4408 013452 001405 BEQ 7$ ;IF DATA OK THEN CONT
4409 013454 ERRDF 4,MSGMDC,ALINFO ;PROBABLY CHIP ENABLE ERROR - MODULE
4410 013454 104455 TRAP C$ERDF
4411 013456 000004 .WORD 4
4412 013460 002617 .WORD MSGMDC
4413 013462 004052 .WORD ALINFO
4414 013464 CKLOOP
4415 013464 104406 TRAP C$CLP1
4416 ;SELECT RAM
4417
4418 ;RESELECT MODULE SELECT RAM 1 BY SETTING SIGNALS MSEL0 H AND MSEL1 H
4419 ;TO A ONE IN CONTROL REGISTER 2.
4420
4421 013466 012737 000014 002264 7$: MOV #MSEL0!MSEL1,R2LOAD ;SETUP BITS TO BE LOADED
4422 013474 004737 005414 JSR PC,LDRDR2 ;GO LOAD, READ AND CHECK REG 2
4423 013500 001405 BEQ 8$ ;IF LOADED OK THEN CONT
4424 013502 ERRDF 2,R2EROR ;REGISTER 2 NOT EQUAL EXPECTED
4425 013502 104455 TRAP C$ERDF
4426 013504 000002 .WORD 2
4427 013506 000000 .WORD 0
4428 013510 003712 .WORD R2EROR
4429 013512 CKLOOP
4430 013512 104406 TRAP C$CLP1
4431
4432 ;READ MODULE SELECT RAM 1 BY READING CONTROL REGISTER 6. ISSUING
4433 ;A READ COMMAND TO CONTROL REGISTER 6 WITH MSEL0 H AND MSEL1 H SET
4434 ;WILL ASSERT THE SIGNAL SMDS1 L. THIS SIGNAL BEING ASSERTED WILL
4435 ;SELECT MODULE SELECT RAM 1.
4436
4437 013514 012737 000010 002302 8$: MOV #BIT3,R6LOAD ;SETUP PREVIOUSLY LOADED DATA
4438 013522 013737 002302 002304 MOV R6LOAD,R6GOOD ;SETUP EXPECTED DATA
4439 013530 004737 005522 JSR PC,READR6 ;GO READ AND CHECK THE DATA
4440 013534 001404 BEQ 9$ ;IF DATA OK THEN CONT
4441 013536 ERRDF 4,MSGMDC,ALINFO ;PROBABLY CHIP ENABLE ERROR
4442 013536 104455 TRAP C$ERDF
4443 013540 000004 .WORD 4
4444 013542 002617 .WORD MSGMDC
4445 013544 004052 .WORD ALINFO
4446 013546 9$: ENDSEG
4447 013546 10000$:
4448 013546 104405 TRAP C$ESEG
4449 013550 ENDTST
4450 013550 L10052:
4451 013550 104401 TRAP C$ETST
4452

```

```

4453 .SBTTL TEST 20: CHECK EACH ADDRESS IN MOD SEL RAM 0 TO BE ADDRESSED
4454
4455
4456 : **
4457 : TEST TO CHECK THAT EACH ADDRESS IN MODULE SELECT RAM 0 CAN BE ADDRESSED
4458 : CORRECTLY. THIS IS DONE BY WRITING A SPECIFIC PATTERN INTO THE RAM AND THEN
4459 : READING THE RAM CHECKING THAT NO LOCATIONS CHANGED. THE PATTERNS LOADED
4460 : INTO THE RAM, STARTING AT THE LOWEST ADDRESS, ARE AS FOLLOWS: 10, 04, 02, 01,
4461 : 00, 01, 10, 04.
4462 : --
4463 013552          BGNTST
4464 013552          T20::
4465 013552 004737 004550 JSR    PC,INITMS      ;SELECT AND INIT THE MEMORY SIMULATOR
4466 013556 005001      CLR    R1              ;CLEAR WORKING MSAD BITS
4467 013560 005002      CLR    R2              ;CLEAR OFFSET TO DATA TABLE
4468
4469 013562          18:  BGNSEG
4470 013562 104404     TRAP   C$BSEG
4471
4472                ;SET SIGNAL MSEL0 H TO A ONE, AND SIGNALS MSEL1 H, MSAD17 H AND MSAD16 H
4473                ;TO A ZERO IN CONTROL REGISTER 2.
4474
4475 013564 012737 000004 002264 MOV    #MSEL0,R2LOAD   ;SETUP TO SET MSEL0 H TO A 1
4476 013572 012737 177740 002270 MOV    #177740,R2MASK  ;SETUP REGISTER 2 MASK WORD
4477 013600 004737 005414 JSR    PC,LDRDR2      ;GO LOAD,READ AND CHECK REG 2
4478 013604 001405      BEQ    2$              ;IF LOADED OK THEN CONTINUE
4479 013606          ERRDF 2,,R2EROR  ;REGISTER 2 NOT EQUAL EXPECTED
4480 013606 104455     TRAP   C$ERDF
4481 013610 000002     .WORD 2
4482 013612 000000     .WORD 0
4483 013614 003712     .WORD R2EROR
4484 013616          CKLOOP
4485 013616 104406     TRAP   C$CLP1
4486
4487                ;SET MSAD BITS 15-13 IN CONTROL REGISTER 4 TO THE ADDRESS TO BE
4488                ;TESTED. MSAD BITS 15-13 ARE USED TO SELECT MODULE SELECT RAM 0
4489                ;ADDRESSES.
4490
4491 013620 010137 002276 28:  MOV    R1,R4LOAD       ;SETUP BITS TO BE LOADED
4492 013624 004737 005462 JSR    PC,LDRDR4      ;GO LOAD, READ AND CHECK REGISTER 4
4493 013630 001405      BEQ    3$              ;IF LOADED OK THEN CONTINUE
4494 013632          ERRDF 3,,R4EROR  ;REGISTER 4 NOT EQUAL EXPECTED
4495 013632 104455     TRAP   C$ERDF
4496 013634 000003     .WORD 3
4497 013636 000000     .WORD 0
4498 013640 004000     .WORD R4EROR
4499 013642          CKLOOP
4500 013642 104406     TRAP   C$CLP1
4501
4502                ;THE FOLLOWING SECTION OF CODE WILL WRITE THE DATA PATTERN INTO RAM
4503                ;AND CHECK THAT THE CORRECT PATTERN WAS WRITTEN. ON A WRITE OR READ
4504                ;COMMAND TO CONTROL REGISTER 6 WITH SIGNAL MSEL0 H SET AND MSEL1 H
4505                ;CLEARED, THE SIGNAL SMDS0 L WILL BE ASSERTED TO SELECT MODULE SELECT
4506                ;RAM 0.
4507
4508 013644 016237 014040 002302 38:  MOV    MSRODT(R2),R6LOAD ;GET DATA PATTERN FROM TABLE

```

4509	013652	012737	177760	002306	MOV	#177760,R6MASK	:SETUP REGISTER 6 MASK WORD
4510	013660	004737	005506		JSR	PC,LDRDR6	:GO LOAD,READ AND CHECK RAM LOCATION
4511	013664	001404			BEQ	4\$:IF OK THEN CONTINUE
4512	013666				ERRDF	4,MSGMDO,ALINFO	:DATA ERROR IN MODULE SELECT RAM 0
4513	013666	104455			TRAP	C\$ERDF	
4514	013670	000004			.WORD	4	
4515	013672	002513			.WORD	MSGMDO	
4516	013674	004052			.WORD	ALINFO	
4517	013676				ENDSEG		
4518	013676						
4519	013676	104405			TRAP	C\$ESEG	
4520	013700	005722			TST	(R2)+	:UPDATE OFFSET POINTER TO DATA TABLE
4521	013702	062701	020000		ADD	#MSAD13,R1	:UPDATE MODULE SELECT RAM 0 ADDRESS
4522	013706	001325			BNE	1\$:IF NOT 0 THEN DO NEXT ADDRESS
4523							
4524	013710	005001			CLR	R1	:RESET WORKING MSAD ADDRESSES TO 0
4525	013712	005002			CLR	R2	:CLEAR OFFSET POINTER TO DATA TABLE
4526							
4527	013714				BGNSEG		
4528	013714	104404			TRAP	C\$BSEG	
4529							
4530							:SET SIGNAL MSEL0 H TO A ONE AND CLEAR SIGNALS MSEL1 H, MSAD17 H AND
4531							:MSAD16 H. THESE BITS ARE 'N CONTROL REGISTER 2.
4532							
4533	013716	012737	000004	002264	MOV	#MSEL0,R2LOAD	:SETUP TO SET MSEL0 H TO A 1
4534	013724	004737	005414		JSR	PC,LDRDR2	:GO LOAD, READ AND CHECK REG 2
4535	013730	001405			BEQ	6\$:IF LOADED OK THEN CONTINUE
4536	013732				ERRDF	2,,R2EROR	:REGISTER 2 NOT EQUAL EXPECTED
4537	013732	104455			TRAP	C\$ERDF	
4538	013734	000002			.WORD	2	
4539	013736	000000			.WORD	0	
4540	013740	003712			.WORD	R2EROR	
4541	013742				CKLOOP		
4542	013742	104406			TRAP	C\$CLP1	
4543							
4544							:SET MSAD BITS 15-13 IN CONTROL REGISTER 4 TO THE ADDRESS TO BF CHECKED.
4545							
4546	013744	010137	002276		MOV	R1,R4LOAD	:SETUP BITS TO BE LOADED
4547	013750	004737	005462		JSR	PC,LDRDR4	:GO LOAD, READ AND CHECK REGISTER 4
4548	013754	001405			BEQ	7\$:IF LOADED OK THEN CONTINUE
4549	013756				ERRDF	3,,R4EROR	:REGISTER 4 NOT EQUAL EXPECTED
4550	013756	104455			TRAP	C\$ERDF	
4551	013760	000003			.WORD	3	
4552	013762	000000			.WORD	0	
4553	013764	004000			.WORD	R4EROR	
4554	013766				CKLOOP		
4555	013766	104406			TRAP	C\$CLP1	
4556							
4557							:THE FOLLOWING SECTION WILL VALIDATE THAT THE DATA WRITTEN IN THE
4558							:BEGINNING OF THIS TEST REMAINS UNCHANGED. IF AN ERROR OCCURS, THE
4559							:PROBLEM IS PROBABLY INTERNAL TO THE RAM CHIP. ON A READ COMMAND
4560							:TO CONTROL REGISTER 6 WITH THE SIGNAL MSEL0 H SET AND MSEL1 H CLEARED,
4561							:THE SIGNAL SMDS0 L WILL BE ASSERTED TO SELECT MODULE SELECT RAM 0.
4562							
4563	013770	016237	014040	002302	MOV	MSRODT(R2),R6LOAD	:SETUP PREVIOUSLY LOADED DATA
4564	013776	013737	002302	002304	MOV	R6LOAD,R6GOOD	:SETUP EXPECTED DATA

HARDWARE TESTS MACY11 30A(1052) 17-JUN-82 12:02 PAGE 90
CVC DAB.P11 17-JUN-82 11:26

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

SEQ 0090

4565	014004	004737	005522		JSR	PC,READR6		:GO READ AND CHECK RAM LOCATION
4566	014010	001404			BEQ	8\$:IF DATA OK THEN CONTINUE
4567	014012				ERRDF	4,MSGMDA,ALINFO		:MODULE SELECT ADDRESSING ERROR
4568	014012	104455			TRAP	C\$ERDF		
4569	014014	000004			.WORD	4		
4570	014016	002667			.WORD	MSGMDA		
4571	014020	004052			.WORD	ALINFO		
4572	014022			8\$:	ENDSEG			
4573	014022			10001\$:				
4574	014022	104405			TRAP	C\$ESEG		
4575								
4576	014024	005722			TST	(R2)+		:UPDATE OFFSET POINTER TO DATA TABLE
4577	014026	062701	020000		ADD	#MSAD13,R1		:UPDATE MODULE SELECT ADDRESS
4578	014032	001330			BNE	5\$:IF NOT DONE CHECK NEXT ADDRESS
4579	014034				EXIT	TST		
4580	014034	104432			TRAP	C\$EXIT		
4581	014036	000022			.WORD	L10053-		
4582								
4583	014040	000010		MSR00T:	.WORD	10		
4584	014042	000004			.WORD	4		
4585	014044	000002			.WORD	2		
4586	014046	000001			.WORD	1		
4587	014050	000000			.WORD	0		
4588	014052	000001			.WORD	1		
4589	014054	000010			.WORD	10		
4590	014056	000004			.WORD	4		
4591								
4592	014060				ENDTST			
4593	014060			L10053:				
4594	014060	104401			TRAP	C\$ETST		
4595								

4596
 4597
 4598
 4599
 4600
 4601
 4602
 4603
 4604
 4605 014062
 4606 014062
 4607 014062 004737 004550
 4608 014066 012701 000014
 4609 014072 005002
 4610
 4611 014074
 4612 014074 104404
 4613
 4614
 4615
 4616
 4617 014076 010137 002264
 4618 014102 012737 177740 002270
 4619 014110 004737 005414
 4620 014114 001405
 4621 014116
 4622 014116 104455
 4623 014120 000002
 4624 014122 000000
 4625 014124 003712
 4626 014126
 4627 014126 104406
 4628
 4629
 4630
 4631
 4632
 4633
 4634 014130 016237 014304 002302 2\$:
 4635 014136 012737 177760 002306
 4636 014144 004737 005506
 4637 014150 001404
 4638 014152
 4639 014152 104455
 4640 014154 000004
 4641 014156 002555
 4642 014160 004052
 4643 014162
 4644 014162
 4645 014162 104405
 4646
 4647 014164 005722
 4648 014166 005201
 4649 014170 032701 000020
 4650 014174 001737
 4651

```
.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.
:--

T21:: BGNTST
      JSR    PC,INITMS           ;SELECT AND INIT THE MEMORY SIMULATOR
      MOV    #MSELO!MSEL1,R1    ;SETUP BITS FOR CONTROL REGISTER 2
      CLR    R2                  ;CLEAR OFFSET POINTER TO DATA TABLE

1$:   BGNSEG
      TRAP   C$BSEG

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

      MOV    R1,R2LOAD          ;SETUP BITS TO BE LOADED
      MOV    #177740,R2MASK     ;SETUP REGISTER 2 MASK WORD
      JSR    PC,LDRDR2          ;GO LOAD, READ AND CHECK REGISTER 2
      BEQ    2$                  ;IF LOADED OK THEN CONTINUE
      ERRDF  2,,R2EROR          ;REGISTER 2 NOT EQUAL EXPECTED DATA
      TRAP   C$ERDF
      .WORD  2
      .WORD  0
      .WORD  R2EROR
      CKLOOP
      TRAP   C$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 MSELO H AND MSEL1 H SET TO A ONE,
      ;THE SIGNAL SMDS1 L WILL BE ASSERTED TO SELECT MODULE SELECT RAM 1.

2$:   MOV    MSR1DT(R2),R6LOAD   ;GET DATA PATTERN TO BE WRITTEN
      MOV    #177760,R6MASK     ;SETUP MASK WORD FOR REGISTER 6
      JSR    PC,LDRDR6          ;GO LOAD, READ AND CHECK RAM LOCATION
      BEQ    3$                  ;IF DATA OK THEN GO DO NEXT LOCATION
      ERRDF  4,MSGMD1,ALINFO    ;DATA ERROR IN MODULE SELECT RAM 1
      TRAP   C$ERDF
      .WORD  4
      .WORD  MSGMD1
      .WORD  ALINFO

3$:   ENDSEG
10000$: TRAP   C$ESEG

      TST    (R2)+              ;UPDATE THE OFFSET POINTER TO DATA TABLE
      INC    R1                  ;UPDATE MSAD BITS
      BIT    #BIT4,R1           ;CHECK IF RAM COMPLETELY WRITTEN
      BEQ    1$                  ;IF NOT THEN DO NEXT LOCATION
```



```

4652 014176 012701 000014      MOV      #MSEL0!MSEL1,R1      ;RESET BITS FOR CONTROL REGISTER 2
4653 014202 005002              CLR      R2                    ;CLEAR OFFSET POINTER TO DATA TABLE
4654
4655 014204              4$:  BGNSEG
4656 014204 104404              TRAP    C$BSEG
4657
4658              ;SET SIGNALS MSEL0 H AND MSEL1 H TO A ONE. SET MSAD BITS 17 AND 16
4659              ;TO THE ADDRESS TO BE TESTED.
4660
4661 014206 010137 002264      MOV      R1,R2LOAD            ;SETUP BITS TO BE LOADED
4662 014212 004737 005414      JSR      PC,LDRDR2           ;GO LOAD, READ AND CHECK REGISTER 2
4663 014216 001405              BEQ      5$                   ;IF LOADED OK THEN CONTINUE
4664 014220              ERRDF   2,,R2EROR           ;REGISTER 2 NOT EQUAL EXPECTED
4665 014220 104455              TRAP    C$ERDF
4666 014222 000002              .WORD   2
4667 014224 000000              .WORD   0
4668 014226 003712              .WORD   R2EROR
4669 014230              CKLOOP
4670 014230 104406              TRAP    C$CLP1
4671
4672              ;THE FOLLOWING SECTION WILL VALIDATE THAT THE DATA WRITTEN IN THE
4673              ;BEGINNING OF THIS TEST REMAINS UNCHANGED. IF AN ERROR OCCURS, THE
4674              ;PROBLEM IS PROBABLY INTERNAL TO THE RAM CHIP. ON A READ COMMAND
4675              ;TO CONTROL REGISTER 6 WITH THE SIGNALS MSEL0 H AND MSEL1 H SET TO
4676              ;A 1, THE SIGNAL SMD51 L WILL BE ASSERTED TO SELECT MODULE SELECT RAM 1.
4677
4678 014232 016237 014304 002302 5$:  MOV      MSR1DT(R2),R6LOAD    ;SETUP DATA PREVIOUSLY WRITTEN
4679 014240 013737 002302 002304      MOV      R6LOAD,R6GOOD      ;SETUP EXPECTED DATA PATTERN
4680 014246 004737 005522      JSR      PC,READR6          ;GO READ THE RAM LOCATION
4681 014252 001404              BEQ      6$                   ;IF DATA OK THEN CONTINUE
4682 014254              ERRDF   4,MSGMD1,ALINFO     ;MODULE SELECT RAM 1 ADDRESSING ERROR
4683 014254 104455              TRAP    C$ERDF
4684 014256 000004              .WORD   4
4685 014260 002555              .WORD   MSGMD1
4686 014262 004052              .WORD   ALINFO
4687 014264              6$:  ENDSEG
4688 014264              10001$:
4689 014264 104405              TRAP    C$ESEG
4690
4691 014266 005722              TST     (R2)+                ;INCREMENT OFFSET POINTER TO DATA TABLE
4692 014270 005201              INC     R1                    ;UPDATE MSAD BITS 17 AND 16
4693 014272 032701 000020      BIT     #BIT4,R1             ;CHECK IF DONE
4694 014276 001742              BEQ     4$                    ;IF NOT CHECK NEXT ADDRESS
4695 014300              EXIT   TST
4696 014300 104432              TRAP    C$EXIT
4697 014302 000012              .WORD   L10054-.
4698 014304 000001      MSR1DT: .WORD   1
4699 014306 000002              .WORD   2
4700 014310 000004              .WORD   4
4701 014312 000010              .WORD   10
4702 014314              ENDTST
4703 014314              L10054:
4704 014314 104401              TRAP    C$ETST
4705

```

4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761

014316
014316
014316 004737 004550
014322 005037 002276
014326 012701 00000i

014332
014332 104404

014334 012737 000004 002264
014342 012737 177740 002270
014350 004737 005414
014354 001405
014356
014356 104455
014360 000002
014362 000000
014364 003712
014366
014366 104406

014370 004737 005462
014374 001405
014376
014376 104455
014400 000003

.SBTTL TEST 22: CHECK MEM SIM RAM'S WITH 1'S + 0'S, AND 0'S + 1'S.

:++
: TEST TO CHECK THAT ALL MEMORY SIMULATOR RAM'S CAN BE WRITTEN AND READ WITH
: DATA PATTERN'S 125252 AND 052525. THIS TEST CHECKS THAT ALL BITS CAN BE
: WRITTEN TO A ONE AND ZERO AND THAT NO ADJACENT BITS ARE SHORTED TO EACH OTHER.
: THIS TEST IS EXECUTED IN 16 BIT MODE. THIS TEST, HOWEVER, DOES NOT CHECK FOR
: INTERNAL ADDRESS SHORTS, ADDRESS BIT DROP OUT, OR THAT THE CORRECT RAM IS
: SELECTED. THE TEST THAT FOLLOW THIS TEST WILL PERFORM THE ABOVE CHECKS.

: 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

T22:: BGNSTST
JSR PC,INITMS ;SELECT AND INIT MEMORY SIMULATOR
CLR R4LOAD ;SET 1ST ADDRESS TO BE TESTED TO 0
MOV #BIT0,R1 ;SETUP DATA FOR MODULE SELECT RAM 0

1\$: BGNSEG
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.

MOV #MSEL0,R2LOAD ;SETUP BITS TO BE LOADED
MOV #177740,R2MASK ;SETUP REGISTER 2 MASK WORD
JSR PC,LDRDR2 ;GO LOAD,READ AND CHECK REG 2
BEQ 2\$;IF LOADED OK THEN CONT
ERRDF 2,,R2EROR ;REGISTER 2 NOT EQUAL EXPECTED
TRAP C\$ERDF
.WORD 2
.WORD 0
.WORD R2EROR
CKLOOP
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

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

4762 014402 000000
4763 014404 004000
4764 014406
4765 014406 104406
4766

.JORD 0
.WORD R4EROR
CKLOOP
TRAP C\$CLP1

4767
4768
4769
4770
4771
4772

;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 0 AND CHECK THAT IS WAS
;WRITTEN CORRECTLY. THE DATA PATTERNS THAT WILL BE WRITTEN ARE 1, 2, 4,
;10. THESE PATTERNS WILL ENABLE THE SIGNALS EN0 H, EN1 H, EN2 H, AND
;EN3 H RESPECTIVELY. THESE SIGNALS ARE USED TO SELECT MEMORY SIMULATOR
;MEMORY WHEN THE SIGNAL SSM L IS ASSERTED LATER ON IN THE TEST.

4773 014410 010137 002302 3\$:
4774 014414 012737 177760 002306
4775 014422 004737 005506
4776 014426 001405
4777 014430
4778 014430 104455
4779 014432 000004
4780 014434 002513
4781 014436 004052
4782 014440
4783 014440 104406
4784

MOV R1,R6LOAD ;GET THE PATTERN TO BE LOADED
MOV #177760,R6MASK ;SETUP REGISTER 6 MASK WORD
JSR PC,LDRDR6 ;GO LOAD, READ AND CHECK RAM 0
BEQ 4\$;IF OK THEN CONTINUE
ERRDF 4,MSGMDO,ALINFO ;DATA ERROR IN MODULE SELECT RAM 0
TRAP C\$ERDF
.WORD 4
.WORD MSGMDO
.WORD ALINFO
CKLOOP
TRAP C\$CLP1

4785
4786
4787
4788
4789
4790

;SET SIGNALS MSEL0 H AND MSEL1 H TO A ONE AND SIGNALS MSAD17 H AND MSAD16 H
;TO A ZERO IN CONTROL REGISTER 2. MSEL0 H AND MSEL1 H ON A ONE, WILL
;CAUSE THE SIGNAL SMDS1 L TO BE ASSERTED ON A WRITE OR READ TO CONTROL
;REGISTER 6. SMDS1 L WILL SELECT MODULE SELECT RAM 1. MSAD BITS 17 AND 16
;ON A ZERO WILL SELECT ADDRESS 0 OF MODULE SELECT RAM 1.

4791 014442 012737 000014 002264 4\$:
4792 014450 004737 005414
4793 014454 001405
4794 014456
4795 014456 104455
4796 014460 000002
4797 014462 000000
4798 014464 003712
4799 014466
4800 014466 104406
4801

MOV #MSEL0!MSEL1,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

4802
4803
4804
4805
4806
4807

;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 1 AND CHECK THAT THE
;PATTERN WAS LOADED CORRECTLY. A DATA PATTERN OF ALL ONES WILL BE
;WRITTEN, HOWEVER, WHEN THE LOCATION IS READ BACK, A DATA PATTERN
;EQUAL TO ONE WILL BE READ. SETTING THE LEAST SIGNIFICANT BIT IN
;THE DATA PATTERN WILL FORCE THE MORE SIGNIFICANT BITS TO ZERO.

4808 014470 012737 000017 002302 5\$:
4809 014476 012737 000001 002304
4810 014504 004737 005514
4811 014510 001404
4812 014512
4813 014512 104455
4814 014514 000004
4815 014516 002555
4816 014520 004052
4817 014522 6\$:

MOV #BIT3!BIT2!BIT1!BIT0,R6LOAD ;GET DATA PATTERN TO BE LOADED
MOV #BIT0,R6GOOD ;SETUP EXPECTED DATA PATTERN
JSR PC,LDRDR6 ;GO LOAD,READ AND CHECK RAM 1
BEQ 6\$;IF DATA OK THEN CONTINUE
ERRDF 4,MSGMD1,ALINFO ;DATA ERROR IN MUDULE SELECT RAM 1
TRAP C\$ERDF
.WORD 4
.WORD MSGMD1
.WORD ALINFO
ENDSEG

```

4818 014522          10000$:
4819 014522 104405   TRAP    C$ESEG
4820
4821 014524 012702 010000   MOV    #4096.,R2          ;SETUP 4K WORD COUNTER TO DO 1 RAM
4822
4823 014530          7$:   BGNSEG
4824 014530 104404   TRAP    C$BSEG
4825
4826                ;CLEAR BITS MSEL0 H, MSEL1 H, MSAD17 H AND MSAD16 H IN CONTROL REGISTER
4827                ;2. MSEL0 H AND MSEL1 H ON A ZERO WILL CAUSE THE SIGNAL SSM L TO BE
4828                ;ASSERTED ON A WRITE OR READ TO CONTROL REGISTER 6. ON A WRITE OR READ
4829                ;TO CONTROL REGISTER 6, DATA WILL BE WRITTEN OR READ FROM THE MEMORY
4830                ;SIMULATOR RAM SELECTED. THE SIMULATOR MEMORY IS SELECTED VIA THE
4831                ;MODULE SELECT RAM'S (0 AND 1).
4832
4833 014532 005037 002264   CLR    R2LOAD            ;SETUP TO CLEAR REGISTER 2
4834 014536 004737 005414   JSR    PC,LDRDR2        ;GO LOAD, READ AND CHECK REG 2
4835 014542 001405                BEQ    8$                ;IF LOADED OK THEN CONTINUE
4836 014544                ERRDF  2.,R2EROR        ;REGISTER 2 NOT EQUAL EXPECTED
4837 014544 104455   TRAP    C$ERDF
4838 014546 000002   .WORD  2
4839 014550 000000   .WORD  0
4840 014552 003712   .WORD  R2EROR
4841 014554                CKLOOP
4842 014554 104406   TRAP    C$CLP1
4843
4844                ;LOAD MEMORY SIMULATOR ADDRESS TO BE TESTED INTO CONTROL REGISTER 4
4845
4846 014556 004737 005462   8$:   JSR    PC,LDRDR4        ;GO LOAD,READ AND CHECK REGISTER 4
4847 014562 001405                BEQ    9$                ;IF LOADED OK THEN CONTINUE
4848 014564                ERRDF  3.,R4EROR        ;REGISTER 4 NOT EQUAL EXPECTED
4849 014564 104455   TRAP    C$ERDF
4850 014566 000003   .WORD  3
4851 014570 000000   .WORD  0
4852 014572 004000   .WORD  R4EROR
4853 014574                CKLOOP
4854 014574 104406   TRAP    C$CLP1
4855
4856                ;LOAD DATA PATTERN 125252 INTO MEMORY SIMULATOR RAM. ON A WRITE OR
4857                ;READ TO CONTROL REGISTER 6, THE SIGNAL SSM L WILL BE ASSERTED. THIS
4858                ;SIGNAL WILL ENABLE BOTH MODULE SELECT RAMS AND ENABLE THE MEMORY
4859                ;SIMULATOR RAM TO BE WRITTEN OR READ.
4860
4861 014576 012737 125252 002302 9$:   MOV    #125252,R6LOAD    ;SETUP DATA PATTERN TO BE LOADED
4862 014604 005037 002306   CLR    R6MASK            ;SETUP TO CHECK ALL 16 BITS OF RAM
4863 014610 004737 005506   JSR    PC,LDRDR6        ;GO LOAD, READ AND CHECK RAM LOCATION
4864 014614 001405                BEQ    10$               ;IF DATA OK THEN GO COMPLEMENT IT
4865 014616                ERRDF  4,MSGMSH,ALINFO    ;DATA ERROR IN MEMORY SIMULATOR RAM
4866 014616 104455   TRAP    C$ERDF
4867 014620 000004   .WORD  4
4868 014622 002732   .WORD  MSGMSR
4869 014624 004052   .WORD  ALINFO
4870 014626                CKLOOP
4871 014626 104406   TRAP    C$CLP1
4872
4873                ;LOAD DATA PATTERN 052525 INTO SAME MEMORY SIMULATOR RAM LOCATION

```

```
4874 ;AND CHECK THAT IT WAS LOADED CORRECTLY
4875
4876 014630 012737 052525 002302 10$: MOV #052525,R6LOAD ;SETUP DATA PATTERN TO BE LOADED
4877 014636 004737 005506 JSR PC,LDRDR6 ;GO LOAD, READ AND CHECK RAM LOCATION
4878 014642 001404 BEQ 11$ ;IF DATA OK THEN CONTINUE
4879 014644 ERRDF 4,MSGMSR,ALINFO ;DATA ERROR IN MEMORY SIMULATOR RAM
4880 014644 104455 TRAP C$ERDF
4881 014646 000004 .WORD 4
4882 014650 002732 .WORD MSGMSR
4883 014652 004052 .WORD ALINFO
4884 014654 11$: ENDSEG
4885 014654 10001$:
4886 014654 104405 TRAP C$ESEG
4887
4888 ;UPDATE MEMORY SIMULATOR ADDRESS TO NEXT ADDRESS TO BE LOADED
4889
4890 014656 062737 000002 002276 ADD #2,R4LOAD ;UPDATE ADDRESS BY 2
4891 014664 005302 DEC R2 ;DECREMENT 2K WORD COUNTER
4892 014666 001320 BNE 7$ ;IF NOT DONE DO NEXT ADDRESS
4893
4894 ;UPDATE MODULE SELECT RAM ENABLE BITS TO BE LOADED INTO MODULE
4895 ;SELECT RAM 0 ON NEXT PASS OF THIS TEST
4896
4897 014670 006301 ASL R1 ;MOVE MODULE SELECT RAM 0 DATA PATTERN
4898 ;LEFT TO ENABLE NEXT MEMORY SIMULATOR RAM
4899 014672 032701 000020 BIT #BIT4,R1 ;CHECK IF ALL RAMS HAVE BEEN TESTED
4900 014676 001002 BNE 12$ ;IF YES THEN EXIT THE TEST
4901 014700 000137 014332 JMP 1$ ;OTHERWISE DO NEXT RAM
4902 014704 12$: ENDTST
4903 014704 L10055:
4904 014704 104401 TRAP C$ETST
4905
```

4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961

014706
014706
014706 004737 004550
014712 005037 002276
014716 012701 000001
014722 012702 011111
014726
014726 104404

014730 012737 000004 002264
014736 012737 177740 002270
014744 004737 005414
014750 001405
014752
014752 104455
014754 000002
014756 000000
014760 003712
014762
014762 104406

014764 004737 005462

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

..**
: THIS TEST WILL CHECK THAT EACH MEMORY SIMULATOR RAM IS SELECTED BY WRITING
: A DIFFERENT 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

T23:: BGNTST
JSR PC,INITMS ;SELECT AND INIT MEMORY SIMULATOR
CLR R4LOAD ;SET 1ST ADDRESS TO BE TESTED TO 0
MOV #BIT0,R1 ;SETUP DATA FOR MODULE SELECT RAM 0
MOV #11111,R2 ;SETUP DATA PATTERN FOR MEM SIM RAM 0

1\$: BGNSEG
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.
MOV #MSEL0,R2LOAD ;SETUP BITS TO BE LOADED
MOV #177740,R2MASK ;SETUP REGISTER 2 MASK WORD
JSR PC,LDRDR2 ;GO LOAD,READ AND CHECK REG 2
BEQ 2\$;IF LOADED OK THEN CONT
ERRDF 2,,R2EROR ;REGISTER 2 NOT EQUAL EXPECTED
TRAP C\$ERDF
.WORD 2
.WORD 0
.WORD R2EROR
CKLOOP
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.

2\$: JSR PC,LDRDR4 ;GO LOAD,READ AND CHECK REGISTER 4

```

4962 014770 001405      BEQ      3$                ;IF LOADED OK THEN CONTINUE
4963 014772              ERRDF    3,,R4EROR      ;REGISTER 4 NOT EQUAL EXPECTED
4964 014772 104455      TRAP    C$ERDF
4965 014774 000003      .WORD   3
4966 014776 000000      .WORD   0
4967 015000 004000      .WORD  R4EROR
4968 015002              CKLOOP
4969 015002 104406      TRAP    C$CLP1
4970
4971              ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 0 AND CHECK THAT IT WAS
4972              ;WRITTEN CORRECTLY. THE DATA PATTERNS THAT WILL BE WRITTEN ARE 1, 2, 4,
4973              ;10. THESE PATTERNS WILL ENABLE THE SIGNALS EN0 H, EN1 H, EN2 H, AND
4974              ;EN3 H RESPECTIVELY. THESE SIGNALS ARE USED TO SELECT MEMORY SIMULATOR
4975              ;MEMORY WHEN THE SIGNAL SSM L IS ASSERTED LATER ON IN THE TEST.
4976
4977 015004 010137 002302 3$:  MOV     R1,R6LOAD          ;GET THE PATTERN TO BE LOADED
4978 015010 012737 177760 002306  MOV     #177760,R6MASK     ;SETUP REGISTER 6 MASK WORD
4979 015016 004737 005506      JSR     PC,LDRDR6         ;GO LOAD, READ AND CHECK RAM 0
4980 015022 001405      BEQ     4$                ;IF OK THEN CONTINUE
4981 015024              ERRDF    4,MSGMDO,ALINFO    ;DATA ERROR IN MODULE SELECT RAM 0
4982 015024 104455      TRAP    C$ERDF
4983 015026 000004      .WORD   4
4984 015030 002513      .WORD  MSGMDO
4985 015032 004052      .WORD  ALINFO
4986 015034              CKLOOP
4987 015034 104406      TRAP    C$CLP1
4988
4989              ;SET SIGNALS MSEL0 H AND MSEL1 H TO A ONE AND SIGNALS MSAD17 H AND MSAD16 H
4990              ;TO A ZERO IN CONTROL REGISTER 2. MSEL0 H AND MSEL1 H ON A ONE, WILL
4991              ;CAUSE THE SIGNAL SMDS1 L TO BE ASSERTED ON A WRITE OR READ TO CONTROL
4992              ;REGISTER 6. SMDS1 L WILL SELECT MODULE SELECT RAM 1. MSAD BITS 17 AND 16
4993              ;ON A ZERO WILL SELECT ADDRESS 0 OF MODULE SELECT RAM 1.
4994
4995 015036 012737 000014 002264 4$:  MOV     #MSEL0!MSEL1,R2LOAD ;SETUP BITS TO BE LOADED
4996 015044 004737 005414      JSR     PC,LDRDR2         ;GO LOAD, READ AND CHECK REGISTER 2
4997 015050 001405      BEQ     5$                ;IF LOADED OK THEN CONTINUE
4998 015052              ERRDF    2,,R2EROR      ;REGISTER 2 NOT EQUAL EXPECTED
4999 015052 104455      TRAP    C$ERDF
5000 015054 000002      .WORD   2
5001 015056 000000      .WORD   0
5002 015060 003712      .WORD  R2EROR
5003 015062              CKLOOP
5004 015062 104406      TRAP    C$CLP1
5005
5006              ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 1 AND CHECK THAT THE
5007              ;PATTERN WAS LOADED CORRECTLY. A DATA PATTERN OF ALL ONES WILL BE
5008              ;WRITTEN, HOWEVER, WHEN THE LOCATION IS READ BACK, A DATA PATTERN
5009              ;EQUAL TO ONE WILL BE READ. SETTING THE LEAST SIGNIFICANT BIT IN
5010              ;THE DATA PATTERN WILL FORCE THE MORE SIGNIFICANT BITS TO ZERO.
5011
5012 015064 012737 000017 002302 5$:  MOV     #BIT3!BIT2!BIT1!BIT0,R6LOAD ;GET DATA PATTERN TO BE LOADED
5013 015072 012737 000001 002304  MOV     #BIT0,R6GOOD      ;SETUP EXPECTED DATA PATTERN
5014 015100 004737 005514      JSR     PC,LDRDR6         ;GO LOAD,READ AND CHECK RAM 1
5015 015104 001405      BEQ     6$                ;IF DATA OK THEN CONTINUE
5016 015106              ERRDF    4,MSGMD1,ALINFO    ;DATA ERROR IN MODULE SELECT RAM 1
5017 015106 104455      TRAP    C$ERDF

```

```

5018 015110 000004      .WORD 4
5019 015112 002555      .WORD MSGMD1
5020 015114 004052      .WORD ALINFO
5021 015116             CKLOOP
5022 015116 104406      TRAP C$CLP1
5023
5024             ;CLEAR BITS MSEL0 H, MSEL1 H, MSAD17 H AND MSAD16 H IN CONTROL REGISTER
5025             ;2. MSEL0 H AND MSEL1 H ON A ZERO WILL CAUSE THE SIGNAL SSM L TO BE
5026             ;ASSERTED ON A WRITE OR READ TO CONTROL REGISTER 6. ON A WRITE OR READ
5027             ;TO CONTROL REGISTER 6, DATA WILL BE WRITTEN OR READ FROM THE MEMORY
5028             ;SIMULATOR RAM SELECTED. THE SIMULATOR MEMORY IS SELECTED VIA THE
5029             ;MODULE SELECT RAM'S (0 AND 1).
5030
5031 015120 005037 002264      6$: CLR R2LOAD           ;SETUP TO CLEAR REGISTER 2
5032 015124 004737 005414     JSR PC,LDRDR2       ;GO LOAD, READ AND CHECK REG 2
5033 015130 001405             BEQ 7$              ;IF LOADED OK THEN CONTINUE
5034 015132             ERRDF 2,,R2EROR          ;REGISTER 2 NOT EQUAL EXPECTED
5035 015132 104455      TRAP C$ERDF
5036 015134 000002      .WORD 2
5037 015136 000000      .WORD 0
5038 015140 003712      .WORD R2EROR
5039 015142             CKLOOP
5040 015142 104406      TRAP C$CLP1
5041
5042             ;LOAD DATA PATTERN INTO MEMORY SIMULATOR RAM AND CHECK THAT THE CORRECT
5043             ;PATTERN WAS LOADED. THE SIGNAL SSM L WILL BE ASSERTED. THIS SIGNAL
5044             ;WILL ENABLE BOTH MODULE SELECT RAMS AND ENABLE THE MEMORY SIMULATOR RAM
5045             ;TO BE WRITTEN OR READ.
5046
5047 015144 010237 002302      7$: MOV R2,R6LOAD       ;SETUP DATA PATTERN TO BE LOADED
5048 015150 005037 002306     CLR R6MASK          ;SET MASK WORD TO 0
5049 015154 004737 005506     JSR PC,LDRDR6       ;GO LOAD, READ AND CHECK RAM LOCATION
5050 015160 001404             BEQ 8$              ;IF DATA OK THEN CONTINUE
5051 015162             ERRDF 4,MSGMSR,ALINFO      ;DATA ERROR IN MEMORY SIMULATOR RAM
5052 015162 104455      TRAP C$ERDF
5053 015164 000004      .WORD 4
5054 015166 002732      .WORD MSGMSR
5055 015170 004052      .WORD ALINFO
5056 015172             8$: ENDSEG
5057 015172             10000$:
5058 015172 104405      TRAP C$ESEG
5059
5060 015174 006301             ASL R1              ;UPDATE MODULE SELECT RAM 0'S DATA
5061 015176 062737 020000 002276 ADD #MSAD13,R4LOAD   ;UPDATE ADDRESS TO ADDRESS 0 OF NEXT RAM
5062 015204 062702 011111     ADD #11111,R2       ;UPDATE DATA PATTERN FOR NEXT RAM
5063 015210 032701 000020     BIT #BIT4,R1        ;CHECK IF ALL 4 RAM'S WRITTEN
5064 015214 001002             BNE 9$              ;IF YES THEN GO READ THEM AGAIN
5065 015216 000137 014726     JMP 1$              ;GO WRITE ADDRESS 0 OF NEXT RAM
5066
5067 015222 005037 002276      9$: CLR R4LOAD          ;RESET ADDRESS TO ADDRESS 0
5068 015226 012702 011111     MOV #11111,R2       ;RESET DATA PATTERN
5069
5070             ;THE FOLLOWING SECTION OF CODE WILL READ ADDRESS 0 OF EACH RAM AND
5071             ;CHECK THAT THE DATA PATTERNS WRITTEN PREVIOUSLY REMAIN UNCHANGED.
5072
5073 015232             10$: BGNSEG

```



```

5074 015232 104404 TRAP C$BSEG
5075
5076 :CLEAR BITS MSEL0 H, MSEL1 H, MSAD17 H AND MSAD 16 H IN CONTROL REGISTER
5077 :2. MSEL0 H AND MSEL1 H ON A ZERO WILL CAUSE THE SIGNAL SSM L TO BE
5078 :ASSERTED ON A READ COMMAND TO CONTROL REGISTER 6. THE SIGNAL SSM L
5079 :WILL ENABLE THE SIMULATOR RAM MEMORY'S TO BE READ.
5080
5081 015234 005037 002264 CLR R2LOAD :CLEAR ALL BITS IN CONTROL REGISTER 2
5082 015240 004737 005414 JSR PC,LDRDR2 :GO LOAD, READ AND CHECK REGISTER 2
5083 015244 001405 BEQ 11$ :IF LOADED OK THEN CONTINUE
5084 015246 ERRDF 2,R2EROR :REGISTER 2 NOT EQUAL EXPECTED
5085 015246 104455 TRAP C$ERDF
5086 015250 000002 .WORD 2
5087 015252 000000 .WORD 0
5088 015254 003712 .WORD R2EROR
5089 015256 CKLOOP
5090 015256 104406 TRAP C$CLP1
5091
5092 :LOAD ADDRESS 0 OF THE RAM TO BE TESTED INTO CONTROL REGISTER 4.
5093
5094 015260 004737 005462 11$: JSR PC,LDRDR4 :GO LOAD, READ AND CHECK REGISTER 4
5095 015264 001405 BEQ 12$ :IF LOADED OK THEN CONTINUE
5096 015266 ERRDF 3,R4EROR :REGISTER 4 NOT EQUAL EXPECTED
5097 015266 104455 TRAP C$ERDF
5098 015270 000003 .WORD 3
5099 015272 000000 .WORD 0
5100 015274 004000 .WORD R4EROR
5101 015276 CKLOOP
5102 015276 104406 TRAP C$CLP1
5103
5104 :READ MEMORY SIMULATOR RAM AND CHECK THAT THE DATA PREVIOUSLY WRITTEN
5105 :DID NOT CHANGE.
5106
5107 015300 010237 002302 12$: MOV R2,R6LOAD :SETUP DATA PREVIOUSLY WRITTEN
5108 015304 013737 002302 002304 MOV R6LOAD,R6GOOD :SETUP EXPECTED DATA
5109 015312 005037 002306 CLR R6MASK :CLEAR THE MASK WORD
5110 015316 004737 005522 JSR PC,READR6 :READ AND CHECK RAM LOCATION
5111 015322 001404 BEQ 13$ :IF DATA OK THEN CONTINUE
5112 015324 ERRDF 4,MSGMSC,ALINFO :CHIP ENABLE ERROR - MEM SIM RAM
5113 015324 104455 TRAP C$ERDF
5114 015326 000004 .WORD 4
5115 015330 002775 .WORD MSGMSC
5116 015332 004052 .WORD ALINFO
5117 015334 13$: ENDSEG
5118 015334 10001$:
5119 015334 104405 TRAP C$ESEG
5120
5121 015336 062702 011111 ADD #11111,R2 :UPDATE DATA PATTERN
5122 015342 062737 020000 002276 ADD #MSAD13,R4LOAD :UPDATE ADDRESS TO 0 OF NEXT RAM
5123 015350 100330 BPL 10$ :GO CHECK NEXT ADDRESS IF NOT DONE
5124 015352 ENDTST
5125 015352 L10056:
5126 015352 104401 TRAP C$ETST
5127

```

5128
5129
5130
5131
5132
5133
5134
5135
5136
5137
5138
5139
5140
5141
5142
5143
5144
5145
5146
5147
5148
5149
5150
5151
5152
5153
5154
5155
5156
5157
5158
5159
5160
5161
5162
5163
5164
5165
5166
5167
5168
5169
5170
5171
5172
5173
5174
5175
5176
5177
5178
5179
5180
5181
5182
5183

015354
015354
015354 004737 004550
015360 005037 002276
015364 012701 000001

015370
015370 104404

015372 012737 000004 002264
015400 012737 177740 002270
015406 004737 005414
015412 001405
015414
015414 104455
015416 000002
015420 000000
015422 003712
015424

```
.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 BE'NG 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

:--

T24:: BGNTST
      JSR    PC,INITMS      ;SELECT AND INIT MEMORY SIMULATOR
      CLR    R4LOAD        ;SET 1ST ADDRESS TO BE TESTED TO 0
      MOV    #BIT0,R1      ;SETUP DATA FOR MODULE SELECT RAM 0

1$:   BGNSEG
      TRAP  C$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 SMD50 L TO BE ASSERTED ON A WRITE OR
      ;READ COMMAND TO CONTROL REGISTER 6. SMD50 L WILL SELECT MODULE SELECT
      ;RAM 0.

      MOV    #MSEL0,R2LOAD  ;SETUP BITS TO BE LOADED
      MOV    #177740,R2MASK ;SETUP REGISTER 2 MASK WORD
      JSR    PC,LDRDR?     ;GO LOAD,READ AND CHECK REG 2
      BEQ    Z$            ;IF LOADED OK THEN CONT
      ERDF  2,,R2EROR     ;REGISTER 2 NOT EQUAL EXPECTED
      TRAP  C$ERDF
      .WORD 2
      .WORD 0
      .WORD R2EROR
      CKLOOP
```

```

5184 015424 104406 TRAP C$CLP1
5185
5186 :LOAD THE ADDRESS TO BE TESTED INTO CONTROL REGISTER 4. THE ADDRESS
5187 :WILL APPEAR ON MSAD BITS 15-0 WHICH IS THE OUTPUT OF CONTROL REGISTER 4
5188
5189 015426 004737 005462 2$: JSR PC,LDRDR4 ;GO LOAD,READ AND CHECK REGISTER 4
5190 015432 001405 BEQ 3$ ;IF LOADED OK THEN CONTINUE
5191 015434 ERRDF 3,,R4EROR ;REGISTER 4 NOT EQUAL EXPECTED
5192 015434 104455 TRAP C$ERDF
5193 015436 000003 .WORD 3
5194 015440 000000 .WORD 0
5195 015442 004000 .WORD R4EROR
5196 015444 CKLOOP
5197 015444 104406 TRAP C$CLP1
5198
5199 :WRITE THE DATA PATTERN INTO MODULE SELECT RAM 0 AND CHECK THAT IS WAS
5200 :WRITTEN CORRECTLY. THE DATA PATTERNS THAT WILL BE WRITTEN ARE 1, 2, 4,
5201 :10. THESE PATTERNS WILL ENABLE THE SIGNALS EN0 H, EN1 H, EN2 H, AND
5202 :EN3 H RESPECTIVELY. THESE SIGNALS ARE USED TO SELECT MEMORY SIMULATOR
5203 :MEMORY WHEN THE SIGNAL SSM L IS ASSERTED LATER ON IN THE TEST.
5204
5205 015446 010137 002302 3$: MOV R1,R6LOAD ;GET THE PATTERN TO BE LOADED
5206 015452 012737 177760 002306 MOV #177760,R6MASK ;SETUP REGISTER 6 MASK WORD
5207 015460 004737 005506 JSR PC,LDRDR6 ;GO LOAD, READ AND CHECK RAM 0
5208 015464 001405 BEQ 4$ ;IF OK THEN CONTINUE
5209 015466 ERRDF 4,MSGMDO,ALINFO ;DATA ERROR IN MODULE SELECT RAM 0
5210 015466 104455 TRAP C$ERDF
5211 015470 000004 .WORD 4
5212 015472 002513 .WORD MSGMDO
5213 015474 004052 .WORD ALINFO
5214 015476 CKLOOP
5215 015476 104406 TRAP C$CLP1
5216
5217 :SET SIGNALS MSEL0 H AND MSEL1 H TO A ONE AND SIGNALS MSAD17 H AND MSAD16 H
5218 :TO A ZERO IN CONTROL REGISTER 2. MSEL0 H AND MSEL1 H ON A ONE, WILL
5219 :CAUSE THE SIGNAL SMDS1 L TO BE ASSERTED ON A WRITE OR READ TO CONTROL
5220 :REGISTER 6. SMDS1 L WILL SELECT MODULE SELECT RAM 1. MSAD BITS 17 AND 16
5221 :ON A ZERO WILL SELECT ADDRESS 0 OF MODULE SELECT RAM 1.
5222
5223 015500 012737 000014 002264 4$: MOV #MSEL0!MSEL1,R2LOAD ;SETUP BITS TO BE LOADED
5224 015506 004737 005414 JSR PC,LDRDR2 ;GO LOAD, READ AND CHECK REGISTER 2
5225 015512 001405 BEQ 5$ ;IF LOADED OK THEN CONTINUE
5226 015514 ERRDF 2,,R2EROR ;REGISTER 2 NOT EQUAL EXPECTED
5227 015514 104455 TRAP C$ERDF
5228 015516 000002 .WORD 2
5229 015520 000000 .WORD 0
5230 015522 003712 .WORD R2EROR
5231 015524 CKLOOP
5232 015524 104406 TRAP C$CLP1
5233
5234 :WRITE THE DATA PATTERN INTO MODULE SELECT RAM 1 AND CHECK THAT THE
5235 :PATTERN WAS LOADED CORRECTLY. A DATA PATTERN OF ALL ONES WILL BE
5236 :WRITTEN, HOWEVER, WHEN THE LOCATION IS READ BACK, A DATA PATTERN
5237 :EQUAL TO ONE WILL BE READ. SETTING THE LEAST SIGNIFICANT BIT IN
5238 :THE DATA PATTERN WILL FORCE THE MORE SIGNIFICANT BITS TO ZERO.
523

```

```

5240 015526 012737 000017 002302 5$: M. #BIT3!BIT2!BIT1!BIT0,R6LOAD ;GET DATA PATTERN TO BE LOADED
5241 015534 012737 000001 002304 MOV #BIT0,R6GOOD ;SETUP EXPECTED DATA PATTERN
5242 015542 004737 005514 JSR PC,LDRD6R ;GO LOAD,READ AND CHECK RAM 1
5243 015546 001404 BEQ 6$ ;IF DATA OK THEN CONTINUE
5244 015550 ERRDF 4,MSGMD1,ALINFO ;DATA ERROR IN MUDULE SELECT RAM 1
5245 015550 104455 TRAP C$ERDF
5246 015552 000004 .WORD 4
5247 015554 002555 .WORD MSGMD1
5248 015556 004052 .WORD ALINFO
5249 015560 6$: ENDSEG
5250 015560 10000$:
5251 015560 104405 TRAP C$ESEG
5252
5253 ^15562 012702 010000 MOV #4096.,R2 ;SETUP 4K WORD COUNTER TO DO 1 RAM
5254
5255 015566 7$: BGNSEG
5256 015566 104404 TRAP C$BSEG
5257
5258 ;CLEAR BITS MSEL0 H, MSEL1 H, MSAD17 H AND MSAD16 H IN CONTROL REGISTER
5259 ;2. MSEL0 H AND MSEL1 H ON A ZERO WILL CAUSE THE SIGNAL SSM L TO BE
5260 ;ASSERTED ON A WRITE OR READ TO CONTROL REGISTER 6. ON A WRITE OR READ
5261 ;TO CONTROL REGISTER 6, DATA WILL BE WRITTEN OR READ FROM THE MEMORY
5262 ;SIMULATOR RAM SELECTED. THE SIMULATOR MEMORY IS SELECTED VIA THE
5263 ;MODULE SELECT RAM'S (0 AND 1).
5264
5265 015570 005037 002264 CLR R2LOAD ;SETUP TO CLEAR REGISTER 2
5266 015574 004737 005414 JSR PC,LDRDR2 ;GO LOAD, READ AND CHECK REG 2
5267 015600 001405 BEQ 8$ ;IF LOADED OK THEN CONTINUE
5268 015602 ERRDF 2,R2EROR ;REGISTER 2 NOT EQUAL EXPECTED
5269 015602 104455 TRAP C$ERDF
5270 015604 000002 .WORD 2
5271 015606 000000 .WORD 0
5272 015610 003712 .WORD R2EROR
5273 015612 CKLOOP
5274 015612 104406 TRAP C$CLP1
5275
5276 ;LOAD MEMORY SIMULATOR ADDRESS TO BE TESTED INTO CONTROL REGISTER 4
5277
5278 015614 004737 005462 8$: JSR PC,LDRDR4 ;GO LOAD,READ AND CHECK REGISTER 4
5279 015620 001405 BEQ 9$ ;IF LOADED OK THEN CONTINUE
5280 015622 ERRDF 3,R4EROR ;REGISTER 4 NOT EQUAL EXPECTED
5281 015622 104455 TRAP C$ERDF
5282 015624 000003 .WORD 3
5283 015626 000000 .WORD 0
5284 015630 004000 .WORD R4EROR
5285 015632 CKLOOP
5286 015632 104406 TRAP C$CLP1
5287
5288 ;LOAD DATA PATTERN EQUAL TO ADDRESS INTO MEMORY SIMULATOR RAM. ON A
5289 ;WRITE OR READ TO CONTROL REGISTER 6, THE SIGNAL SSM L WILL BE ASSERTED.
5290 ;THIS SIGNAL WILL ENABLE BOTH MODULE SELECT RAMS AND ENABLE THE MEMORY
5291 ;SIMULATOR RAM TO BE WRITTEN OR READ.
5292
5293 015634 013737 002276 002302 9$: MOV R4LOAD,R6LOAD ;SETUP DATA PATTERN TO BE LOADED
5294 015642 005037 002306 CLR R6MASK ;SETUP TO CHECK ALL 16 BITS OF RAM
5295 015646 004737 005506 JSR PC,LDRDR6 ;GO LOAD, READ AND CHECK RAM LOCATION
    
```

```

5296 015652 001404          BEQ      10$          ;IF DATA OK THEN GO COMPLEMENT IT
5297 015654                ERRDF   4,MSGMSR,ALINFO ;DATA ERROR IN MEMORY SIMULATOR RAM
5298 015654 104455          TRAP   C$ERDF
5299 015656 000004          .WORD  4
5300 015660 002732          .WORD  MSGMSR
5301 015662 004052          .WORD  ALINFO
5302 015664                10$: ENDSEG
5303 015664                10001$:
5304 015664 104405          TRAP   C$ESEG
5305
5306                ;UPDATE MEMORY SIMULATOR ADDRESS TO NEXT ADDRESS TO BE LOADED
5307
5308 015666 062737 000002 002276 ADD     #2,R4LOAD      ;UPDATE ADDRESS BY 2
5309 015674 005302                DEC     R2             ;DECREMENT 2K WORD COUNTER
5310 015676 001333                BNE    7$             ;IF NOT DONE DO NEXT ADDRESS
5311
5312                ;UPDATE MODULE SELECT RAM ENABLE BITS TO BE LOADED INTO MODULE
5313                ;SELECT RAM 0 ON NEXT PASS OF THIS TEST
5314
5315 015700 006301          ASL    R1             ;MOVE MODULE SELECT RAM 0 DATA PATTERN
5316                ;LEFT TO ENABLE NEXT MEMORY SIMULATOR RAM
5317 015702 032701 000020          BIT    #BIT4,R1      ;CHECK IF ALL RAMS HAVE BEEN TESTED
5318 015706 001002                BNE    11$           ;IF YES THEN EXIT THE TEST
5319 015710 000137 015370          JMP    1$             ;OTHERWISE DO NEXT RAM
5320
5321                ;THE FOLLOWING SECTION OF CODE WILL READ THE MEMORY SIMULATOR RAMS
5322                ;AND CHECK THAT EACH ADDRESS OF THE RAM CONTAINS AS DATA ITS ADDRESS.
5323
5324 015714 005037 002276          11$: CLR    R4LOAD        ;RESET THE ADDRESS TO 0
5325
5326 015720                12$: BGNSEG
5327 015720 104404          TRAP   C$BSEG
5328
5329                ;LOAD THE ADDRESS TO BE CHECKED INTO CONTROL REGISTER 4. THE MODULE
5330                ;SELECT RAMS HAVE BEEN INITIALIZED PREVIOUSLY.
5331
5332 015722 004737 005462          JSR    PC,LDRDR4     ;GO LOAD, READ AND CHECK REGISTER 4
5333 015726 001405                BEQ    13$           ;IF LOADED OK THEN CONTINUE
5334 015730                ERRDF   3,R4EROR   ;REGISTER 4 NOT EQUAL EXPECTED.
5335 015730 104455          TRAP   C$ERDF
5336 015732 000003          .WORD  3
5337 015734 000000          .WORD  0
5338 015736 004000          .WORD  R4EROR
5339 015740                CKLOOP
5340 015740 104406          TRAP   C$CLP1
5341
5342                ;READ THE MEMORY SIMULATOR RAM AND CHECK THAT THE ADDRESS CONTAINS ITS
5343                ;ADDRESS AS DATA.
5344
5345 015742 013737 002276 002302 13$: MOV    R4LOAD,R6LOAD ;SETUP DATA THAT WAS PREVIOUSLY WRITTEN
5346 015750 013737 002302 002304 MOV    R6LOAD,R6GOOD ;SETUP EXPECTED DATA
5347 015756 005037 002306          CLR    R6MASK        ;SET REGISTER 6 MASK TO ZERO
5348 015762 004737 005522          JSR    PC,READR6     ;GO READ AND CHECK THE RAM LOCATION
5349 015766 001404                BEQ    14$           ;IF DATA EQUALS ADDRESS THEN CONTINUE
5350 015770                ERRDF   4,MSGMSR,ALINFO ;DATA ERROR IN MEMORY SIMULATOR RAM
5351 015770 104455          TRAP   C$ERDF
    
```

```

5352 015772 000004      .WORD 4
5353 015774 002732      .WORD MSGMSR
5354 015776 004052      .WORD ALINFO
5355 016000      14$: ENDSEG
5356 016000      10002$:
5357 016000 104405      TRAP C$ESEG
5358
5359 016002 062737 000002 002276      ADD #2,R4LOAD      ;UPDATE THE ADDRESS TO THE NEXT LOCATION
5360 016010 100343      BPL 12$            ;IF NOT DONE CHECK NEXT ADDRESS
5361
5362      ;THE FOLLOWING SECTION OF CODE WILL READ THE CONTENTS OF A LOCATION
5363      ;CHECKING THAT THE CONTENTS EQUALS THE ADDRESS. THE TEST WILL THEN
5364      ;WRITE THE ONES COMPLEMENT OF THE ADDRESS INTO THE LOCATION AND CHECK
5365      ;THAT THE LOCATION WAS WRITTEN CORRECTLY.
5366
5367 016012 005037 002276      CLR R4LOAD        ;RESET THE ADDRESS TO ADDRESS 0
5368
5369 016016      15$: BGNSEG
5370 016016 104404      TRAP C$BSEG
5371
5372      ;LOAD THE ADDRESS TO BE TESTED INTO CONTROL REGISTER 4
5373
5374 016020 004737 005462      JSR PC,LDRDR4     ;GO LOAD, READ, AND CHECK REGISTER 4
5375 016024 001405      BEQ 16$          ;IF LOADED OK THEN CONTINUE
5376 016026      ERRDF 3,R4EROR ;REGISTER 4 NOT EQUAL EXPECTED
5377 016026 104455      TRAP C$ERDF
5378 016030 000003      .WORD 3
5379 016032 000000      .WORD 0
5380 016034 004000      .WORD R4EROR
5381 016036      CKLOOP
5382 016036 104406      TRAP C$CLP1
5383
5384      ;READ THE LOCATION CHECKING THAT THE DATA STORED IN LOCATION EQUALS
5385      ;THE ADDRESS OF THE LOCATION
5386
5387 016040 013737 002276 002302 16$: MOV R4LOAD,R6LOAD ;GET THE ADDRESS
5388 016046 013737 002302 002304      MOV R6LOAD,R6GOOD ;SETUP FOR EXPECTED DATA
5389 016054 004737 005522      JSR PC,READR6    ;GO READ AND CHECK THE DATA
5390 016060 001405      BEQ 17$          ;IF ADDRESS EQUALS DATA THEN CONT
5391 016062      ERRDF 4,MSGMSR,ALINFO ;ADDRESS SHORT OR DATA NEQ EXPECTED
5392 016062 104455      TRAP C$ERDF
5393 016064 000004      .WORD 4
5394 016066 002732      .WORD MSGMSR
5395 016070 004052      .WORD ALINFO
5396 016072      CKLOOP
5397 016072 104406      TRAP C$CLP1
5398
5399      ;WRITE THE COMPLEMENT OF THE ADDRESS IN THE LOCATION ADDRESSED.
5400
5401 016074 013737 002276 002302 17$: MOV R4LOAD,R6LOAD ;GET THE ADDRESS
5402 016102 005137 002302      COM R6LOAD        ;COMPLEMENT IT
5403 016106 004737 005506      JSR PC,LDRDR6    ;GO LOAD, READ AND CHECK RAM
5404 016112 001404      BEQ 18$          ;IF DATA OK THEN CONTINUE
5405 016114      ERRDF 4,MSGMSR,ALINFO ;DATA ERROR IN MEMORY SIMULATOR RAM
5406 016114 104455      TRAP C$ERDF
5407 016116 000004      .WORD 4
  
```

```

5408 016120 002732      .WORD  MSGMSR
5409 016122 004052      .WORD  ALINFO
5410 016124             18$:  ENDSEG
5411 016124             10003$:
5412 016124 104405      TRAP   C$ESEG
5413
5414 016126 062737 000002 002276  ADD   #2,R4LOAD      ;UPDATE THE TEST ADDRESS BY 2
5415 016134 100330      BPL    15$           ;!F NOT DONE GO DO NEXT ADDRESS
5416
5417                       ;THE FOLLOWING SECTION OF CODE WILL READ THE MEMORY SIMULATOR RAM'S
5418                       ;CHECKING THAT EACH ADDRESS CONTAINS ITS ONES COMPLEMENT AS DATA
5419
5420 016136 005037 002276  CLR    R4LOAD        ;RESET THE ADDRESS POINTER TO 0
5421
5422 016142             19$:  BGNSEG
5423 016142 104404      TRAP   C$BSEG
5424
5425                       ;LOAD ADDRESS TO BE TESTED INTO CONTROL REGISTER 4
5426
5427 016144 004737 005462  JSR    PC,LDRDR4     ;GO LOAD, READ AND CHECK REGISTER 4
5428 016150 001405      BEQ    20$           ;IF LOADED OK THEN CONTINUE
5429 016152             ERRDF 3,R4EROR      ;REGISTER 4 NOT EQUAL EXPECTED
5430 016152 104455      TRAP   C$ERDF
5431 016154 000003      .WORD  3
5432 016156 000000      .WORD  0
5433 016160 004000      .WORD  R4EROR
5434 016162             CKLOOP
5435 016162 104406      TRAP   C$CLP1
5436
5437                       ;READ THE MEMORY SIMULATOR RAM AND CHECK THAT THE ADDRESS CONTAINS
5438                       ;ITS COMPLEMENT AS DATA
5439
5440 016164 013737 002276 002302 20$:  MOV   R4LOAD,R6LOAD  ;GET THE ADDRESS
5441 016172 005137 002302      COM   R6LOAD        ;SET IT TO ITS ONES COMPLEMENT
5442 016176 013737 002302 002304  MOV   R6LOAD,R6GOOD  ;SETUP EXPECTED DATA
5443 016204 004737 005522      JSR   PC,READR6     ;GO READ AND CHEK THE RAM LOCATION
5444 016210 001404      BEQ   21$           ;IF DATA OK THEN CONTINUE
5445 016212             ERRDF 4,MSGMSR,ALINFO ;LOCATION NOT EQUAL TO ITS ONES COMP
5446 016212 104455      TRAP   C$ERDF
5447 016214 000004      .WORD  4
5448 016216 002732      .WORD  MSGMSR
5449 016220 004052      .WORD  ALINFO
5450
5451 016222             21$:  ENDSEG      ;OR AN ADDRESS SHORT IN RAM
5452 016222             10004$:
5453 016222 104405      TRAP   C$ESEG
5454
5455 016224 062737 000002 002276  ADD   #2,R4LOAD      ;UPDATE THE ADDRESS TO BE TESTED
5456 016232 100343      BPL    19$           ;GO DO NEXT ADDRESS
5457 016234             ENDTST
5458 016234             L10057:
5459 016234 104401      TRAP   C$ETST
5460

```

5461
5462
5463
5464
5465
5466
5467
5468
5469
5470
5471
5472
5473
5474
5475
5476
5477
5478
5479
5480
5481
5482
5483
5484
5485
5486
5487
5488
5489
5490
5491
5492
5493
5494
5495
5496
5497
5498
5499
5500
5501
5502
5503
5504
5505
5506
5507
5508
5509
5510
5511
5512
5513
5514
5515
5516

016236
016236
016236 004737 004550
016242 012737 000001 002276
016250 012701 000001
016254 012703 125252
016260
016260 104404
016262 012737 000004 002264
016270 012737 177740 002270
016276 004737 005414
016302 001405
016304
016304 104455
016306 000002
016310 000000
016312 003712
016314
016314 104406

.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

T25:: BGNTST
 JSR PC,INITMS ;SELECT AND INIT MEMORY SIMULATOR
 MOV #BIT0,R4LOAD ;SET 1ST ADDRESS TO BE TESTED TO 1
 MOV #BIT0,R1 ;SETUP DATA FOR MODULE SELECT RAM 0
 MOV #125252,R3 ;INIT DATA PATTERN TO 125252
 1\$: BGNSEG
 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.
 MOV #MSEL0,R2LOAD ;SETUP BITS TO BE LOADED
 MOV #177740,R2MASK ;SETUP REGISTER 2 MASK WORD
 JSR PC,LDRDR2 ;GO LOAD,READ AND CHECK REG 2
 BEQ 2\$;IF LOADED OK THEN CONT
 ERDF 2,R2EROR ;REGISTER 2 NOT EQUAL EXPECTED
 TRAP C\$ERDF
 .WORD 2
 .WORD 0
 .WORD R2EROR
 CKLOOP
 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


```

5517 016316 004737 005462 2$: JSR PC,LDRDR4 ;GO LOAD,READ AND CHECK REGISTER 4
5518 016322 001405 BEQ 3$ ;IF LOADED OK THEN CONTINUE
5519 016324 ERRDF 3,,R4EROR ;REGISTER 4 NOT EQUAL EXPECTED
5520 016324 104455 TRAP C$ERRDF
5521 016326 000003 .WORD 3
5522 016330 000000 .WORD 0
5523 016332 004000 .WORD R4EROR
5524 016334 CKLOOP
5525 016334 104406 TRAP C$CLP1
5526
5527 ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 0 AND CHECK THAT IS WAS
5528 ;WRITTEN CORRECTLY. THE DATA PATTERNS THAT WILL BE WRITTEN ARE 1, 2, 4,
5529 ;10. THESE PATTERNS WILL ENABLE THE SIGNALS EN0 H, EN1 H, EN2 H, AND
5530 ;EN3 H RESPECTIVELY. THESE SIGNALS ARE USED TO SELECT MEMORY SIMULATOR
5531 ;MEMORY WHEN THE SIGNAL SSM L IS ASSERTED LATER ON IN THE TEST.
5532
5533 016336 010137 002302 3$: MOV R1,R6LOAD ;GET THE PATTERN TO BE LOADED
5534 016342 012737 177760 002306 MOV #177760,R6MASK ;SETUP REGISTER 6 MASK WORD
5535 016350 004737 005506 JSR PC,LDRDR6 ;GO LOAD, READ AND CHECK RAM 0
5536 016354 001405 BEQ 4$ ;IF OK THEN CONTINUE
5537 016356 ERRDF 4,MSGMDO,ALINFO ;DATA ERROR IN MODULE SELECT RAM 0
5538 016356 104455 TRAP C$ERRDF
5539 016360 000004 .WORD 4
5540 016362 002513 .WORD MSGMDO
5541 016364 004052 .WORD ALINFO
5542 016366 CKLOOP
5543 016366 104406 TRAP C$CLP1
5544
5545 ;SET SIGNALS MSEL0 H AND MSEL1 H TO A ONE AND SIGNALS MSAD17 H AND MSAD16 H
5546 ;TO A ZERO IN CONTROL REGISTER 2. MSEL0 H AND MSEL1 H ON A ONE, WILL
5547 ;CAUSE THE SIGNAL SMDS1 L TO BE ASSERTED ON A WRITE OR READ TO CONTROL
5548 ;REGISTER 6. SMDS1 L WILL SELECT MODULE SELECT RAM 1. MSAD BITS 17 AND 16
5549 ;ON A ZERO WILL SELECT ADDRESS 0 OF MODULE SELECT RAM 1.
5550
5551 016370 012737 000014 002264 4$: MOV #MSEL0!MSEL1,R2LOAD ;SETUP BITS TO BE LOADED
5552 016376 004737 005414 JSR PC,LDRDR2 ;GO LOAD, READ AND CHECK REGISTER 2
5553 016402 001405 BEQ 5$ ;IF LOADED OK THEN CONTINUE
5554 016404 ERRDF 2,,R2EROR ;REGISTER 2 NOT EQUAL EXPECTED
5555 016404 104455 TRAP C$ERRDF
5556 016406 000002 .WORD 2
5557 016410 000000 .WORD 0
5558 016412 003712 .WORD R2EROR
5559 016414 CKLOOP
5560 016414 104406 TRAP C$CLP1
5561
5562 ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 1 AND CHECK THAT THE
5563 ;PATTERN WAS LOADED CORRECTLY. A DATA PATTERN OF ALL ONES WILL BE
5564 ;WRITTEN, HOWEVER, WHEN THE LOCATION IS READ BACK, A DATA PATTERN
5565 ;EQUAL TO ONE WILL BE READ. SETTING THE LEAST SIGNIFICANT BIT IN
5566 ;THE DATA PATTERN WILL FORCE THE MORE SIGNIFICANT BITS TO ZERO.
5567
5568 016416 012737 000017 002302 5$: MOV #BIT3!BIT2!BIT1!BIT0,R6LOAD ;GET DATA PATTERN TO BE LOADED
5569 016424 012737 000001 002304 MOV #BIT0,R6GOOD ;SETUP EXPECTED DATA PATTERN
5570 016432 004737 005514 JSR PC,LDRDR6 ;GO LOAD,READ AND CHECK RAM 1
5571 016436 001404 BEQ 6$ ;IF DATA OK THEN CONTINUE
5572 016440 ERRDF 4,MSGMD1,ALINFO ;DATA ERROR IN MODULE SELECT RAM 1

```

```

5573 016440 104455          TRAP    C$ERDF
5574 016442 000004          .WORD  4
5575 016444 002555          .WORD  MSGMD1
5576 016446 004052          .WORD  ALINFO
5577 016450                6$:    ENDSEG
5578 016450                10000$:
5579 016450 104405          TRAP    C$ESEG
5580
5581 016452 012702 010000      MOV     #4096.,R2          ;SETUP 4K WORD COUNTER TO DO 1 RAM
5582
5583 016456                7$:    BGNSEG
5584 016456 104404          TRAP    C$BSEG
5585
5586                ;CLEAR BITS MSEL0 H, MSEL1 H, MSAD17 H AND MSAD16 H IN CONTROL REGISTER
5587                ;2. MSEL0 H AND MSEL1 H ON A ZERO WILL CAUSE THE SIGNAL SSM L TO BE
5588                ;ASSERTED ON A WRITE OR READ TO CONTROL REGISTER 6. ON A WRITE OR READ
5589                ;TO CONTROL REGISTER 6, DATA WILL BE WRITTEN OR READ FROM THE MEMORY
5590                ;SIMULATOR RAM SELECTED. THE SIMULATOR MEMORY IS SELECTED VIA THE
5591                ;MODULE SELECT RAM'S (0 AND 1).
5592
5593 016460 005037 002264      CLR     R2LOAD            ;SETUP TO CLEAR REGISTER 2
5594 016464 004737 005414      JSR    PC,LDRDR2         ;GO LOAD, READ AND CHECK REG 2
5595 016470 001405                BEQ    8$                ;IF LOADED OK THEN CONTINUE
5596 016472                ERRDF  2.,R2EROR      ;REGISTER 2 NOT EQUAL EXPECTED
5597 016472 104455          TRAP    C$ERDF
5598 016474 000002          .WORD  2
5599 016476 000000          .WORD  0
5600 016500 0C3712          .WORD  R2EROR
5601 016502                CKLOOP
5602 016502 104406          TRAP    C$CLP1
5603
5604                ;LOAD MEMORY SIMULATOR ADDRESS TO BE TESTED INTO CONTROL REGISTER 4
5605
5606 016504 004737 005462      8$:    JSR    PC,LDRDR4         ;GO LOAD, READ AND CHECK REGISTER 4
5607 016510 001405                BEQ    9$                ;IF LOADED OK THEN CONTINUE
5608 016512                ERRDF  3.,R4EROR      ;REGISTER 4 NOT EQUAL EXPECTED
5609 016512 104455          TRAP    C$ERDF
5610 016514 000003          .WORD  3
5611 016516 000000          .WORD  0
5612 016520 004000          .WORD  R4EROR
5613 016522                CKLOOP
5614 016522 104406          TRAP    C$CLP1
5615
5616                ;LOAD DATA PATTERN 125252 OR 052525 INTO MEMORY SIMULATOR RAM. ON A
5617                ;WRITE OR READ TO CONTROL REGISTER 6, THE SIGNAL SSM L WILL BE ASSERTED.
5618                ;THIS SIGNAL WILL ENABLE BOTH MODULE SELECT RAMS AND ENABLE THE MEMORY
5619                ;SIMULATOR RAM TO BE WRITTEN OR READ.
5620
5621 016524 010337 002302      9$:    MOV     R3,R6LOAD      ;SETUP DATA PATTERN TO BE LOADED
5622 016530 005037 002306      CLR     R6MASK           ;SETUP TO CHECK ALL 16 BITS OF RAM
5623 016534 004737 005506      JSR    PC,LDRDR6         ;GO LOAD, READ AND CHECK RAM LOCATION
5624 016540 001404                BEQ    10$               ;IF DATA OK THEN GO COMPLEMENT IT
5625 016542                ERRDF  4,MSGMSR,ALINFO ;DATA ERROR IN MEMORY SIMULATOR RAM
5626 016542 104455          TRAP    C$ERDF
5627 016544 000004          .WORD  4
5628 016546 002732          .WORD  MSGMSR

```

5629	016550	004052				.WORD	ALINFO	
5630	016552				10\$:	: CSEG		
5631	016552				10001\$:			
5632	016552	104405				TRAP	C\$ESEG	
5633								
5634								:UPDATE MEMORY SIMULATOR ADDRESS TO NEXT ADDRESS TO BE LOADED
5635								:COMPLEMENT THE DATA PATTERN FOR NEXT LOAD TO RAM
5636								
5637	016554	005103				COM	R3	:COMPLEMENT THE DATA
5638	016556	062737	000002	002276		ADD	#2,R4LOAD	:UPDATE ADDRESS BY 2
5639	016564	005302				DEC	R2	:DECREMENT 2K WORD COUNTER
5640	016566	001333				BNE	7\$:IF NOT DONE DO NEXT ADDRESS
5641								
5642								:UPDATE MODULE SELECT RAM ENABLE BITS TO BE LOADED INTO MODULE
5643								:SELECT RAM 0 ON NEXT PASS OF THIS TEST
5644								
5645	016570	006301				ASL	R1	:MOVE MODULE SELECT RAM 0 DATA PATTERN
5646								:LEFT TO ENABLE NEXT MEMORY SIMULATOR RAM
5647	016572	032701	000020			BIT	#BIT4,R1	:CHECK IF ALL RAMS HAVE BEEN TESTED
5648	016576	001002				BNE	11\$:IF YES THEN EXIT THE TEST
5649	016600	000137	016260			JMP	1\$:OTHERWISE DO NEXT RAM
5650								
5651								:THE FOLLOWING SECTION OF CODE WILL READ THE MEMORY SIMULATOR RAMS
5652								:AND CHECK EACH ADDRESS TO CONTAIN EITHER 125252 OR 052525.
5653								
5654	016604	005037	002276		11\$:	CLR	R4LOAD	:RESET THE ADDRESS TO 0
5655	016610	012703	125252			MOV	#125252,R3	:RESET THE INITIAL DATA PATTERN
5656								
5657	016614				12\$:	BGNSEG		
5658	016614	104404				TRAP	C\$BSEG	
5659								
5660								:LOAD THE ADDRESS TO BE CHECKED INTO CONTROL REGISTER 4. THE MODULE
5661								:SELECT RAMS HAVE BEEN INITIALIZED PREVIOUSLY.
5662								
5663	016616	004737	005462			JSR	PC,LDRDR4	:GO LOAD, READ AND CHECK REGISTER 4
5664	016622	001405				BEQ	13\$:IF LOADED OK THEN CONTINUE
5665	016624					ERRDF	3,,R4EROR	:REGISTER 4 NOT EQUAL EXPECTED.
5666	016624	104455				TRAP	C\$ERDF	
5667	016626	000003				.WORD	3	
5668	016630	000000				.WORD	0	
5669	016632	004000				.WORD	R4EROR	
5670	016634					CKLOOP		
5671	016634	104406				TRAP	C\$CLP1	
5672								

```
5673
5674
5675
5676 016636 010337 002302 002304 13$: MOV R3,R6LOAD ;SETUP DATA THAT WAS PREVIOUSLY WRITTEN
5677 016642 013737 002302 002304 MOV R6LOAD,R6GOOD ;SETUP EXPECTED DATA
5678 016650 005037 002306 CLR R6MASK ;SET REGISTER 6 MASK TO ZERO
5679 016654 004737 005522 JSR PC,READR6 ;GO READ AND CHECK THE RAM LOCATION
5680 016660 001404 BEQ 14$ ;IF DATA EQUALS ADDRESS THEN CONTINUE
5681 016662 ERRDF 4,MSGMSR,ALINFO ;DATA ERROR IN MEMORY SIMULATOR RAM
5682 016662 104455 TRAP C$ERDF
5683 016664 000004 .WORD 4
5684 016666 002732 .WORD MSGMSR
5685 016670 004052 .WORD ALINFO
5686 016672 14$: ENDSEG
5687 016672 10002$: TRAP C$ESEG
5688 016672 104405
5689
5690 016674 005103 COM R3 ;COMPLEMENT THE DATA PATTERN
5691 016676 062737 000002 002276 ADD #2,R4LOAD ;UPDATE THE ADDRESS TO THE NEXT LOCATION
5692 016704 100343 BPL 12$ ;IF NOT DONE CHECK NEXT ADDRESS
5693
5694 016706 ENDTST
5695 016706 L10060:
5696 016706 104401 TRAP C$ETST
5697
```

5698
5699
5700
5701
5702
5703
5704
5705
5706
5707
5708
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751
5752
5753

016710
016710
016710 004737 004550
016714 005037 002276
016720 012701 000001

016724
016724 104404

016726 012737 000004 002264
016734 012737 177740 002270
016742 004737 005414
016746 001405
016750
016750 104455
016752 000002
016754 000000
016756 003712
016760
016760 104406

.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 THE 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

T26:: BGNTST

JSR PC,INITMS ;SELECT AND INIT MEMORY SIMULATOR
CLR R4LOAD ;SET 1ST ADDRESS TO BE TESTED TO 0
MOV #BIT0,R1 ;SETUP DATA FOR MODULE SELECT RAM 0

1\$: BGNSEG
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.

MOV #MSEL0,R2LOAD ;SETUP BITS TO BE LOADED
MOV #177740,R2MASK ;SETUP REGISTER 2 MASK WORD
JSR PC,LDRDR2 ;GO LOAD,READ AND CHECK REG 2
BEQ 2\$;IF LOADED OK THEN CONT
ERRDF 2,R2EROR ;REGISTER 2 NOT EQUAL EXPECTED
TRAP C\$ERDF
.WORD 2
.WORD 0
.WORD R2EROR
CKLOOP
TRAP C\$CLP1

```

5754
5755
5756
5757
5758 016762 004737 005462 2$: JSR PC,LDRDR4 ;GO LOAD,READ AND CHECK REGISTER 4
5759 016766 001405 BEQ 3$ ;IF LOADED OK THEN CONTINUE
5760 016770 ERRDF 3,,R4EROR ;REGISTER 4 NOT EQUAL EXPECTED
5761 016770 104455 TRAP C$ERDF
5762 016772 000003 .WORD 3
5763 016774 000000 .WORD 0
5764 016776 004000 .WORD R4EROR
5765 017000 CKLOOP
5766 017000 104406 TRAP C$CLP1
5767
5768 ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 0 AND CHECK THAT IS WAS
5769 ;WRITTEN CORRECTLY. THE DATA PATTERNS THAT WILL BE WRITTEN ARE 1, 2, 4,
5770 ;10. THESE PATTERNS WILL ENABLE THE SIGNALS EN0 H, EN1 H, EN2 H, AND
5771 ;EN3 H RESPECTIVELY. THESE SIGNALS ARE USED TO SELECT MEMORY SIMULATOR
5772 ;MEMORY WHEN THE SIGNAL SSM L IS ASSERTED LATER ON IN THE TEST.
5773
5774 017002 010137 002302 3$: MOV R1,R6LOAD ;GET THE PATTERN TO BE LOADED
5775 017006 012737 177760 002306 MOV #177760,R6MASK ;SETUP REGISTER 6 MASK WORD
5776 017014 004737 005506 JSR PC,LDRDR6 ;GO LOAD, READ AND CHECK RAM 0
5777 017020 001405 BEQ 4$ ;IF OK THEN CONTINUE
5778 017022 ERRDF 4,MSGMDO,ALINFO ;DATA ERROR IN MODULE SELECT RAM 0
5779 017022 104455 TRAP C$ERDF
5780 017024 000004 .WORD 4
5781 017026 002513 .WORD MSGMDO
5782 017030 004052 .WORD ALINFO
5783 017032 CKLOOP
5784 017032 104406 TRAP C$CLP1
5785
5786 ;SET SIGNALS MSEL0 H AND MSEL1 H TO A ONE AND SIGNALS MSAD17 H AND MSAD16 H
5787 ;TO A ZERO IN CONTROL REGISTER 2. MSEL0 H AND MSEL1 H ON A ONE, WILL
5788 ;CAUSE THE SIGNAL SMDS1 L TO BE ASSERTED ON A WRITE OR READ TO CONTROL
5789 ;REGISTER 6. SMDS1 L WILL SELECT MODULE SELECT RAM 1. MSAD BITS 17 AND 16
5790 ;ON A ZERO WILL SELECT ADDRESS 0 OF MODULE SELECT RAM 1.
5791
5792 017034 012737 000014 002264 4$: MOV #MSEL0!MSEL1,R2LOAD ;SETUP BITS TO BE LOADED
5793 017042 004737 005414 JSR PC,LDRDR2 ;GO LOAD, READ AND CHECK REGISTER 2
5794 017046 001405 BEQ 5$ ;IF LOADED OK THEN CONTINUE
5795 017050 ERRDF 2,,R2EROR ;REGISTER 2 NOT EQUAL EXPECTED
5796 017050 104455 TRAP C$ERDF
5797 017052 000002 .WORD 2
5798 017054 000000 .WORD 0
5799 017056 003712 .WORD R2EROR
5800 017060 CKLOOP
5801 017060 104406 TRAP C$CLP1
5802
5803 ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 1 AND CHECK THAT THE
5804 ;PATTERN WAS LOADED CORRECTLY. A DATA PATTERN OF ALL ONES WILL BE
5805 ;WRITTEN, HOWEVER, WHEN THE LOCATION IS READ BACK, A DATA PATTERN
5806 ;EQUAL TO ONE WILL BE READ. SETTING THE LEAST SIGNIFICANT BIT IN
5807 ;THE DATA PATTERN WILL FORCE THE MORE SIGNIFICANT BITS TO ZERO.
5808
5809 017062 012737 000017 002302 5$: MOV #BIT3!BIT2!BIT1!BIT0,R6LOAD ;GET DATA PATTERN TO BE LOADED

```

```

5810 017070 012737 000001 002304      MOV      #BIT0,R6GOOD      ;SETUP EXPECTED DATA PATTERN
5811 017076 004737 005514              JSR      PC,LDRD6R        ;GO LOAD,READ AND CHECK RAM 1
5812 017102 001405                      BEQ      6$              ;IF DATA OK THEN CONTINUE
5813 017104                      ERRDF   4,MSGMD1,ALINFO   ;DATA ERROR IN MUDULE SELECT RAM 1
5814 017104 104455                      TRAP    C$ERDF
5815 017106 000004                      .WORD   4
5816 017110 002555                      .WORD   MSGMD1
5817 017112 004052                      .WORD   ALINFO
5818 017114                      CKLOOP
5819 017114 104406                      TRAP    C$CLP1
5820
5821                      ;CLEAR BITS MSEL0 H, MSEL1 H, MSAD17 H AND MSAD16 H IN CONTROL REGISTER
5822                      ;2. MSEL0 H AND MSEL1 H ON A ZERO WILL CAUSE THE SIGNAL SSM L TO BE
5823                      ;ASSERTED ON A WRITE OR READ TO CONTROL REGISTER 6. ON A WRITE OR READ
5824                      ;TO CONTROL REGISTER 6, DATA WILL BE WRITTEN OR READ FROM THE MEMORY
5825                      ;SIMULATOR RAM SELECTED. THE SIMULATOR MEMORY IS SELECTED VIA THE
5826                      ;MODULE SELECT RAM'S (0 AND 1).
5827
5828 017116 005037 002264      6$:    CLR      R2LOAD      ;SETUP TO CLEAR REGISTER 2
5829 017122 004737 005414      JSR      PC,LDRDR2      ;GO LOAD, READ AND CHECK REG 2
5830 017126 001404                      BEQ      7$              ;IF LOADED OK THEN CONTINUE
5831 017130                      ERRDF   2,,R2EROR      ;REGISTER 2 NOT EQUAL EXPECTED
5832 017130 104455                      TRAP    C$ERDF
5833 017132 000002                      .WORD   2
5834 017134 000000                      .WORD   0
5835 017136 003712                      .WORD   R2EROR
5836 017140      7$:    ENDSEG
5837 017140      10000$:
5838 017140 104405                      TRAP    C$ESEG
5839
5840 017142 013705 002242      MOV      REG6,R5        ;PUT THE ADDRESS OF CONTROL REGISTER
5841                      ;6 INTO R5
5842
5843 017146      8$:    BGNSEG
5844 017146 104404                      TRAP    C$BSEG
5845
5846                      ;WRITE DATA PATTERN OF 125 INTO LOW BYTE OF MEMORY SIMULATOR ADDRESS
5847
5848 017150 005037 002306      CLR      R6MASK        ;CLEAR REGISTER 6 MASK WORD
5849 017154 012737 000125 002302      MOV      #125,R6LOAD    ;SETUP BYTE TO BE LOADED
5850 017162 013737 002302 002304      MOV      R6LOAD,R6GOOD  ;SETUP EXPECTED DATA PATTERN
5851 017170 005037 002310      CLR      R6READ        ;CLEAR DATA TO BE READ (HIGH BYTE)
5852 017174 113715 002302      MOVB    R6LOAD,(R5)    ;WRITE LOW BYTE INTO MEM SIM RAM
5853 017200 111537 002310      MOVB    (R5),R6READ    ;READ THE LOW BYTE BACK FROM RAM
5854 017204 013737 002310 002312      MOV      R6READ,R6BAD  ;COPY DATA FOR POSSIBLE ERROR REPORT
5855 017212 023737 002304 002312      CMP     R6GOOD,R6BAD   ;CHECK BYTE WRITTEN AGAINST WORD READ
5856 017220 001405                      BEQ     9$              ;IF LOW BYTE OK THEN CONTINUE
5857 017222                      ERRDF   4,MSGMSR,ALINFO ;DATA ERROR READING LOW BYTE OF RAM
5858 017222 104455                      TRAP    C$ERDF
5859 017224 000004                      .WORD   4
5860 017226 002732                      .WORD   MSGMSR
5861 017230 004052                      .WORD   ALINFO
5862 017232                      CKLOOP
5863 017232 104406                      TRAP    C$CLP1
5864
5865                      ;WRITE DATA PATTERN OF 252 INTO HIGH BYTE OF MEMORY SIMULATOR ADDRESS

```

```

5866
5867 017234 012737 000252 002302 9$: MOV #252,R6LOAD ;SETUP DATA PATTERN TO BE LOADED
5868 017242 013737 002302 002304 MOV R6LOAD,R6GOOD ;SETUP EXPECTED DATA
5869 017250 005037 002310 CLR R6READ ;CLEAR DATA WORD TO BE READ
5870 017254 113765 002304 000001 MOVB R6GOOD,1(R5) ;WRITE HIGH BYTE INTO MEM SIM RAM
5871 017262 116537 000001 002310 MOVB 1(R5),R6READ ;READ HIGH BYTE BACK FROM RAM
5872 017270 013737 002310 002312 MOV R6READ,R6BAD ;COPY DATA FOR POSSIBLE ERROR REPORT
5873 017276 023737 002304 002312 CMP R6GOOD,R6BAD ;CHECK BYTE WRITTEN AGAINST BYTE READ
5874 017304 001405 BEQ 10$ ;IF HIGH BYTE OK THEN CONTINUE
5875 017306 ERRDF 4,MSGMSR,ALINFO ;DATA ERROR READING HIGH BYTE OF RAM
5876 017306 104455 TRAP C$ERDF
5877 017310 000004 .WORD 4
5878 017312 002732 .WORD MSGMSR
5879 017314 004052 .WORD ALINFO
5880 017316 CKLOOP
5881 017316 104406 TRAP C$CLP1
5882
5883 ;READ THE RAM LOCATION AS A WORD CHECKING THAT THE LOW AND HIGH BYTE
5884 ;WERE WRITTEN INTO THE CORRECT BYTE. THE WORD SHOULD BE 125125 IF THE
5885 ;BYTES WERE WRITTEN CORRECTLY
5886
5887 017320 012737 000125 002302 10$: MOV #125,R6LOAD ;SETUP LOW BYTE THAT WAS WRITTEN
5888 017326 112737 000252 002303 MOVB #252,R6LOAD+1 ;SETUP HIGH BYTE THAT WAS WRITTEN
5889 017334 013737 002302 002304 MOV R6LOAD,R6GOOD ;SETUP EXPECTED DATA PATTERN
5890 017342 004737 005522 JSR PC,READR6 ;GO READ AND CHECK RAM LOCATION AS WORD
5891 017346 001405 BEQ 11$ ;IF LOW AND HIGH BYTE OK THEN CONTINUE
5892 017350 ERRDF 4,MSGMSR,ALINFO ;DATA ERROR - BYTE OPERATION FAILED
5893 017350 104455 TRAP C$ERDF
5894 017352 000004 .WORD 4
5895 017354 002732 .WORD MSGMSR
5896 017356 004052 .WORD ALINFO
5897 017360 CKLOOP
5898 017360 104406 TRAP C$CLP1
5899
5900 ;WRITE DATA PATTERN OF 252 INTO LOW BYTE OF MEMORY SIMULATOR ADDRESS
5901
5902 017362 012737 000252 002302 11$: MOV #252,R6LOAD ;SETUP DATA TO BE LOADED
5903 017370 013737 002302 002304 MOV R6LOAD,R6GOOD ;SETUP EXPECTED DATA PATTERN
5904 017376 005037 002310 CLR R6READ ;CLEAR DATA WORD TO BE READ
5905 017402 113715 002302 MOVB R6LOAD,(R5) ;WRITE LOW BYTE INTO RAM LOCATION
5906 017406 111537 002310 MOVB (R5),R6READ ;READ LOW BYTE BACK FROM RAM LOCATION
5907 017412 013737 002310 002312 MOV R6READ,R6BAD ;COPY DATA READ FOR POSSIBLE ERROR
5908 017420 023737 002304 002312 CMP R6GOOD,R6BAD ;CHECK BYTE WRITTEN AGAINST BYTE READ
5909 017426 001405 BEQ 12$ ;IF DATA BYTE OK THEN CONTINUE
5910 017430 ERRDF 4,MSGMSR,ALINFO ;DATA ERROR READING LOW BYTE OF RAM
5911 017430 104455 TRAP C$ERDF
5912 017432 000004 .WORD 4
5913 017434 002732 .WORD MSGMSR
5914 017436 004052 .WORD ALINFO
5915 017440 CKLOOP
5916 017440 104406 TRAP C$CLP1
5917
5918 ;READ RAM LOCATION AS A WORD CHECKING THE DATA PATTERN TO BE 125252
5919
5920 017442 012737 000252 002302 12$: MOV #252,R6LOAD ;BYTE WHICH WAS JUST LOADED
5921 017450 112737 000252 002303 MOVB #252,R6LOAD+1 ;BYTE THAT WAS PREVIOUSLY WRITTEN
    
```



```

5922 017456 013737 002302 002304      MOV      R6LOAD,R6GOOD      ;SETUP EXPECTED DATA PATTERN
5923 017464 004737 005522              JSR      PC,READR6          ;GO READ AND CHECK RAM LOC AS A WORD
5924 017470 001405              BEQ      13$                ;IF DATA OK THEN CONTINUE
5925 017472              ERRDF   4,MSGMSR,ALINFO     ;WRITING LOW BYTE CHANGED HIGH BYTE
5926 017472 104455              TRAP    C$ERDF
5927 017474 000004              .WORD   4
5928 017476 002732              .WORD   MSGMSR
5929 017500 004052              .WORD   ALINFO
5930 017502              CKLOOP
5931 017502 104406              TRAP    C$CLP1
5932
5933              ;WRITE DATA PATTERN OF 125 INTO HIGH BYTE OF MEMORY SIMULATOR ADDRESS
5934
5935 017504 012737 000125 002302 13$:      MOV      #125,R6LOAD        ;SETUP DATA PATTERN TO BE LOADED
5936 017512 013737 002302 002304      MOV      R6LOAD,R6GOOD     ;SETUP EXPECTED DATA
5937 017520 005037 002310              CLR      R6READ            ;CLEAR DATA WORD TO BE READ
5938 017524 113765 002302 000001      MOVB    R6LOAD,1(R5)       ;WRITE HIGH BYTE INTO RAM LOCATION
5939 017532 116537 000001 002310      MOVB    1(R5),R6READ       ;READ HIGH BYTE BACK FROM RAM LOCATION
5940 017540 013737 002310 002312      MOV      R6?READ,R6BAD    ;COPY DATA READ FOR POSSIBLE ERROR
5941 017546 023737 002304 002312      CMP     R6GOOD,R6BAD      ;CHECK BYTE WRITTEN AGAINST BYTE READ
5942 017554 001405              BEQ      14$                ;IF HIGH BYTE OK THEN CONTINUE
5943 017556              ERRDF   4,MSGMSR,ALINFO     ;DATA ERROR READING HIGH BYTE
5944 017556 104455              TRAP    C$ERDF
5945 017560 000004              .WORD   4
5946 017562 002732              .WORD   MSGMSR
5947 017564 004052              .WORD   ALINFO
5948 017566              CKLOOP
5949 017566 104406              TRAP    C$CLP1
5950
5951              ;READ RAM LOCATION CHECKING DATA PATTERN TO BE 052652
5952
5953 017570 012737 000252 002302 14$:      MOV      #252,R6LOAD       ;SETUP LOW BYTE THAT WAS WRITTEN
5954 017576 112737 000125 002303      MOVB    #125,R6LOAD+1     ;SETUP HIGH BYTE THAT WAS WRITTEN
5955 017604 013737 002302 002304      MOV      R6LOAD,R6GOOD    ;SETUP EXPECTED WORD TO BE READ
5956 017612 004737 005522              JSR      PC,READR6        ;GO READ RAM LOCATION AS A WORD
5957 017616 001404              BEQ      15$                ;IF DATA OK THEN CONTINUE
5958 017620              ERRDF   4,MSGMSR,ALINFO     ;WRITING HIGH BYTE CHANGED LOW BYTE
5959 017620 104455              TRAP    C$ERDF
5960 017622 000004              .WORD   4
5961 017624 002732              .WORD   MSGMSR
5962 017626 004052              .WORD   ALINFO
5963 017630              15$:      ENDSEG
5964 017630              10001$:
5965 017630 104405              TRAP    C$ESEG
5966
5967 017632 006301              ASL     R1                  ;UPDATE MODULE SELECT RAM 0 DATA
5968 017634 062737 020000 002276      ADD     #MSAD13,R4LOAD    ;UPDATE ADDRESS TO 0 OF NEXT RAM
5969 017642 100402              BMI     16$                ;IF ADDRESS 0 OF EACH RAM DONE - EXIT
5970 017644 000137 016724              JMP     1$                  ;RETURN TO WRITE MODULE SELECT RAM AND
5971
5972 017650              16$:      ENDTST
5973 017650              L10061:
5974 017650 104401              TRAP    C$ETST
5975

```



```

6032
6033
6034           ;LOAD THE ADDRESS TO BE TESTED INTO CONTROL REGISTER 4. THE ADDRESS
6035           ;WILL APPEAR ON MSAD BITS 15-0 WHICH IS THE OUTPUT OF CONTROL REGISTER 4
6036 017724 004737 005462      2$: JSR    PC,LDRDR4           ;GO LOAD,READ AND CHECK REGISTER 4
6037 017730 001405           BEQ    3$                ;IF LOADED OK THEN CONTINUE
6038 017732           ERRDF  3,,R4EROR      ;REGISTER 4 NOT EQUAL EXPECTED
6039 017732 104455           TRAP   C$ERDF
6040 017734 000003           .WORD 3
6041 017736 000000           .WORD 0
6042 017740 004000           .WORD R4EROR
6043 017742           CKLOOP
6044 017742 104406           TRAP   C$CLP1
6045
6046           ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 0 AND CHECK THAT IS WAS
6047           ;WRITTEN CORRECTLY. THE DATA PATTERNS THAT WILL BE WRITTEN ARE 1, 2, 4,
6048           ;10. THESE PATTERNS WILL ENABLE THE SIGNALS EN0 H, EN1 H, EN2 H, AND
6049           ;EN3 H RESPECTIVELY. THESE SIGNALS ARE USED TO SELECT MEMORY SIMULATOR
6050           ;MEMORY WHEN THE SIGNAL SSM L IS ASSERTED LATER ON IN THE TEST.
6051
6052 017744 010137 002302      3$: MOV    R1,R6LOAD           ;GET THE PATTERN TO BE LOADED
6053 017750 012737 177760 002306 MOV    #177760,R6MASK      ;SETUP REGISTER 6 MASK WORD
6054 017756 004737 005506      JSR    PC,LDRDR6           ;GO LOAD, READ AND CHECK RAM 0
6055 017762 001405           BEQ    4$                ;IF OK THEN CONTINUE
6056 017764           ERRDF  4,MSGMDO,ALINFO  ;DATA ERROR IN MODULE SELECT RAM 0
6057 017764 104455           TRAP   C$ERDF
6058 017766 000004           .WORD 4
6059 017770 002513           .WORD MSGMDO
6060 017772 004052           .WORD ALINFO
6061 017774           CKLOOP
6062 017774 104406           TRAP   C$CLP1
6063
6064           ;SET SIGNALS MSEL0 H AND MSEL1 H TO A ONE AND SIGNALS MSAD17 H AND MSAD16 H
6065           ;TO A ZERO IN CONTROL REGISTER 2. MSEL0 H AND MSEL1 H ON A ONE, WILL
6066           ;CAUSE THE SIGNAL SMDS1 L TO BE ASSERTED ON A WRITE OR READ TO CONTROL
6067           ;REGISTER 6. SMDS1 L WILL SELECT MODULE SELECT RAM 1. MSAD BITS 17 AND 16
6068           ;ON A ZERO WILL SELECT ADDRESS 0 OF MODULE SELECT RAM 1.
6069
6070 017776 012737 000014 002264 4$: MOV    #MSEL0!MSEL1,R2LOAD  ;SETUP BITS TO BE LOADED
6071 020004 004737 005414      JSR    PC,LDRDR2           ;GO LOAD, READ AND CHECK REGISTER 2
6072 020010 001405           BEQ    5$                ;IF LOADED OK THEN CONTINUE
6073 020012           ERRDF  2,,R2EROR      ;REGISTER 2 NOT EQUAL EXPECTED
6074 020012 104455           TRAP   C$ERDF
6075 020014 000002           .WORD 2
6076 020016 000000           .WORD 0
6077 020020 003712           .WORD R2EROR
6078 020022           CKLOOP
6079 020022 104406           TRAP   C$CLP1
6080
6081           ;WRITE THE DATA PATTERN INTO MODULE SELECT RAM 1 AND CHECK THAT THE
6082           ;PATTERN WAS LOADED CORRECTLY. A DATA PATTERN OF ALL ONES WILL BE
6083           ;WRITTEN, HOWEVER, WHEN THE LOCATION IS READ BACK, A DATA PATTERN
6084           ;EQUAL TO ONE WILL BE READ. SETTING THE LEAST SIGNIFICANT BIT IN
6085           ;THE DATA PATTERN WILL FORCE THE MORE SIGNIFICANT BITS TO ZERO.
6086
6087 020024 012737 000017 002302 5$: MOV    #BIT3!BIT2!BIT1!BIT0,R6LOAD ;GET DATA PATTERN TO BE LOADED

```

```

6088 020032 012737 000001 002304      MOV      #BIT0,R6GOOD      ;SETUP EXPECTED DATA PATTERN
6089 020040 004737 005514              JSR      PC,LDRD6R        ;GO LOAD,READ AND CHECK RAM 1
6090 020044 001405              BEQ      6$              ;IF DATA OK THEN CONTINUE
6091 020046              ERRDF    4,MSGMD1,ALINFO  ;DATA ERROR IN MUDULE SELECT RAM 1
6092 020046 104455              TRAP    C$ERDF
6093 020050 000004              .WORD   4
6094 020052 002555              .WORD   MSGMD1
6095 020054 004052              .WORD   ALINFO
6096 020056              CKLOOP
6097 020056 104406              TRAP    C$CLP1
6098
6099
6100
6101
6102
6103
6104
6105
6106 020060 005037 002264      6$:      CLR      R2LOAD          ;SETUP TO CLEAR REGISTER 2
6107 020064 004737 005414              JSR      PC,LDRDR2        ;GO LOAD, READ AND CHECK REG 2
6108 020070 001404              BEQ      7$              ;IF LOADED OK THEN CONTINUE
6109 020072              ERRDF    2,R2EROR        ;REGISTER 2 NOT EQUAL EXPECTED
6110 020072 104455              TRAP    C$ERDF
6111 020074 000002              .WORD   2
6112 020076 000000              .WORD   0
6113 020100 003712              .WORD   R2EROR
6114 020102              7$:      ENDSEG
6115 020102              10000$:
6116 020102 104405              TRAP    C$ESEG
6117
6118 020104              8$:      BGNSEG
6119 020104 104404              TRAP    C$BSEG
6120
6121
6122
6123 020106 042737 000010 002252      ;SET MEMORY SIMULATOR TO 16 BIT MODE
6124 020114 004737 005346              BIC      #BIT8H,ROLOAD    ;SETUP TO CLEAR 8 BIT MODE IF SET
6125 020120 001405              JSR      PC,LDRDR0        ;GO LOAD, READ AND CHECK REG 0
6126 020122              BEQ      9$              ;IF LOADED OK THEN CONTINUE
6127 020122 104455              ERRDF    1,ROEROR        ;REGISTER 0 NOT EQUAL EXPECTED
6128 020124 000001              TRAP    C$ERDF
6129 020126 000000              .WORD   1
6130 020130 003624              .WORD   0
6131 020132              .WORD   ROEROR
6132 020132 104406              CKLOOP
6133              TRAP    C$CLP1
6134
6135
6136 020134 042737 000001 002276 9$:      ;LOAD THE ADDRESS TO BE TESTED INTO CONTROL REGISTER 4
6137 020142 004737 005462              BIC      #BIT0,R4LOAD    ;CLEAR ODD ADDRESS BIT IF SET
6138 020146 001405              JSR      PC,LDRDR4        ;GO LOAD, READ AND CHECK REGISTER 4
6139 020150              BEQ      10$             ;IF LOADED OK THEN CONTINUE
6140 020150 104455              ERRDF    3,R4EROR        ;REGISTER 4 NOT EQUAL EXPECTED
6141 020152 000003              TRAP    C$ERDF
6142 020154 000000              .WORD   3
6143 020156 004000              .WORD   0
              .WORD   R4EROR

```

```

6144 020160          CKLOOP
6145 020160 104406  TRAP   C$CLP1
6146
6147          ;WRITE DATA PATTERN OF 125252 INTO MEMORY SIMULATOR RAM ADDRESSES
6148          ;BY CONTROL REGISTER 4
6149
6150 020162 005037 002306 10$: CLR   R6MASK          ;CLEAR REGISTER 6 MASK WORD
6151 020166 012737 125252 002302 MOV   #125252,R6LOAD  ;SETUP WORD TO BE LOADED
6152 020174 004737 005506 JSR   PC,LDRDR6      ;GO LOAD, READ AND CHECK RAM LOCATION
6153 020200 001405 BEQ   11$            ;IF DATA OK THEN CONTINUE
6154 020202 ERRDF 4,MSGMSR,ALINFO ;DATA ERROR IN MEMORY SIMULATOR RAM
6155 020202 104455 TRAP  C$ERDF
6156 020204 000004 .WORD 4
6157 020206 002732 .WORD MSGMSR
6158 020210 004052 .WORD ALINFO
6159 020212 CKLOOP
6160 020212 104406 TRAP  C$CLP1
6161
6162          ;SET MEMORY SIMULATOR TO 8 BIT MODE
6163
6164 020214 052737 000010 002252 11$: BIS   #BIT8H,ROLOAD  ;SETUP TO SET 8 BIT MODE
6165 020222 004737 005346 JSR   PC,LDRDR0      ;GO LOAD, READ AND CHECK REGISTER 0
6166 020226 001405 BEQ   12$            ;IF LOADED OK THEN CONTINUE
6167 020230 ERRDF 1,,ROEROR    ;REGISTER 0 NOT EQUAL EXPECTED
6168 020230 104455 TRAP  C$ERDF
6169 020232 000001 .WORD 1
6170 020234 000000 .WORD 0
6171 020236 003624 .WORD ROEROR
6172 020240 CKLOOP
6173 020240 104406 TRAP  C$CLP1
6174
6175          ;WRITE DATA PATTERN OF 031463 INTO MEMORY SIMULATOR RAM. ONLY THE
6176          ;LOW BYTE 063 SHOULD BE WRITTEN WHEN THE ADDRESS SELECTED IS EVEN.
6177
6178 020242 012737 031463 002302 12$: MOV   #31463,R6LOAD  ;SETUP BITS TO BE WRITTEN
6179 020250 012737 000063 002304 MOV   #63,R6GOOD     ;SETUP EXPECTED DATA TO BE READ
6180 020256 012737 177400 002306 MOV   #177400,R6MASK ;SETUP TO IGNORE HIGH BYTE ON READ
6181 020264 004737 005514 JSR   PC,LDRDR6      ;GO LOAD, READ AND CHECK LOW BYTE
6182 020270 001405 BEQ   13$            ;IF LOW BYTE OK THEN CONTINUE
6183 020272 ERRDF 4,MSGMSR,ALINFO ;8 BIT MODE FAILED TO WRITE OR READ
6184 020272 104455 TRAP  C$ERDF
6185 020274 000004 .WORD 4
6186 020276 002732 .WORD MSGMSR
6187 020300 004052 .WORD ALINFO
6188 020302 CKLOOP
6189 020302 104406 TRAP  C$CLP1
6190          ;LOW BYTE OF MEMORY SIMULATOR RAM
6191
6192          ;RFSET MEMORY SIMULATOR TO 16 BIT MODE
6193
6194 020304 042737 000010 002252 13$: BIC   #BIT8H,ROLOAD  ;SETUP TO CLEAR 8 BIT MODE
6195 020312 004737 005346 JSR   PC,LDRDR0      ;GO LOAD, READ AND CHECK REGISTER 0
6196 020316 001405 BEQ   14$            ;IF LOADED OK THEN CONTINUE
6197 020320 ERRDF 1,,ROEROR    ;REGISTER 0 NOT EQUAL EXPECTED
6198 020320 104455 TRAP  C$ERDF
6199 020322 000001 .WORD 1

```

```

6200 020324 000000      .WORD 0
6201 020326 003624      .WORD ROEROR
6202 020330
6203 020330 104406      CKLOOP
6204
6205
6206
6207
6208 020332 012737 125063 002302 14$: MOV #125063,R6LOAD      ;SETUP PREVIOUS LOADED BITS
6209 020340 013737 002302 002304 MOV R6LOAD,R6GOOD      ;SETUP EXPECTED DATA
6210 020346 005037 002306 CLR R6MASK              ;SETUP TO CHECK ALL 16 BITS
6211 020352 004737 005522 JSR PC,READR6          ;GO READ AND CHECK 16 BIT WORD
6212 020356 001405 BEQ 15$                ;IF DATA OK THEN CONTINUE
6213 020360 ERRDF 4,MSGMSR,ALINFO ;8 BIT MOD" CHANGED HIGH BYTE
6214 020360 104455 TRAP C$ERDF
6215 020362 000004 .WORD 4
6216 020364 002732 .WORD MSGMSR
6217 020366 004052 .WORD ALINFO
6218 020370 CKLOOP
6219 020370 104406 TRAP C$CLP1
6220
6221
6222
6223 020372 052737 000010 002252 15$: ;RESET MEMORY SIMULATOR TO 8 BIT MODE
6224 020400 004737 005346 JSR #BIT8H,ROLOAD      ;SETUP TO SET 8 BIT MODE
6225 020404 001405 BEQ 16$                ;GO LOAD, READ AND CHECK REG 0
6226 020406 ERRDF 1,,ROEROR ;IF LOADED OK THEN CONTINUE
6227 020406 104455 TRAP C$ERDF ;REGISTER 0 NOT EQUAL EXPECTED
6228 020410 000001 .WORD 1
6229 020412 000000 .WORD 0
6230 020414 003624 .WORD ROEROR
6231 020416 CKLOOP
6232 020416 104406 TRAP C$CLP1
6233
6234
6235
6236
6237 020420 052737 000001 002276 16$: ;SET THE MEMORY SIMULATOR ADDRESS IN CONTROL REGISTER 4 TO AN
6238 020426 004737 005462 BIS #BIT0,R4LOAD ;ODD ADDRESS
6239 020432 001405 JSR PC,LDRDR4 ;SETUP TO SET AN ODD ADDRESS
6240 020434 BEQ 17$                ;GO LOAD, READ AND CHECK REGISTER 4
6241 020434 104455 ERRDF 3,,R4EROR ;IF LOADED OK THEN CONTINUE
6242 020436 000003 TRAP C$ERDF ;REGISTER 4 NOT EQUAL EXPECTED
6243 020440 000000 .WORD 3
6244 020442 004000 .WORD 0
6245 020444 .WORD R4EROR
6246 020444 104406 CKLOOP
6247 TRAP C$CLP1
6248
6249
6250
6251 020446 012737 146314 002302 17$: ;WRITE DATA PATTERN OF 146314 INTO MEMORY SIMULATOR RAM, ONLY THE
6252 020454 012737 146000 002304 ;HIGH BYTE, 314, SHOULD BE WRITTEN WHEN THE ADDRESS IS ODD.
6253 020462 012737 000377 002306 MOV #146314,R6LOAD      ;SETUP DATA TO BE LOADED
6254 020470 004737 005514 MOV #146000,R6GOOD      ;SETUP EXPECTED DATA
6255 020474 001405 MOV #377,R6MASK         ;SETUP MASK TO IGNORE LOW BYTE
JSR PC,LDRD6R          ;GO LOAD, READ AND CHECK HIGH BYTE
BEQ 18$                ;IF HIGH BYTE OK THEN CONTINUE

```

```

6256 020476 ERRDF 4,MSGMSR,ALINFO ;FAILED TO WRITE HIGH BYTE INTO MEMORY
6257 020476 104455 TRAP C$ERDF
6258 020500 000004 .WORD 4
6259 020502 002732 .WORD MSGMSR
6260 020504 004052 .WORD ALINFO
6261 020506 CKLOOP
6262 020506 104406 TRAP C$CLP1
6263 ;SIMULATOR RAM IN 8 BIT MODE
6264
6265 ;RESET THE MEMORY SIMULATOR TO 16 BIT MODE
6266
6267 020510 042737 000010 002252 18$: BIC #BIT8H,R0LOAD ;SETUP TO CLEAR 8 BIT MODE
6268 020516 004737 005346 JSR PC,LDRDRO ;GO LOAD, READ AND CHECK REGISTER 0
6269 020522 001405 BEQ 19$ ;IF LOADED OK THEN CONTINUE
6270 020524 ERRDF 1,,R0EROR ;REGISTER 0 NOT EQUAL EXPECTED DATA
6271 020524 104455 TRAP C$ERDF
6272 020526 000001 .WORD 1
6273 020530 000000 .WORD 0
6274 020532 003624 .WORD R0EROR
6275 020534 CKLOOP
6276 020534 104406 TRAP C$CLP1
6277
6278 ;READ ADDRESS 0 AND 1 OF MEMORY SIMULATOR RAM CHECKING THAT ONLY THE
6279 ;HIGH BYTE WAS CHANGED DURING A WRITE OPERATION IN 8 BIT MODE
6280
6281 020536 012737 146063 002302 19$: MOV #146063,R6LOAD ;SETUP THE DATA PREVIOUSLY WRITTEN
6282 020544 013737 002302 002304 MOV R6LOAD,R6GOOD ;SETUP EXPECTED DATA
6283 020552 005037 002306 CLR R6MASK ;SETUP TO READ ALL BITS
6284 020556 004737 005522 JSR PC,READR6 ;GO READ AND CHECK 16 BIT WORD
6285 020562 001404 BEQ 20$ ;IF DATA OK THEN CONTINUE
6286 020564 ERRDF 4,MSGMSR,ALINFO ;WRITING HIGH BYTE IN 8 BIT MODE CHANGED
6287 020564 104455 TRAP C$ERDF
6288 020566 000004 .WORD 4
6289 020570 002732 .WORD MSGMSR
6290 020572 004052 .WORD ALINFO
6291
6292 020574 042737 000001 002276 20$: BIC #BIT0,R4LOAD ;LOW BYTE OF DATA
6293 020602 ENDSEG ;RESET ADDRESS TO BE LOADED TO 0
6294 020602 10001$:
6295 020602 104405 TRAP C$ESEG
6296
6297 020604 006301 ASL R1 ;UPDATE MODULE SELECT RAM 0 DATA PATTERN
6298 020606 062737 020000 002276 ADD #MSAD13,R4LOAD ;UPDATE ADDRESS TO NEXT MEM SIM RAM
6299 020614 100402 BMI 21$ ;IF DONE - THEN EXIT
6300 020616 000137 017666 JMP 1$ ;GO SETUP FOR NEXT MEMORY SIMULATOR RAM
6301 020622 21$:
6302 020622 L10062:
6303 020622 104401 TRAP C$ETST
  
```

```

6304
6305
6306
6307
6308
6309
6310
6311
6312
6313
6314
6315 020624
6316 020624
6317 020624 004737 004550
6318
6319 020630
6320 020630
6321 020630 104402
6322
6323
6324
6325 020632
6326 020632 104404
6327 020634 112737 000001 002252
6328 020642 004737 005346
6329 020646 001405
6330 020650
6331 020650 104455
6332 020652 000001
6333 020654 000000
6334 020656 003624
6335 020660
6336 020660 104406
6337 020662 042737 000001 002252 1$:
6338 020670 004737 005346
6339 020674 001405
6340 020676
6341 020676 104455
6342 020700 000001
6343 020702 000000
6344 020704 003624
6345 020706
6346 020706 104406
6347
6348
6349
6350 020710 012737 000010 002264 2$:
6351 020716 012737 177740 002270
6352 020724 004737 005414
6353 020730 001405
6354 020732
6355 020732 104455
6356 020734 000002
6357 020736 000000
6358 020740 003712
6359 020742
  
```

```

.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.
:--

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

T28.1: BGNSUB
      TRAP  C$BSUB

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

      BGNSEG
      TRAP  C$BSEG
      MOVB  #RSTH,ROLOAD      ;SET UP TO SET RST H IN REG 0
      JSR   PC,LDRDR0        ;GO LOAD, READ AND CHECK REG 0
      BEQ   1$              ;IF OK THEN CONTINUE
      ERDF  1,,ROEROR       ;REGISTER NOT EQUAL EXPECTED
      TRAP  C$ERDF
      .WORD 1
      .WORD 0
      .WORD ROEROR
      CKLOOP
      TRAP  C$CLP1
      BIC   #RSTH,ROLOAD      ;SET UP TO CLEAR RST H IN REG 0
      JSR   PC,LDRDR0        ;GO LOAD, READ AND CHECK REG 0
      BEQ   2$              ;IF OK THEN CONTINUE
      ERDF  1,,ROEROR       ;REGISTER NOT EQUAL EXPECTED
      TRAP  C$ERDF
      .WORD 1
      .WORD 0
      .WORD ROEROR
      CKLOOP
      TRAP  C$CLP1

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

      MOV   #MSEL1,R2LOAD    ;SET UP TO SET MSEL1 H TO A ONE
      MOV   #177740,R2MASK   ;SET UP TO READ LOWER 4 BITS
      JSR   PC,LDRDR2        ;GO LOAD, READ AND CHECK REG 2
      BEQ   3$              ;IF OKAY THEN CONTINUE
      ERDF  2,,R2EROR       ;REG 2 NOT EQUAL EXPECTED
      TRAP  C$ERDF
      .WORD 2
      .WORD 0
      .WORD R2EROR
      CKLOOP
  
```



```

6360 020742 104406 TRAP C$CLP1
6361
6362 ;LOAD MSAD BIT 15 THROUGH 8 TO A 0 TO SELECT ADDRESS 0
6363
6364 020744 005037 002276 3$: CLR R4LOAD ;SET UP TO SET ALL BITS TO 0
6365 020750 004737 005462 JSR PC,LDRDR4 ;GO LOAD, READ AND CHECK REG 4
6366 020754 001405 BEQ 4$ ;IF NO ERRORS THEN CONTINUE
6367 020756 ERRDF 3,R4EROR ;REGISTER 4 NOT ALL ZEROS
6368 020756 104455 TRAP C$ERDF
6369 020760 000003 .WORD 3
6370 020762 000000 .WORD 0
6371 020764 004000 .WORD R4EROR
6372 020766 CKLOOP
6373 020766 104406 TRAP C$CLP1
6374
6375 ;SET BITS MPIN H AND MUTB H TO A ONE, AND WRE H AND RDE H TO A ZERO.
6376 ;WRE H AND RDE H ON A ZERO WILL CLEAR(1) THE WRV AND THE RDV FLIP-
6377 ;FLOPS, WHEN BIT CK H IS TOGGLED IN CONTROL REGISTER 0.
6378
6379 020770 012737 000011 002302 4$: MOV #MPINH:MUTBH,R6LOAD ;SET UP TO LOAD THE MAP PROTECT RAM
6380 020776 012737 177760 002306 MOV #177760,R6MASK ;SET UP REG 6 MASK WORD
6381 021004 004737 005506 JSR PC,LDRDR6 ;GO LOAD, READ AND CHECK REG 6
6382 021010 001405 BEQ 5$ ;IF OK THEN CONTINUE
6383 021012 ERRDF 4,MSGMP,ALINFO ;MAP PROTECT RAM DATA ERROR
6384 021012 104455 TRAP C$ERDF
6385 021014 000004 .WORD 4
6386 021016 002356 .WORD MSGMP
6387 021020 004052 .WORD ALINFO
6388 021022 CKLOOP
6389 021022 104406 TRAP C$CLP1
6390
6391 ;SET SIGNAL CK H IN CONTROL REGISTER 0 TO CLOCK THE WRV AND RDV F/F'S
6392 ;BOTH F/F'S SHOULD BE CLEAR(1)
6393
6394 021024 052737 000160 002254 5$: BIS #CKH:WRV:RDV:ROGOOD ;SET UP EXPECTED DATA
6395 021032 052737 000100 002252 BIC #R0LOAD ;SET UP BIT TO BE LOADED
6396 021040 004737 005354 JSR PC,LDRDR ;GO LOAD, READ AND CHECK REG 0
6397 021044 001404 BEQ 6$ ;IF OK THEN CONTINUE
6398 021046 ERRDF 1,MSGMPL,ALRCIN ;WRV F/F OR RDV F/F PROBABLY NOT SET(0)
6399 021046 104455 TRAP C$ERDF
6400 021050 000001 .WORD 1
6401 021052 002463 .WORD MSGMPL
6402 021054 004100 .WORD ALRCIN
6403 021056 68: ENDSEG
6404 021056 100008: TRAP C$ESEG
6405 021056 104405
6406
6407 ;ISSUE A BRESET INSTRUCTION. BOTH RDV AND WRV FLIP-FLOPS SHOULD BE SET
6408
6409 021060 BGNSEG
6410 021060 104404 TRAP C$BSEG
6411 021062 BRESET ;PRESET THE RDV AND WRV F/F'S
6412 021062 104433 TRAP C$BRESET
6413 021064 SETVEC #4,#7$,#PRI07
6414 021064 012746 000340 MOV #PRI07,-(SP)
6415 021070 012746 021136 MOV #7$,-(SP)

```

6416	021074	012746	000004		MOV	#4,-(SP)	
6417	021100	012746	000003		MOV	#3,-(SP)	
6418	021104	104437			TRAP	C\$SVEC	
6419	021106	062706	000010		ADD	#10,SP	
6420	021112	013705	002234		MOV	REG0,R5	:SAVE ADDRESS OF REG 0
6421	021116	113765	002245	000001	MOVB	IDDEV+1,1(R5)	:SAVE ID NUMBER
6422	021124	000240			NOP		
6423	021126				CLRVEC	#4	:RELEASE TIMEOUT VECTOR
6424	021126	012700	000004		MOV	#4,R0	
6425	021132	104436			TRAP	C\$CVEC	
6426	021134	000420			BR	8\$:IF NO DEVICE TIMEOUT THEN CONTINUE
6427							
6428							:A DEVICE TIMEOUT OCCURED WHICH INDICATES THAT THERE IS NO DEVICE 0
6429							:IN THE SYSTEM, THEREFORE, THE MEMORY SIMULATOR HAS TO BE RESELECTED
6430							:BY DOING A 'MOV WORD' OPERATION. A 'MOVB' OPERATION PERFORMED ABOVE
6431							:DOES A READ/MODIFY WRITE. THEREFORE, IF THERE IS NO DEVICE 0 IN THE
6432							:SYSTEM, A DEVICE TIMEOUT WILL OCCUR TO ADDRESS 4.
6433							
6434	021136	005726		7\$:	TST	(SP)+	:CLEAN UP THE STACK FROM TIMEOUT
6435	021140	005726			TST	(SP)+	:
6436	021142				CLRVEC	#4	:RELEASE DEVICE TIMEOUT VECTOR
6437	021142	012700	000004		MOV	#4,R0	
6438	021146	104436			TRAP	C\$CVEC	
6439	021150	105037	002252		CLRB	R0LOAD	:SETUP TO LOAD ALL ZEROS (RESULT OF INIT)
6440	021154	004737	005346		JSR	PC,LDRDRO	:RESELECT THE DEVICE AFTER 'INIT'
6441	021160	001417			BEQ	9\$:IF LOADED OK THEN CONTINUE
6442	021162				ERRDF	1,,ROEROR	: 'INIT' FAILED TO 0 RDV OR WRV F/F'S
6443	021162	104455			TRAP	C\$ERDF	
6444	021164	000001			.WORD	1	
6445	021166	000000			.WORD	0	
6446	021170	003624			.WORD	ROEROR	
6447	021172				CKLOOP		
6448	021172	104406			TRAP	C\$CLP1	
6449	021174	000411			BR	9\$:GO TO END OF SEGMENT IF NO LOOPING
6450							
6451	021176	105037	002254	8\$:	CLRB	R0GOOD	:CLEAR LOWER BYTE OF EXPECTED DATA
6452	021202	004737	005362		JSR	PC,READRO	:GO READ AND CHECK REG 0
6453	021206	001404			BEQ	9\$:IF OK THEN CONTINUE
6454	021210				ERRDF	1,,ROEROR	:REGISTER 0 NOT EQUAL EXPECTED
6455	021210	104455			TRAP	C\$ERDF	
6456	021212	000001			.WORD	1	
6457	021214	000000			.WORD	0	
6458	021216	003624			.WORD	ROEROR	
6459	021220			9\$:	ENDSEG		
6460	021220			10001\$:			
6461	021220	104405			TRAP	C\$ESEG	
6462	021222				ENDSUB		
6463	021222			L10064:			
6464	021222	104403			TRAP	C\$ESUB	
6465							
6466	021224				BGNSUB		
6467	021224			T28.2:			
6468	021224	104402			TRAP	C\$BSUB	
6469							
6470							:CHECK THAT RST H, CTS H, MP H, 8 BIT H AND CK H CAN BE SET TO 1.
6471							:RDV H AND WRV H WILL BE CHECKED TO BE 0.

6472								
6473	021226				BGNSEG			
6474	021226	104404			TRAP	C\$BSEG		
6475	021230	052737	000117	002252	BIS	#RSTH!CTSH!MPH!BIT8H!CKH,ROLOAD	;SET ALL R/W BITS TO 1	
6476	021236	004737	005346		JSR	PC,LDRDRO	;GO LOAD, READ AND COMPARE REG 0	
6477	021242	001404			BEQ	1\$;IF OK THEN CONTINUE	
6478	021244				ERRDF	1,ROEROR	;R/W BITS NOT ALL SET	
6479	021244	104455			TRAP	C\$ERDF		
6480	021246	000C01			.WORD	1		
6481	021250	000000			.WORD	0		
6482	021252	003624			.WORD	ROEROR		
6483	021254				1\$:	ENDSEG		
6484	021254				10000\$:			
6485	021254	104405			TRAP	C\$ESEG		
6486								
6487							;ISSUE A BRESET INSTRUCTION. THE R/W BITS SHOULD THEN BE ZEROS.	
6488								
6489	021256				BGNSEG			
6490	021256	104404			TRAP	C\$BSEG		
6491	021260				BRESET		;CLEAR THE LOW BYTE OF REG 0	
6492	021260	104433			TRAP	C\$RESET		
6493	021262				SETVEC	#4,#2\$,#PRI07	;SETUP INCASE OF DEVICE TIMEOUT	
6494	021262	012746	000340		MOV	#PRI07,-(SP)		
6495	021266	012746	021334		MOV	#2\$,-(SP)		
6496	021272	012746	000004		MOV	#4,-(SP)		
6497	021276	012746	000003		MOV	#3,-(SP)		
6498	021302	104437			TRAP	C\$SVEC		
6499	021304	062706	000010		ADD	#10,SP		
6500	021310	013705	002234		MOV	REG0,R5	;SAVE ADDRESS OF REG 0	
6501	021314	113765	002245	000001	MOVB	IDDEV+1,1(R5)	;SAVE ID NUMBER	
6502	021322	000240			NOP			
6503	021324				CLRVEC	#4	;RELEASE TIMEOUT VECTOR	
6504	021324	012700	000004		MOV	#4,RO		
6505	021330	104436			TRAP	C\$CVEC		
6506	021332	000420			BR	3\$;NO DEVICE TIMEOUT - CONTINUE	
6507								
6508							;A DEVICE TIMEOUT OCCURED WHICH INDICATES THAT THERE IS NO DEVICE #0	
6509							;IN THE SYSTEM, THEREFORE, THE MEMORY SIMULATOR HAS TO BE RESELECTED BY	
6510							;DOING A 'MOV WORD' OPERATION. A 'MOVB' OPERATION PERFORMED ABOVE DOES	
6511							;A READ/MODIFY WRITE. THEREFORE, IF THERE IS NO DEVICE #0 IN THE SYSTEM,	
6512							;A DEVICE TIMEOUT WILL OCCUR TO ADDRESS 4.	
6513								
6514	021334	005726			2\$:	TST	(SP)+	;CLEAN UP THE STACK FROM TIMEOUT
6515	021336	075726				TST	(SP)+	
6516	021340					CLRVEC	#4	;RELEASE TIMEOUT VECTOR
6517	021340	012700	000004		MOV	#4,RO		
6518	021344	104436			TRAP	C\$CVEC		
6519	021346	105037	002252		CLRB	ROLOAD	;SETUP TO LOAD ALL ZEROES	
6520	021352	004737	005346		JSR	PC,LDRDRO	;GO LOAD, READ AND CHECK REG 0	
6521	021356	001417			BEQ	4\$;IF LOADED OK THEN CONTINUE	
6522	021360				ERRDF	1,ROEROR	;REGISTER 0 NOT EQUAL EXPECTED	
6523	021360	104455			TRAP	C\$ERDF		
6524	021362	000001			.WORD	1		
6525	021364	000000			.WORD	0		
6526	021366	003624			.WORD	ROEROR		
6527	021370				CKLOOP			

```

6528 021370 104406          TRAP  C$CLP1
6529 021372 000411          BR    48          ;CONTINUE IF NO LOOPING
6530
6531 021374 105037 002254   38:   CLR  ROGOOD          ;CLEAR LOWER BYTE OF EXPECTED
6532 021400 004737 005362   JSR  PC,READRO        ;GO READ AND CHECK REG 0
6533 021404 001404          BEQ  48          ;IF OK THEN CONTINUE
6534 021406          ERRDF 1,ROEROR        ;REG 0 NOT EQUAL TO EXPECTED
6535 021406 104455          TRAP  C$ERDF
6536 021410 000001          .WORD 1
6537 021412 000000          .WORD 0
6538 021414 003624          .WORD ROEROR
6539 021416          48:   ENDSEG
6540 021416          ?00018:
6541 021416 104405          TRAP  C$ESEG
6542 021420          ENDSUB
6543 021420          L10065:
6544 021420 104403          TRAP  C$ESUB
6545 021422          ENDIST
6546 021422          L10063:
6547 021422 104401          TRAP  C$ETST
6548 021424          ENDMOD
  
```

6549
6550
6551
6552
6553
6554
6555
6556
6557
6558
6559
6560
6561
6562
6563
6564
6565
6566
6567
6568
6569
6570
6571
6572
6573
6574
6575
6576
6577
6578
6579
6580
6581
6582
6583
6584
6585
6586
6587
6588
6589
6590
6591
6592
6593
6594
6595
6596
6597
6598
6599
6600
6601
6602
6603
6604

021424

021424 000011
021424
021426

021426 000031
021426 021450
021430 021450
021432 000000
021434 177777
021436
021436 001032
021440 021464
021442 177777
021444 000000
021446 000017

```
.TITLE PARAMETER CODING
.SBTTL  HARDWARE PARAMETER CODING SECTION
      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.
:--

      BGNHRD
      .WORD L10066-L$HARD/2
L$HARD::

: HARDWARE P-TABLE QUESTIONS
:
: ASK FOR CDS MEMORY SIMULATOR CSR ADDRESS
: ASK FOR CDS MEMORY SIMULATOR DEVICE NUMBER
:
      GPRMA  MSG1,0,0,0,177777,YES
      .WORD  T$CODE
      .WORD  MSG1
      .WORD  T$LLOLIM
      .WORD  T$HILIM
      GPRMD  MSG2,2,0,177777,0,000017,YES
      .WORD  T$CODE
      .WORD  MSG2
      .WORD  177777
      .WORD  T$LLOLIM
      .WORD  T$HILIM

      ENDHRD
      .EVEN
L10066:

: HARDWARE P-TABLEt MESSAGES
:
MSG1:  .ASCIZ  /CSR ADDRESS/
MSG2:  .ASCIZ  /DEVICE NUMBER/

      .EVEN

.SBTTL  SOFTWARE PARAMETER CODING SECTION
```

6605
 6606
 6607
 6608
 6609
 6610
 6611
 6612
 6613
 6614
 6615 021502
 6616 021502 000000
 6617 021504
 6618
 6619
 6620
 6621
 6622 021504
 6623
 6624 021504
 6625
 6626 021504
 6627 021504 000010
 6628
 6629 021524
 6630
 6631 021524 021540
 6632 021526 000004
 6633 021530
 6634 021530
 6635
 6636 021530
 6637 021530
 6638 021530 000000
 6639 021532 000002
 6640 021534
 6641 021534 163010
 6642 021536 000000
 6643 021540
 6644 021540
 6645 021540
 6646 000001

```

:++
: 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.
:--

```

```

      BGNSFT
      .WORD L10067-L$$SOFT/2
L$$SOFT::
      .EVEN
      ENDSFT
      .EVEN
L10067:
      SPATCH::
      .BLKW 10
      LASTAD
      .EVEN
      .WORD T$FREE
      .WORD T$$SIZE
L$LAST::
      ENDMOD
      BGNSETUP 1.
      BGNPTAB
      .WORD 0
      .WORD L10072-./2-1
L10070:
      .WORD 163010
      .WORD 0
      ENDPTAB
L10072:
      ENDSETUP
.END

```

ADR = 000020 G	1345#												
ALINFO 004052 G	1733#	2945	2958	3111	3140	3232	3300	3320	3425	3511	3705	3791	3985
	4185	4271	4349	4382	4413	4445	4516	4571	4642	4686	4781	4816	4869
	4883	4985	5020	5055	5116	5213	5248	5301	5354	5395	5409	5449	5541
	5576	5629	5685	5782	5817	5861	5879	5896	5914	5929	5947	5962	6060
	6095	6158	6187	6217	6260	6290	6387						
ALROIN 004100 G	1746#	3437	3468	3482	3523	3553	3580	3593	3717	3748	3762	3803	3833
	3860	3873	3997	4028	4043	4070	4083	6402					
ALR2IN 004126 G	1759#	3126	3155	3453	3495	3536	3565	3607	3733	3775	3816	3845	3887
	4013	4056	4096										
ASSEMB= 000010	1094												
BIT0 = 000001 G	1318#	1383	1406	1427	1433	4167	4253	4731	4808	4809	4933	5012	5013
	5163	5240	5241	5489	5490	5568	5569	5732	5809	5810	6010	6087	6088
	6136	6237	6292										
BIT00 = 000001 G	1307#	1318											
BIT01 = 000002 G	1306#	1317											
BIT02 = 000004 G	1305#	1316											
BIT03 = 000010 G	1304#	1315											
BIT04 = 000020 G	1303#	1314											
BIT05 = 000040 G	1302#	1313											
BIT06 = 000100 G	1301#	1312											
BIT07 = 000200 G	1300#	1311											
BIT08 = 000400 G	1299#	1310											
BIT09 = 001000 G	1298#	1309											
BIT1 = 000002 G	1317#	1381	1405	1426	1434	4169	4171	4255	4257	4342	4405	4808	5012
	5240	5568	5809	6087									
BIT10 = 002000 G	1297#	1370	1417										
BIT11 = 004000 G	1296#	1369	1416										
BIT12 = 010000 G	1295#	1415											
BIT13 = 020000 G	1294#	1414											
BIT14 = 040000 G	1293#	1413											
BIT15 = 100000 G	1292#	1362	1412										
BIT2 = 000004 G	1316#	1380	1398	1425	1435	4169	4173	4175	4255	4259	4261	4808	5012
	5240	5568	5809	6087									
BIT3 = 000010 G	1315#	1379	1397	1424	1436	4169	4173	4177	4255	4259	4263	4375	4437
	4808	5012	5240	5568	5809	6087							
BIT4 = 000020 G	1314#	1378	1423	2527	4280	4649	4693	4899	5063	5317	5647		
BIT5 = 000040 G	1313#	1377	1393	1422									
BIT6 = 000100 G	1312#	1376	1392	1421									
BIT7 = 000200 G	1311#	1391	1420										
BIT8 = 000400 G	1310#	1372	1419	2792	2820								
BIT8H = 000010 G	1379#	2411	2475	6123	6164	6194	6223	6267	6475				
BIT9 = 001000 G	1309#	1371	1418										
BOE = 000400 G	1349#												
CKH = 000100 G	1376#	2411	2457	2529	2531	3460	3461	3474	3475	3545	3546	3740	3741
	3754	3755	3825	3826	4020	4021	4035	4036	6394	6395	6475		
CTSH = 000002 G	1381#	2411	2475	6475									
CSAU = 000052	1094#	2346											
CSAUTO= 000061	1094#	2275											
CSBRK = 000022	1094#												
CSBSEG= 000004	1094#	1927	1972	1989	2007	2025	2068	2410	2428	2456	2474	2510	2562
	2579	2605	2622	2657	2691	2707	2737	2753	2780	2809	2842	2902	3034
	3190	3255	3306	3350	3630	3909	4125	4224	4303	4470	4528	4612	4656
	4734	4824	4937	5074	5166	5256	5327	5370	5423	5494	5584	5658	5735
	5844	6013	6119	6326	6410	6474	6490						
CSBSUB= 000002	1094#	2404	2448	2556	2597	3186	3246	6321	6468				

CSGPHR= 000042	1094#	2228												
CSGPLO= 000030	1094#													
CSGPRI= 000040	1094#													
CSINIT= 000011	1094#	2258												
CSINLP= 000020	1094#													
CSMANI= 000050	1094#													
CSMEM = 000031	1094#													
CSMSG = 000023	1094#	1687	1710	1730	1743	1756	1769	1781	1793	1805				
CSOPEN= 000034	1094#													
CSPNTB= 000014	1094#	1667	1695	1718	1738	1751	1764	1777	1789	1801				
CSPNTF= 000017	1094#													
CSPNTS= 000016	1094#													
CSPNTX= 000015	1094#	1678	1706	1726	1811	1822	1828	1839	1845	1853	1859	1870		
CSQIO = 000377	1094#													
CSRDBU= 000007	1094#													
CSREFG= 000047	1094#	2199	2204	2209	2216	2222								
CSRESE= 000033	1094#	2213	6412	6492										
CSREVI= 000003	1094#	1155												
CSRFLA= 000021	1094#													
CSRPT = 000025	1094#	2170												
CSSEFG= 000046	1094#													
CSSPRI= 000041	1094#	2246												
CSSVEC= 000037	1094#	1933	2031	2074	6418	6498								
CSTPRI= 000013	1094#													
DFPTBL 002216 G	1248#													
DIAGMC= 000000	1094													
EF.CON= 000036 G	1325#	2221												
EF.NEW= 000035 G	1326#	2215												
EF.PWR= 000034 G	1327#	2208												
EF.RES= 000037 G	1324#	2203												
EF.STA= 000040 G	1323#	2198												
EMSGRO 003046 G	1581#	1664	1748											
EMSGR2 003076 G	1585#	1692	1761											
EMSGR4 003126 G	1589#	1715												
EMSGR6 003156 G	1593#	1735												
ERRBLK 002232 G	1450#													
ERRMSG 002230 G	1449#													
ERRNBR 002226 G	1448#													
ERRTYP 002224 G	1447#													
ESRH = 000040 G	1393#	3118	3119	3147	3445	3446	3488	3529	3725	3726	3768	3809	4005	
	4006	4049												
EVL = 000004 G	1343#													
E\$END = 002100	1094#													
E\$LOAD= 000035	1094#	1179												
FRMTR0 003256 G	1605#	1675	1703	1819	1836	1867								
FRMTR4 003363 G	1617#	1723	1850											
F\$AU = 000015	1094#	2334	2345											
F\$AUTO= 000020	1094#	2270	2274											
F\$BGN = 000040	1094#	1101	1276	1281	1662	1690	1713	1733	1746	1759	1772	1784	1796	
	1927	1972	1989	2007	2025	2068	2146	2151	2158	2164	2180	2196	2250	
	2270	2285	2293	2311	2317	2334	2340	2349	2355	2376	2383	2399	2403	
	2410	2428	2443	2447	2456	2474	2489	2493	2506	2510	2534	2550	2555	
	2562	2579	2592	2596	2605	2622	2635	2638	2651	2657	2672	2685	2691	
	2707	2720	2731	2737	2753	2766	2776	2780	2795	2805	2809	2823	2839	
	2842	2875	2896	2902	2969	3028	3034	3166	3182	3185	3190	3242	3245	
	3255	3306	3330	3333	3347	3350	3612	3627	3630	3892	3906	3909	4101	

LSENV1	002044	G	1150#		
LSERRT	002224	G	1446#		
LSETP	002102	G	1180#		
LSEXP1	002046	G	1152#		
LSEXP4	002064	G	1166#		
LSEXP5	002066	G	1168#		
LSHARD	021426	G	1129	6565	6566#
LSHIME	002120	G	1194#		
LSHPCP	002016	G	1128#		
LSHPTP	002022	G	1132#		
LSHW	002216	G	1133	1246	1247#
LSICP	002104	G	1182#		
LSINIT	005570	G	1183	2196#	
LSLADP	002026	G	1136#		
LSLAST	021530	G	1137	6633#	6646
LSLOAD	002100	G	1178#		
LSLUN	002074	G	1174#		
LSMREV	002050	G	1154#		
LSNAME	002000	G	1111#		
LSPRIO	002042	G	1148#		
LSPROT	005562	G	1189	2180#	
LSPRT	002112	G	1188#		
LSREPP	002062	G	1164#		
LSREV	002010	G	1120#		
LSRPT	005554	G	2158#		
LSSOFT	021504	G	6616	6617#	
LSSPC	002056	G	1160#		
LSSPCP	002020	G	1130#		
LSSPTP	002024	G	1134#		
LSSTA	002030	G	1138#		
LSSW	002224	G	1267	1268#	
LSTEST	002114	G	1190#		
LSTIML	002014	G	1126#		
LSUNIT	002012	G	1124#		
L10000	002222		1246	1255#	
L10001	002224		1267	1273#	
L10002	003710		1686#		
L10003	003776		1709#		
L10004	004050		1729#		
L10005	004076		1742#		
L10006	004124		1755#		
L10007	004152		1768#		
L10010	004174		1780#		
L10011	004216		1792#		
L10012	004240		1804#		
L10013	005560		2163	2169#	
L10015	005744		2251	2257#	
L10016	005746		2274#		
L10017	005776		2294	2300#	
L10020	006004		2316	2322#	
L10021	006012		2339	2345#	
L10022	006020		2383#		
L10023	006224		2493#		
L10024	006122		2443#		
L10025	006222		2489#		
L10026	006340		2534#		

L10027	006522	2638#												
L10030	006434	2592#												
L10031	006520	2635#												
L10032	006600	2672#												
L10033	006664	2720#												
L10034	006752	2766#												
L10035	007022	2795#												
L10036	007066	2823#												
L10037	007164	2875#												
L10040	007364	2969#												
L10041	007740	3166#												
L10042	010330	3333#												
L10043	010114	3242#												
L10044	010326	3330#												
L10045	011206	3612#												
L10046	012064	3892#												
L10047	012572	4101#												
L10050	013010	4198#												
L10051	013204	4283#												
L10052	013550	4450#												
L10053	014060	4581	4593#											
L10054	014314	4697	4703#											
L10055	014704	4903#												
L10056	015352	5125#												
L10057	016234	5458#												
L10060	016706	5695#												
L10061	017650	5973#												
L10062	020622	6302#												
L10063	021422	6546#												
L10064	021222	6463#												
L10065	021420	6543#												
L10066	021450	6565	6591#											
L10067	021504	6616	6624#											
L10070	021534	6640#												
L10072	021540	6639	6644#											
MPH =	000004	G 1380#	2411	2457	2847	3043	3359	3573	3639	3853	3919	4063	6475	
MPINH =	000001	G 1433#	2938	3104	3312	3417	3504	3697	3784	3977	6379			
MSAD0 =	000001	G 1427#												
MSAD1 =	000002	G 1426#												
MSAD10 =	002000	G 1417#												
MSAD11 =	004000	G 1416#												
MSAD12 =	010000	G 1415#												
MSAD13 =	020000	G 1414#	4195	4521	4577	5061	5122	5968	6298					
MSAD14 =	040000	G 1413#												
MSAD15 =	100000	G 1412#												
MSAD16 =	000001	G 1406#	2563	2606	2965	3162	3238	3326						
MSAD17 =	000002	G 1405#	2563	2623										
MSAD2 =	000004	G 1425#												
MSAD3 =	000010	G 1424#												
MSAD4 =	000020	G 1423#												
MSAD5 =	000040	G 1422#												
MSAD6 =	000100	G 1421#												
MSAD7 =	000200	G 1420#												
MSAD8 =	000400	G 1419#	2963	3160	3236	3324								
MSAD9 =	001000	G 1418#												
MSBRKH =	000200	G 1391#	3488	3529	3600	3768	3809	3880	4049	4089				

PARAMETER CODING	MACY11	30A(1052)	17-JUN-82	12:02	PAGE	139	I 11	CROSS REFERENCE TABLE -- USER SYMBOLS						SEQ 0138
CVC DAB.P11	17-JUN-82	11:26												
MSELO = 000004		1398#	2563	2606	2966	3163	3239	3327	4132	4220	4311	4359	4389	4421
		4475	4533	4608	4652	4742	4791	4945	4995	5174	5223	5502	5551	5743
		5792	6021	6070										
MSEL1 = 000010	G	1397#	2563	2623	2907	3072	3194	3259	3388	3668	3948	4220	4359	4421
		4608	4652	4791	4995	5223	5551	5792	6070	6350				
MSGMDA 002667	G	1555#	4570											
MSGMDC 002617	G	1548#	4412	4444										
MSGMDO 002513	G	1536#	4184	4348	4515	4780	4984	5212	5540	5781	6059			
MSGMD1 002555	G	1542#	4270	4381	4641	4685	4815	5019	5247	5575	5816	6094		
MSGMP 002356	G	1518#	2944	2957	3110	3139	3231	3319	3424	3510	3704	3790	3984	6386
MSGMPL 002463	G	1531#	3125	3154	3436	3452	3467	3481	3494	3522	3535	3552	3564	3579
		3592	3606	3716	3732	3747	3761	3774	3802	3815	3832	3844	3859	3872
		3886	3996	4012	4027	4042	4055	4069	4082	4095	6401			
MSGMPS 002417	G	1524#	3299											
MSGMSC 002775	G	1567#	5115											
MSGMSR 002732	G	1561#	4868	4882	5054	5300	5353	5394	5408	5448	5628	5684	5860	5878
		5895	5913	5928	5946	5961	6157	6186	6216	6259	6289			
MSGTMO 003417	G	1622#	1774											
MSGTM2 003473	G	1630#	1786											
MSGTM4 003547	G	1638#	1798											
MSG1 021450		6577	6597#											
MSG2 021464		6582	6599#											
MSRODT 014040		4508	4563	4583#										
MSR1DT 014304		4634	4678	4698#										
MUTBH = 000010	G	1436#	2938	3133	3312	3417	3504	3697	3784	3977	6379			
ONEFIL = 000001		1#	1092#	1095	1276#	1278	2147#	2149	2350	2351#	2353	6549	6550#	6552
OSAPTS = 000000		1094#	1138											
OSAU = 000000		1094#	1170											
OSBGNR = 000000		1094#	1164											
OSBGNS = 000000		1094#	1130											
OSDU = 000000		1094#	1172											
OSERRT = 000000		1094#	1180											
OSGNSW = 000000		1094#	1134											
OSPOIN = 000001		1094#	1108#	1196										
OSSETU = 000001		1094#	1108#	1124	6631									
PNT = 001000	G	1350#												
PRI = 002000	G	1351#												
PRI00 = 000000	G	1339#												
PRI01 = 000040	G	1338#												
PRI02 = 000100	G	1337#												
PRI03 = 000140	G	1336#												
PRI04 = 000200	G	1335#												
PRI05 = 000240	G	1334#												
PRI06 = 000300	G	1333#												
PRI07 = 000340	G	1149	1332#	1929	2027	2070	2245	6414	6494					
PRNTAL 004242	G	1740	1753	1766	1807#									
RDEH = 000004	G	1435#	2938	3104	3312	3417	3504	3784						
RDVH = 000020	G	1378#	1939	1992	2412	2430	2458	476	2511	2514	2848	3044	3360	3640
		3740	3920	4020	6394									
READR0 005362	G	2109#	3431	3517	3711	3797	3991	6452	6532					
READR2 005430	G	2120#	3120	3149	3447	3489	3530	3559	3601	3727	3769	3810	3839	3881
		4007	4050	4090										
READR4 005470	G	2130#												
READR6 005522	G	2139#	3294	4407	4439	4565	4680	5110	5348	5389	5443	5679	5890	5923
		5956	6211	6284										
REGO 002234	G	1456#	1941*	1942	2108*	2109	2232	2286*	6420	6500				

	3697*	3784*	3977*	4165*	4251*	4342*	4375*	4405*	4406	4437*	4438	4508*	4563*
	4564	4634*	4678*	4679	4773*	4808*	4861*	4876*	4977*	5012*	5047*	5107*	5108
	5205*	5240*	5293*	5345*	5346	5387*	5388	5401*	5402*	5440*	5441*	5442	5533*
	5568*	5621*	5676*	5677	5774*	5809*	5849*	5850	5852	5867*	5868	5887*	5888*
	5889	5902*	5903	5905	5920*	5921*	5922	5935*	5936	5938	5953*	5954*	5955
	6052*	6087*	6151*	6178*	6208*	6209	6251*	6281*	6282	6379*			
R6MASK 002306 G	1482#	1864	2141	2937*	3103*	3224*	3291*	3313*	3418*	3698*	3978*	4164*	4250*
	4341*	4509*	4635*	4774*	4862*	4978*	5048*	5109*	5206*	5294*	5347*	5534*	5622*
	5678*	5775*	5848*	6053*	6150*	6180*	6210*	6253*	6283*	6380*			
R6READ 002310 G	1483#	1865	2139*	2140	5851*	5853*	5854	5869*	5871*	5872	5904*	5906*	5907
	5937*	5939*	5940										
SFPTBL 002224 G	1269#												
SIG10H= 002000 G	1370#												
SIG11H= 004000 G	1369#												
SIG8H = 000400 G	1372#	2243											
SIG9H = 001000 G	1371#												
SVCGBL= 000000	1094#	1111	1112	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
	1157	1158	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	1206
	1207	1247	1248	1249	1268	1269	1270	1446	1447	1497	1498	1506	1507
	1662	1663	1690	1691	1713	1714	1733	1734	1746	1747	1759	1760	1772
	1773	1784	1785	1796	1797	2158	2159	2180	2181	2196	2197	2270	2271
SVCINS= 000000	2285	2286	2311	2312	2334	2335	6566	6567	6617	6618	6633#	6634	
	1094#	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	1196	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	1235
	1246	1247	1267	1268	1498	1500	1501	1507	1511	1512	1664	1665	1666
	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679
	1680	1687	1688	1692	1693	1694	1695	1696	1697	1698	1699	1700	1701
	1702	1703	1704	1705	1706	1707	1708	1710	1711	1715	1716	1717	1718
	1719	1720	1721	1722	1723	1724	1725	1726	1727	1728	1730	1731	1735
	1736	1737	1738	1739	1740	1743	1744	1748	1749	1750	1751	1752	1753
	1756	1757	1761	1762	1763	1764	1765	1766	1769	1770	1774	1775	1776
	1777	1778	1779	1781	1782	1786	1787	1788	1789	1790	1791	1793	1794
	1798	1799	1800	1801	1802	1803	1805	1806	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	1872	1927	1928	1929	1930	1931
	1932	1933	1934	1935	1948	1949	1950	1951	1952	1953	1954	1958	1959
	1960	1961	1962	1963	1964	1965	1967	1968	1972	1973	1978	1979	1980
	1981	1982	1984	1985	1989	1990	1996	1997	1998	1999	2000	2002	2003
	2007	2008	2012	2013	2014	2015	2016	2018	2019	2025	2026	2027	2028
	2029	2030	2031	2032	2033	2043	2044	2045	2046	2047	2048	2049	2053
	2054	2055	2056	2057	2058	2059	2060	2062	2063	2068	2069	2070	2071
	2072	2073	2074	2075	2076	2082	2083	2084	2085	2086	2087	2088	2092

2093	2094	2095	2096	2097	2098	2099	2101	2102	2162	2163	2164	2170
2171	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209
2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2221	2222	2223
2224	2225	2227	2228	2229	2230	2231	2232	2245	2246	2247	2250	2251
2252	2258	2259	2275	2276	2293	2294	2295	2301	2302	2315	2316	2317
2323	2324	2338	2339	2340	2346	2347	2384	2385	2404	2405	2410	2411
2416	2417	2418	2419	2420	2422	2423	2428	2429	2434	2435	2436	2437
2438	2440	2441	2444	2445	2448	2449	2456	2457	2462	2463	2464	2465
2466	2468	2469	2474	2475	2480	2481	2482	2483	2484	2486	2487	2490
2491	2494	2495	2510	2511	2518	2519	2520	2521	2522	2524	2525	2535
2536	2556	2557	2562	2563	2567	2568	2569	2570	2571	2573	2574	2579
2580	2584	2585	2586	2587	2588	2590	2591	2593	2594	2597	2598	2605
2606	2610	2611	2612	2613	2614	2616	2617	2622	2623	2627	2628	2629
2630	2631	2633	2634	2636	2637	2639	2640	2657	2658	2661	2662	2663
2664	2665	2667	2668	2673	2674	2691	2692	2696	2697	2698	2699	2700
2702	2703	2707	2708	2712	2713	2714	2715	2716	2718	2719	2721	2722
2737	2738	2742	2743	2744	2745	2746	2748	2749	2753	2754	2758	2759
2760	2761	2762	2764	2765	2767	2768	2780	2781	2784	2785	2786	2787
2788	2790	2791	2796	2797	2809	2810	2813	2814	2815	2816	2817	2819
2820	2824	2825	2842	2843	2852	2853	2854	2855	2856	2857	2858	2867
2868	2869	2870	2871	2873	2874	2876	2877	2902	2903	2913	2914	2915
2916	2917	2918	2919	2926	2927	2928	2929	2930	2931	2932	2942	2943
2944	2945	2946	2947	2948	2955	2956	2957	2958	2959	2961	2962	2970
2971	3034	3035	3048	3049	3050	3051	3052	3053	3054	3062	3063	3064
3065	3066	3067	3068	3078	3079	3080	3081	3082	3083	3084	3091	3092
3093	3094	3095	3096	3097	3108	3109	3110	3111	3112	3113	3114	3123
3124	3125	3126	3127	3128	3129	3137	3138	3139	3140	3141	3142	3143
3152	3153	3154	3155	3156	3158	3159	3167	3168	3186	3187	3190	3191
3200	3201	3202	3203	3204	3205	3206	3213	3214	3215	3216	3217	3218
3219	3229	3230	3231	3232	3233	3235	3236	3243	3244	3246	3247	3255
3256	3265	3266	3267	3268	3269	3270	3271	3278	3279	3280	3281	3282
3283	3284	3297	3298	3299	3300	3301	3303	3304	3306	3307	3317	3318
3319	3320	3321	3323	3324	3331	3332	3334	3335	3350	3351	3364	3365
3366	3367	3368	3369	3370	3378	3379	3380	3381	3382	3383	3384	3393
3394	3395	3396	3397	3398	3399	3406	3407	3408	3409	3410	3411	3412
3422	3423	3424	3425	3426	3427	3428	3434	3435	3436	3437	3438	3439
3440	3450	3451	3452	3453	3454	3455	3456	3465	3466	3467	3468	3469
3470	3471	3479	3480	3481	3482	3483	3484	3485	3492	3493	3494	3495
3496	3497	3498	3508	3509	3510	3511	3512	3513	3514	3520	3521	3522
3523	3524	3525	3526	3533	3534	3535	3536	3537	3538	3539	3550	3551
3552	3553	3554	3555	3556	3562	3563	3564	3565	3566	3567	3568	3577
3578	3579	3580	3581	3582	3583	3590	3591	3592	3593	3594	3595	3596
3604	3605	3606	3607	3608	3610	3611	3613	3614	3630	3631	3644	3645
3646	3647	3648	3649	3650	3658	3659	3660	3661	3662	3663	3664	3673
3674	3675	3676	3677	3678	3679	3686	3687	3688	3689	3690	3691	3692
3702	3703	3704	3705	3706	3707	3708	3714	3715	3716	3717	3718	3719
3720	3730	3731	3732	3733	3734	3735	3736	3745	3746	3747	3748	3749
3750	3751	3759	3760	3761	3762	3763	3764	3765	3772	3773	3774	3775
3776	3777	3778	3788	3789	3790	3791	3792	3793	3794	3800	3801	3802
3803	3804	3805	3806	3817	3814	3815	3816	3817	3818	3819	3830	3831
3832	3833	3834	3835	3836	3842	3843	3844	3845	3846	3847	3848	3857
3858	3859	3860	3861	3862	3863	3870	3871	3872	3873	3874	3875	3876
3884	3885	3886	3887	3888	3890	3891	3893	3894	3909	3910	3924	3925
3926	3927	3928	3929	3930	3938	3939	3940	3941	3942	3943	3944	3953
3954	3955	3956	3957	3958	3959	3966	3967	3968	3969	3970	3971	3972
3982	3983	3984	3985	3986	3987	3988	3994	3995	3996	3997	3998	3999

4000	4010	4011	4012	4013	4014	4015	4016	4025	4026	4027	4028	4029
4030	4031	4040	4041	4042	4043	4044	4045	4046	4053	4054	4055	4056
4057	4058	4059	4067	4068	4069	4070	4071	4072	4073	4080	4081	4082
4083	4084	4085	4086	4093	4094	4095	4096	4097	4099	4100	4102	4103
4125	4126	4137	4138	4139	4140	4141	4142	4143	4151	4152	4153	4154
4155	4156	4157	4182	4183	4184	4185	4186	4189	4190	4199	4200	4224
4225	4237	4238	4239	4240	4241	4242	4243	4268	4269	4270	4271	4272
4274	4275	4284	4285	4303	4304	4316	4317	4318	4319	4320	4321	4322
4330	4331	4332	4333	4334	4335	4336	4346	4347	4348	4349	4350	4351
4352	4363	4364	4365	4366	4367	4368	4369	4379	4380	4381	4382	4383
4384	4385	4393	4394	4395	4396	4397	4398	4399	4410	4411	4412	4413
4414	4415	4416	4425	4426	4427	4428	4429	4430	4431	4442	4443	4444
4445	4446	4448	4449	4451	4452	4470	4471	4480	4481	4482	4483	4484
4485	4486	4495	4496	4497	4498	4499	4500	4501	4513	4514	4515	4516
4517	4519	4520	4528	4529	4537	4538	4539	4540	4541	4542	4543	4550
4551	4552	4553	4554	4555	4556	4568	4569	4570	4571	4572	4574	4575
4580	4581	4582	4594	4595	4612	4613	4622	4623	4624	4625	4626	4627
4628	4639	4640	4641	4642	4643	4645	4646	4656	4657	4665	4666	4667
4668	4669	4670	4671	4683	4684	4685	4686	4687	4689	4690	4696	4697
4698	4704	4705	4734	4735	4747	4748	4749	4750	4751	4752	4753	4760
4761	4762	4763	4764	4765	4766	4778	4779	4780	4781	4782	4783	4784
4795	4796	4797	4798	4799	4800	4801	4813	4814	4815	4816	4817	4819
4820	4824	4825	4837	4838	4839	4840	4841	4842	4843	4849	4850	4851
4852	4853	4854	4855	4866	4867	4868	4869	4870	4871	4872	4880	4881
4882	4883	4884	4886	4887	4904	4905	4937	4938	4950	4951	4952	4953
4954	4955	4956	4964	4965	4966	4967	4968	4969	4970	4982	4983	4984
4985	4986	4987	4988	4999	5000	5001	5002	5003	5004	5005	5017	5018
5019	5020	5021	5022	5023	5035	5036	5037	5038	5039	5040	5041	5052
5053	5054	5055	5056	5058	5059	5074	5075	5085	5086	5087	5088	5089
5090	5091	5097	5098	5099	5100	5101	5102	5103	5113	5114	5115	5116
5117	5119	5120	5126	5127	5166	5167	5179	5180	5181	5182	5183	5184
5185	5192	5193	5194	5195	5196	5197	5198	5210	5211	5212	5213	5214
5215	5216	5227	5228	5229	5230	5231	5232	5233	5245	5246	5247	5248
5249	5251	5252	5256	5257	5269	5270	5271	5272	5273	5274	5275	5281
5282	5283	5284	5285	5286	5287	5298	5299	5300	5301	5302	5304	5305
5327	5328	5335	5336	5337	5338	5339	5340	5341	5351	5352	5353	5354
5355	5357	5358	5370	5371	5377	5378	5379	5380	5381	5382	5383	5392
5393	5394	5395	5396	5397	5398	5406	5407	5408	5409	5410	5412	5413
5423	5424	5430	5431	5432	5433	5434	5435	5436	5446	5447	5448	5449
5450	5453	5454	5459	5460	5494	5495	5507	5508	5509	5510	5511	5512
5513	5520	5521	5522	5523	5524	5525	5526	5538	5539	5540	5541	5542
5543	5544	5555	5556	5557	5558	5559	5560	5561	5573	5574	5575	5576
5577	5579	5580	5584	5585	5597	5598	5599	5600	5601	5602	5603	5609
5610	5611	5612	5613	5614	5615	5626	5627	5628	5629	5630	5632	5633
5658	5659	5666	5667	5668	5669	5670	5671	5672	5682	5683	5684	5685
5686	5688	5689	5696	5697	5735	5736	5748	5749	5750	5751	5752	5753
5754	5761	5762	5763	5764	5765	5766	5767	5779	5780	5781	5782	5783
5784	5785	5796	5797	5798	5799	5800	5801	5802	5814	5815	5816	5817
5818	5819	5820	5832	5833	5834	5835	5836	5838	5839	5844	5845	5858
5859	5860	5861	5862	5863	5864	5876	5877	5878	5879	5880	5881	5882
5893	5894	5895	5896	5897	5898	5899	5911	5912	5913	5914	5915	5916
5917	5926	5927	5928	5929	5930	5931	5932	5944	5945	5946	5947	5948
5949	5950	5959	5960	5961	5962	5963	5965	5966	5974	5975	6013	6014
6026	6027	6028	6029	6030	6031	6032	6039	6040	6041	6042	6043	6044
6045	6057	6058	6059	6060	6061	6062	6063	6074	6075	6076	6077	6078
6079	6080	6092	6093	6094	6095	6096	6097	6098	6110	6111	6112	6113

	6114	6116	6117	6119	6120	6127	6128	6129	6130	6131	6132	6133	6140
	6141	6142	6143	6144	6145	6146	6155	6156	6157	6158	6159	6160	6161
	6168	6169	6170	6171	6172	6173	6174	6184	6185	6186	6187	6188	6189
	6190	6198	6199	6200	6201	6202	6203	6204	6214	6215	6216	6217	6218
	6219	6220	6227	6228	6229	6230	6231	6232	6233	6241	6242	6243	6244
	6245	6246	6247	6257	6258	6259	6260	6261	6262	6263	6271	6272	6273
	6274	6275	6276	6277	6287	6288	6289	6290	6291	6295	6296	6303	6304
	6321	6322	6326	6327	6331	6332	6333	6334	6335	6336	6337	6341	6342
	6343	6344	6345	6346	6347	6355	6356	6357	6358	6359	6360	6361	6368
	6369	6370	6371	6372	6373	6374	6384	6385	6386	6387	6388	6389	6390
	6399	6400	6401	6402	6403	6405	6406	6410	6411	6412	6413	6414	6415
	6416	6417	6418	6419	6420	6424	6425	6426	6437	6438	6439	6443	6444
	6445	6446	6447	6448	6449	6455	6456	6457	6458	6459	6461	6462	6464
	6465	6468	6469	6474	6475	6479	6480	6481	6482	6483	6485	6486	6490
	6491	6492	6493	6494	6495	6496	6497	6498	6499	6500	6504	6505	6506
	6517	6518	6519	6523	6524	6525	6526	6527	6528	6529	6535	6536	6537
	6538	6539	6541	6542	6544	6545	6547	6548	6565	6566	6576	6577	6578
	6579	6580	6581	6582	6583	6584	6585	6586	6590	6591	6616	6617	6623
	6624	6630	6631	6632	6633	6638	6639	6640					
SVCSUB= 00000	1094#	2403	2404	2447	2448	2555	2556	2596	2597	3185	3186	3245	3246
	6320	6321	6467	6468									
SVCTAG= 00000	1094#	1255	1256	1273	1274	1686	1687	1709	1710	1729	1730	1742	1743
	1755	1756	1768	1769	1780	1781	1792	1793	1804	1805	1966	1967	1983
	1984	2001	2002	2017	2018	2061	2062	2100	2101	2169	2170	2257	2258
	2274	2275	2300	2301	2322	2323	2345	2346	2383	2384	2421	2422	2439
	2440	2443	2444	2467	2468	2485	2486	2489	2490	2493	2494	2523	2524
	2534	2535	2572	2573	2589	2590	2592	2593	2615	2616	2632	2633	2635
	2636	2638	2639	2666	2667	2672	2673	2701	2702	2717	2718	2720	2721
	2747	2748	2763	2764	2766	2767	2789	2790	2795	2796	2818	2819	2823
	2824	2872	2873	2875	2876	2960	2961	2969	2970	3157	3158	3166	3167
	3234	3235	3242	3243	3302	3303	3322	3323	3330	3331	3333	3334	3609
	3610	3612	3613	3889	3890	3892	3893	4098	4099	4101	4102	4188	4189
	4198	4199	4273	4274	4283	4284	4447	4448	4450	4451	4518	4519	4573
	4574	4593	4594	4644	4645	4688	4689	4703	4704	4818	4819	4885	4886
	4903	4904	5057	5058	5118	5119	5125	5126	5250	5251	5303	5304	5356
	5357	5411	5412	5452	5453	5458	5459	5578	5579	5631	5632	5687	5688
	5695	5696	5837	5838	5964	5965	5973	5974	6115	6116	6294	6295	6302
	6303	6404	6405	6460	6461	6463	6464	6484	6485	6540	6541	6543	6544
	6546	6547	6591	6592	6624	6625	6640	6641	6644	6645			
SVCTST= 00000	1094#	2376	2377	2399	2400	2506	2507	2550	2551	2651	2652	2685	2686
	2731	2732	2776	2777	2805	2806	2839	2840	2896	2897	3028	3029	3182
	3183	3347	3348	3627	3628	3906	3907	4119	4120	4218	4219	4299	4300
	4464	4465	4606	4607	4728	4729	4929	4930	5160	5161	5487	5488	5729
	5730	6007	6008	6316	6317								
SSI SYM= 010000	1094#	1256#	1274#	1687#	1710#	1730#	1743#	1756#	1769#	1781#	1793#	1805#	1927#
	1972#	1989#	2007#	2025#	2068#	2170#	2258#	2275#	2301#	2323#	2346#	2384#	2410#
	2428#	2444#	2456#	2474#	2490#	2494#	2510#	2535#	2562#	2579#	2593#	2605#	2622#
	2636#	2639#	2657#	2673#	2691#	2707#	2721#	2737#	2753#	2767#	2780#	2796#	2809#
	2824#	2842#	2876#	2902#	2970#	3034#	3167#	3190#	3243#	3255#	3306#	3331#	3334#
	3350#	3613#	3630#	3893#	3909#	4102#	4125#	4199#	4224#	4284#	4303#	4451#	4470#
	4528#	4594#	4612#	4656#	4704#	4734#	4824#	4904#	4937#	5074#	5126#	5166#	5256#
	5327#	5370#	5423#	5459#	5494#	5584#	5658#	5696#	5735#	5844#	5974#	6013#	6119#
	6303#	6326#	6410#	6464#	6474#	6490#	6544#	6547#	6592#	6625#			
T\$ARGC= 000006	1112#	1113#	1114#	1115#	1116#	1117#	1664#	1668	1670#	1679	1692#	1696	1698#
	1707	1715#	1719	1721#	1727	1735#	1739	1748#	1752	1761#	1765	1774#	1778
	1786#	1790	1798#	1802	1808#	1812	1814#	1823	1825#	1829	1831#	1840	1842#

TSCODE= 001032	1846	1848#	1854	1856#	1860	1862#	1871							
TSERRN= 000001	6576#	6581#												
	1094#	1949#	1959#	1979#	1997#	2013#	2044#	2054#	2083#	2093#	2417#	2435#	2463#	
	2481#	2519#	2568#	2585#	2611#	2628#	2662#	2697#	2713#	2743#	2759#	2785#	2814#	
	2853#	2868#	2914#	2927#	2943#	2956#	3049#	3063#	3079#	3092#	3109#	3124#	3138#	
	3153#	3201#	3214#	3230#	3266#	3279#	3298#	3318#	3365#	3379#	3394#	3407#	3423#	
	3435#	3451#	3466#	3480#	3493#	3509#	3521#	3534#	3551#	3563#	3578#	3591#	3605#	
	3645#	3659#	3674#	3687#	37.3#	3715#	3731#	3746#	3760#	3773#	3789#	3801#	3814#	
	3831#	3843#	3858#	3871#	3885#	3925#	3939#	3954#	3967#	3983	3995#	4011#	4026#	
	4041#	4054#	4068#	4081#	4094#	4138#	4152#	4183#	4238#	4267#	4317#	4331#	4347#	
	4364#	4380#	4394#	4411#	4426#	4443#	4461#	4496#	4514#	4538#	4551#	4569#	4623#	
	4640#	4666#	4684#	4748#	4'61#	4779#	4796#	4814#	4838#	4850#	4867#	4881#	4951#	
	4965#	4983#	5000#	5018#	5036#	5053#	5086#	5098#	5114#	5180#	5193#	5211#	5228#	
	5246#	5270#	5282#	5299#	5336#	5352#	5378#	5393#	5407#	5431#	5447#	5508#	5521#	
	5539#	5556#	5574#	5598#	5610#	5627#	5667#	5683#	5749#	5762#	5780#	5797#	5815#	
	5833#	5859#	5877#	5894#	5912#	5927#	5945#	5960#	6027#	6040#	6058#	6075#	6093#	
	6111#	6128#	6141#	6156#	6169#	6185#	6199#	6215#	6228#	6242#	6258#	6272#	6288#	
	6332#	6342#	6356#	6369#	6385#	6400#	6444#	6456#	6480#	6524#	6536#			
TSEXCP= 000000	6576#	6580	6581#	6586										
TSFLAG= 000040	2162#	2164	2250#	2293#	2315#	2317	2338#	2340	4580#	4696#				
TSFREE= 021540	6631	6646#												
TSGMAN= 000000	1094#													
TSHILI= 000017	6576#	6579	6581#	6585										
TSLAST= 000001	1094#	6631#	6637											
TSLOLI= 000000	6576#	6578	6581#	6584										
TSLSYM= 010000	1094#	1256	1274	1687	1710	1730	1743	1756	1769	1781	1793	1805	2170	
	2258	2275	2301	2323	2346	2384	2444	2490	2494	2535	2593	2636	2639	
	2673	2721	2767	2796	2824	2876	2970	3167	3243	3331	3334	3613	3893	
	4102	4199	4284	4451	4594	4704	4904	5126	5459	5696	5974	6303	6464	
	6544	6547	6592	6625										
TSLTNO= 000034	6634#													
TSNEST= 177777	1094#	1101#	1246#	1255#	1267#	1273#	1276#	1281#	1662#	1686#	1690#	1709#	1713#	
	1729#	1733#	1742#	1746#	1755#	1759#	1768#	1772#	1780#	1784#	1792#	1796#	1904#	
	1927#	1966#	1972#	1983#	1989#	2001#	2007#	2017#	2025#	2061#	2068#	2100#	2146#	
	2151#	2158#	2169#	2180#	2187#	2196#	2257#	2270#	2274#	2285#	2300#	2311#	2322#	
	2334#	2345#	2349#	2355#	2377#	2383#	2400#	2404#	2410#	2421#	2428#	2439#	2443#	
	2448#	2456#	2467#	2474#	2485#	2489#	2493#	2507#	2510#	2523#	2534#	2551#	2555#	
	2562#	2572#	2579#	2589#	2592#	2597#	2605#	2615#	2622#	2632#	2635#	2638#	2652#	
	2657#	2666#	2672#	2686#	2691#	2701#	2707#	2717#	2720#	2732#	2737#	2747#	2753#	
	2763#	2766#	2777#	2780#	2789#	2795#	2806#	2809#	2818#	2823#	2840#	2842#	2872#	
	2875#	2897#	2902#	2960#	2969#	3029#	3034#	3157#	3166#	3183#	3186#	3190#	3234#	
	3242#	3246#	3255#	3302#	3306#	3322#	3330#	3333#	3348#	3350#	3609#	3612#	3628#	
	3630#	3889#	3892#	3907#	3909#	4098#	4101#	4120#	4125#	4188#	4198#	4219#	4224#	
	4273#	4283#	4300#	4303#	4447#	4450#	4465#	4470#	4518#	4528#	4573#	4593#	4607#	
	4612#	4644#	4656#	4688#	4703#	4729#	4734#	4818#	4824#	4885#	4903#	4930#	4937#	
	5057#	5074#	5118#	5125#	5161#	5166#	5250#	5256#	5303#	5327#	5356#	5370#	5411#	
	5423#	5452#	5458#	5488#	5494#	5578#	5584#	5631#	5658#	5687#	5695#	5730#	5735#	
	5837#	5844#	5964#	5973#	6008#	6013#	6115#	6119#	6294#	6302#	6317#	6321#	6326#	
	6404#	6410#	6460#	6463#	6468#	6474#	6484#	6490#	6540#	6543#	6546#	6549#	6554#	
	6565#	6590#	6616#	6623#	6635#									
TSNSO = 000000	1101#	1276	1281#	2146	2151#	2349	2355#	6549	6554#	6635				
TSNS1 = 000005	1246#	1255	1267#	1273	1662#	1686	1690#	1709	1713#	1729	1733#	1742	1746#	
	1755	1759#	1768	1772#	1780	1784#	1792	1796#	1804	1927#	1966	1972#	1983	
	1989#	2001	2007#	2017	2025#	2061	2068#	2100	2158#	2169	2180#	2187	2196#	
	2257	2270#	2274	2285#	2300	2311#	2322	2334#	2345	2377#	2383	2400#	2493	
	2507#	2534	2551#	2638	2652#	2672	2686#	2720	2732#	2766	2777#	2795	2806#	

	2823	2840#	2875	2897#	2969	3029#	3166	3183#	3333	3348#	3612	3628#	3892
	3907#	4101	4120#	4198	4219#	4283	4300#	4450	4465#	4593	4607#	4703	4729#
	4903	4930#	5125	5161#	5458	5488#	5695	5730#	5973	6008#	6302	6317#	6546
T\$NS2 = 000002	6565#	6590	6616#	6623									
	2404#	2443	2448#	2489	2510#	2523	2556#	2592	2597#	2635	2657#	2666	2691#
	2701	2707#	2717	2737#	2747	2753#	2763	2780#	2789	2809#	2818	2842#	2872
	2902#	2960	3034#	3157	3186#	3242	3246#	3330	3350#	3609	3630#	3889	3909#
	4098	4125#	4188	4224#	4273	4303#	4447	4470#	4518	4528#	4573	4612#	4644
	4656#	4688	4734#	4818	4824#	4885	4937#	5057	5074#	5118	5166#	5250	5256#
	5303	5327#	5356	5370#	5411	5423#	5452	5494#	5578	5584#	5631	5658#	5687
T\$NS3 = 000003	5735#	5837	5844#	5964	6013#	6115	6119#	6294	6321#	6463	6468#	6543	
	2410#	2421	2428#	2439	2456#	2467	2474#	2485	2562#	2572	2579#	2589	2605#
	2615	2622#	2632	3190#	3234	3255#	3302	3306#	3322	6326#	6404	6410#	6460
	6474#	6484	6490#	6540									
T\$PCNT= 000000	6637#	6638#											
T\$PTAB= 010071	6638#	6641											
T\$PTHV= 000001	1125	6646#											
T\$PTNU= 000001	1094#	6641#	6646										
T\$SAVL= 177777	1094#												
T\$SEGL= 177777	1094#	1927#	1966#	1968	1972#	1983#	1985	1989#	2001#	2003	2007#	2017#	2019
	2025#	2061#	2063	2068#	2100#	2102	2410#	2421#	2423	2428#	2439#	2441	2456#
	2467#	2469	2474#	2485#	2487	2510#	2523#	2525	2562#	2572#	2574	2579#	2589#
	2591	2605#	2615#	2617	2622#	2632#	2634	2657#	2666#	2668	2691#	2701#	2703
	2707#	2717#	2719	2737#	2747#	2749	2753#	2763#	2765	2780#	2789#	2791	2809#
	2818#	2820	2842#	2872#	2874	2902#	2960#	2962	3034#	3157#	3159	3190#	3234#
	3236	3255#	3302#	3304	3306#	3322#	3324	3350#	3609#	3611	3630#	3889#	3891
	3909#	4098#	4100	4125#	4188#	4190	4224#	4273#	4275	4303#	4447#	4449	4470#
	4518#	4520	4528#	4573#	4575	4612#	4644#	4646	4656#	4688#	4690	4734#	4818#
	4820	4824#	4885#	4887	4937#	5057#	5055	5074#	5118#	5120	5166#	5250#	5252
	5256#	5303#	5305	5327#	5356#	5358	5370#	5411#	5413	5423#	5452#	5454	5494#
	5578#	5580	5584#	5631#	5633	5658#	5687#	5689	5735#	5837#	5839	5844#	5964#
	5966	6013#	6115#	6117	6119#	6294#	6296	6326#	6404#	6406	6410#	6460#	6462
	6474#	6484#	6486	6490#	6540#	6542							
T\$SEK0= 010001	1927#	1966	1972#	1983	1989#	2001	2007#	2017	2025#	2061	2068#	2100	2410#
	2421	2428#	2439	2456#	2467	2474#	2485	2510#	2523	2562#	2572	2579#	2589
	2605#	2615	2622#	2632	2657#	2666	2691#	2701	2707#	2717	2737#	2747	2753#
	2763	2780#	2789	2809#	2818	2842#	2872	2902#	2960	3034#	3157	3190#	3234
	3255#	3302	3306#	3322	3350#	3609	3630#	3889	3909#	4098	4125#	4188	4224#
	4273	4303#	4447	4470#	4518	4528#	4573	4612#	4644	4656#	4688	4734#	4818
	4824#	4885	4937#	5057	5074#	5118	5166#	5250	5256#	5303	5327#	5356	5370#
	5411	5423#	5452	5494#	5578	5584#	5631	5658#	5687	5735#	5837	5844#	5964
	6013#	6115	6119#	6294	6326#	6404	6410#	6460	6474#	6484	6490#	6540	
T\$SIZE= 000004	6632	6646#											
T\$SUBN= 000002	1094#	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#		
T\$TAGL= 177777	1094#												
T\$TAGN= 010073	1094#	1246#	1267#	1662#	1690#	1713#	1733#	1746#	1759#	1772#	1784#	1796#	2158#
	2180#	2196#	2270#	2285#	2311#	2334#	2377#	2400#	2404#	2448#	2507#	2551#	2556#
	2597#	2652#	2686#	2732#	2777#	2806#	2840#	2897#	3029#	3183#	3186#	3246#	3348#
	3628#	3907#	4120#	4219#	4300#	4465#	4607#	4729#	4930#	5161#	5488#	5730#	6008#
	6317#	6321#	6468#	6565#	6616#	6637#	6638#	6639#					
T\$TEMP= 000000	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#	1235#	1255#	1273#	1276#	1686#	1709#	1729#	1742#	1755#	1768#	1780#
	1792#	1804#	1966#	1983#	2001#	2017#	2061#	2100#	2146#	2162#	2163	2169#	2187#

TSTEST= 000034

TSTSTM= 177777

2250#	2251	2257#	2274#	2293#	2294	2300#	2315#	2316	2322#	2338#	2339	2345#
2349#	2383#	2421#	2439#	2443#	2467#	2485#	2489#	2493#	2523#	2534#	2572#	2589#
2592#	2615#	2632#	2635#	2638#	2666#	2672#	2701#	2717#	2720#	2747#	2763#	2766#
2789#	2795#	2818#	2823#	2872#	2875#	2960#	2969#	3157#	3166#	3234#	3242#	3302#
3322#	3330#	3333#	3609#	3612#	3889#	3892#	4098#	4101#	4188#	4198#	4273#	4283#
4447#	4450#	4518#	4573#	4580#	4581	4593#	4644#	4688#	4696#	4697	4703#	4818#
4885#	4903#	5057#	5118#	5125#	5250#	5303#	5356#	5411#	5452#	5458#	5578#	5631#
5687#	5695#	5837#	5964#	5973#	6115#	6294#	6302#	6404#	6460#	6463#	6484#	6540#
6543#	6546#	6549#	6576#	6581#	6590#	6623#	6635#					
1094#	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	6634	
1094#	1667	1678	1687	1695	1706	1710	1718	1726	1730	1738	1743	1751
1756	1764	1769	1777	1781	1789	1793	1801	1805	1811	1822	1828	1839
1845	1853	1859	1870	1927	1933	1948	1953	1958	1964	1967	1972	1978
1984	1989	1996	2002	2007	2012	2018	2025	2031	2043	2048	2053	2059
2062	2068	2074	2082	2087	2092	2098	2101	2170	2199	2204	2209	2213
2216	2222	2228	2246	2250	2258	2275	2293	2301	2323	2346	2384	2404
2410	2416	2422	2428	2434	2440	2444	2448	2456	2462	2468	2474	2480
2486	2490	2494	2510	2518	2524	2535	2556	2562	2567	2573	2579	2584
2590	2593	2597	2605	2610	2616	2622	2627	2633	2636	2639	2657	2661
2667	2673	2691	2696	2702	2707	2712	2718	2721	2737	2742	2748	2753
2758	2764	2767	2780	2784	2790	2796	2809	2813	2819	2824	2842	2852
2857	2867	2873	2876	2902	2913	2918	2926	2931	2942	2947	2955	2961
2970	3034	3048	3053	3062	3067	3078	3083	3091	3096	3108	3113	3123
3128	3137	3142	3152	3158	3167	3186	3190	3200	3205	3213	3218	3229
3235	3243	3246	3255	3265	3270	3278	3283	3297	3303	3306	3317	3323
3331	3334	3350	3364	3369	3378	3383	3393	3398	3406	3411	3422	3427
3434	3439	3450	3455	3465	3470	3479	3484	3492	3497	3508	3513	3520
3525	3533	3538	3550	3555	3562	3567	3577	3582	3590	3595	3604	3610
3613	3630	3644	3649	3658	3663	3673	3678	3686	3691	3702	3707	3714
3719	3730	3735	3745	3750	3759	3764	3772	3777	3788	3793	3800	3805
3813	3818	3830	3835	3842	3847	3857	3862	3870	3875	3884	3890	3893
3909	3924	3929	3938	3943	3953	3958	3966	3971	3982	3987	3994	3999
4010	4015	4025	4030	4040	4045	4053	4058	4067	4072	4080	4085	4093
4099	4102	4125	4137	4142	4151	4156	4182	4189	4199	4224	4237	4242
4268	4274	4284	4303	4316	4321	4330	4335	4346	4351	4363	4368	4379
4384	4393	4398	4410	4415	4425	4430	4442	4448	4451	4470	4480	4485
4495	4500	4513	4519	4528	4537	4542	4550	4555	4568	4574	4580	4594
4612	4622	4627	4639	4645	4656	4665	4670	4683	4689	4696	4704	4734
4747	4752	4760	4765	4778	4783	4795	4800	4813	4819	4824	4837	4842
4849	4854	4866	4871	4880	4886	4904	4937	4950	4955	4964	4969	4982
4987	4999	5004	5017	5022	5035	5040	5052	5058	5074	5085	5090	5097
5102	5113	5119	5126	5166	5179	5184	5192	5197	5210	5215	5227	5232
5245	5251	5256	5269	5274	5281	5286	5298	5304	5327	5335	5340	5351
5357	5370	5377	5382	5392	5397	5406	5412	5423	5430	5435	5446	5453
5459	5494	5507	5512	5520	5525	5538	5543	5555	5560	5573	5579	5584
5597	5602	5609	5614	5626	5632	5658	5666	5671	5682	5688	5696	5735
5748	5753	5761	5766	5779	5784	5796	5801	5814	5819	5832	5838	5844
5858	5863	5876	5881	5893	5898	5911	5916	5926	5931	5944	5949	5959
5965	5974	6013	6026	6031	6039	6044	6057	6062	6074	6079	6092	6097
6110	6116	6119	6127	6132	6140	6145	6155	6160	6168	6173	6184	6189
6198	6203	6214	6219	6227	6232	6241	6246	6257	6262	6271	6276	6287
6295	6303	6321	6326	6331	6336	6341	6346	6355	6360	6368	6373	6384
6389	6399	6405	6410	6412	6418	6425	6438	6443	6448	6455	6461	6464
6468	6474	6479	6485	6490	6492	6498	6505	6518	6523	6528	6535	6541

T24	015354 G	1230	5160#											
T25	016236 G	1231	5487#											
T26	016710 G	1232	5729#											
T27	017652 G	1233	6007#											
T28	020624 G	1234	6316#											
T28.1	020630	6320#												
T28.2	021224	6467#												
T3	006226 G	1209	2506#											
T4	006342 G	1210	2550#											
T4.1	006354	2555#												
T4.2	006436	2596#												
T5	006524 G	1211	2651#											
T6	006602 G	1212	2685#											
T7	006666 G	1213	2731#											
T8	006754 G	1214	2776#											
T9	007024 G	1215	2805#											
UAM	= 000200 G	1348#												
UNITNB	= 002246 G	1462#	2219*	2225*	2227									
WREH	= 000002 G	1434#	2938	3133	3312	3504	3697	3784						
WRENH	= 000100 G	1392#	3118	3147	3148	3445	3529	3725	3726	3768	3809	4005		
WRVH	= 000040 G	1377#	1939	1992	2412	247r	2458	2476	2511	2514	2848	3044	3360	3460
		3640	3920	4020	6394									
XSALWA	= 000000	1094#												
XSALS	= 000040	1094#												
XSOFFS	= 000400	1094#												
XSTRUE	= 000020	1094#												
\$PATCH	021504 G	6626#												
.	= 021540	1098#	1500#	1511#	1648#	2163	2251	2294	2316	2339	4581	4697	6627#	6639
		6646												

. 'BS. 021540 000

ERRORS DETECTED: 0

CVCDAB.OBJ, CVCDAB.SEQ/CRF:SYM/SOL/NL:TOC=SVC/MI, CVCDAB.P11
 RUN-TIME: 44 40 3 SECONDS
 RUN-TIME RATIO: 378/88=4.2
 CORE USED: 16K (31 PAGES)