DMP-11
DMR-11

M8207 STATIC DIAG.#2
CZDMQB0

AH-E229B-MC
FICHE 1 OF 1

```
                        .REM @


                                IDENTIFICATION
                                --------------

                        PRODUCT CODE:   AC-E228B-MC

                        PRODUCT NAME:   CZDMQB0 M8207 STATIC DIAG #2

                        PRODUCT DATE:   AUG. 1980

                        MAINTAINER:     DIAGNOSTICS MERRIMACK

                        AUTHOR:         ED BADGER


                        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) 1979,1980 BY DIGITAL EQUIPMENT CORPORATION

                THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

                        DIGITAL         PDP             UNIBUS          MASSBUS
                        DEC             DECUS           DECTAPE
```

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 PROGRAM ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST OUT THE M8200, M8204, CR M8207 MICROPROCESSOR. IT IS THE SECOND OF TWO DIAGNOSTICS FOR THESE OPTIONS.

THE PROGRAM WAS IMPLEMENTED USING THE DIAGNOSTIC SUPERVISOR.

THROUGH DIALOGUE WITH THE OPERATOR, THE PROGRAM WILL ALLOW MODIFICATION OF DEVICE PARAMETERS, SUCH AS UNIBUS ADDRESS, VECTOR ADDRESS, AND PROCESSOR TYPE.

1.2 HARDWARE INTRODUCTION

THE M820X MICROPROCESSOR USES AN EIGHT BIT DATA PATH WITH A SIXTEEN BIT INSTRUCTION MEMORY. THE INSTRUCTION MEMORY AND DATA MEMORY ARE TWO SEPARATE MEMORIES. THE MICROPROCESSOR IS DESIGNED FOR MOVING DATA AT HIGH RATES TO WORK AS A HIGH SPEED LINK BETWEEN PROCESSORS WHEN USED WITH A LINE UNIT. THE M8200 AND M8207 HAVE PROM INSTRUCTION MEMORIES. THE M8204 HAS WRITEABLE CONTROL STORE. THE MEMORY SIZES BETWEEN ALL THREE PROCESSORS VARY ALSO.

2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE M8207 LOGIC TESTS:

PDP-11/04,05,10.20,30,34,35,40,45,50,60, OR 70
16K MEMORY
CONSOLE TERMINAL

3.0 PRELIMINARY PROGRAM REQUIREMENTS

THE PROCESSOR AND MEMORY SHOULD BE THOROUGHLY TESTED PRIOR TO RUNNING THIS DIAGNOSTIC.

4.0 GENERAL PROGRAM CONSIDERATIONS

4.1 DIAGNOSTIC SUPERVISOR

THIS PROGRAM IS COMPATIBLE WITH THE STANDALONE DIAGNOSTIC SUPERVISOR, AND MUST BE LOADED TO BE CO-RESIDENT WITH THE

144 SUPERVISOR, OR BE PREVIOUSLY COMBINED WITH THE SUPERVISOR

AND LOADED AS A SINGLE FILE. IN EITHER CASE, THE COMBINED
PROGRAM WILL NOT EXCEED 16K OF MEMORY.

4.2 EXECUTION TIME

THE TOTAL TIME REQUIRED TO RUN THE M8207 STATIC TESTS IS
ABOUT 120 SECONDS PER PASS FOR EACH UNIT.

4.3 XXDP+

THIS PROGRAM MAY BE LOADED UNDER XXDP+, AND MAY BE RUN IN
DUMP MODE OR CHAIN MODE.

4.4 ACT/SLIDE

THIS PROGRAM MAY BE LOADED UNDER ACT OR SLIDE AND MAY BE RUN
IN DUMP MODE OR CHAIN MODE.

4.5 APT

THIS PROGRAM MAY BE LOADED BY THE APT SYSTEM (INCLUDING
APT-RD) AND RUN IN PROGRAM MODE OR SCRIPT MODE.

4.6 MEMORY MANAGEMENT

MEMORY MANAGEMENT IS NOT UTILIZED IN THIS PROGRAM. IF IT IS
INSTALLED, IT IS DISABLED BY THE PROGRAM.

4.7 MEMORY PARITY OPTION

IF PARITY MEMORY IS INSTALLED, MEMORY PARITY TRAPS ARE
DISABLED BY THE PROGRAM.

4.8 ERROR LOGGING

THE NUMBER OF ERRORS WHICH HAVE OCCURRED ON EACH DEVICE
UNDER TEST SINCE THE LAST START OR RESTART COMMAND IS KEPT
IN AN ERROR LOG.   THIS LOG MAY BE PRINTED BY USING THE
"PRINT" COMMAND (SEE SECTION 6.3.8).

5.0 PROGRAM LOAD MEDIA

THIS PROGRAM CAN BE LOADED FROM PAPER TAPE USING THE

```
   201                          ABSOLUTE  LOADER OR FROM ACT, SLIDE, OR APT SYSTEMS, OR FROM
```

ANY MEDIA SUPPORTED BY XXDP+. WHEN USING THE PAPER TAPE ABSOLUTE LOADER, THE PROGRAM SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC SUPERVISOR. WHEN USING XXDP+, THE DIAGNOSTIC SUPERVISOR SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC PROGRAM.

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

THIS PROGRAM MAY BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER. IT MAY ALSO BE LOADED FROM ANY XXDP+ LOAD MEDIA. WHEN LOADED UNDER XXDP+, THE DIAGNOSTIC SUPERVISOR WILL BE LOADED AUTOMATICALLY.

6.1.2 STARTING PROCEDURES

THE PROGRAM STARTS AT LOCATION 200. USE STANDARD DEC PROCEDURES TO START THE PROGRAM.

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

THE DIAGNOSTIC CAN BE EXECUTED STANDALONE UNDER XXDP+ WITHOUT READING THE REMAINDER OF THIS DOCUMENT, AS FOLLOWS:

A) LOAD AND START DIAGNOSTIC USING RUN COMMAND
B) RECEIVE DIAGNOSTIC SUPERVISOR PROMPT (DR>)
C) ENTER STA<CR>
D) ANSWER HARDWARE AND SOFTWARE QUESTIONS
E) GET END OF PASS MESSAGES OR ERROR MESSAGES
F) TO END EXECUTION, ENTER CONTROL/C

6.2 INITIAL DIALOGUE

AFTER THE PROGRAM AND THE SUPERVISOR ARE LOADED AND THE PROGRAM IS STARTED , THE FOLLOWING IDENTIFICATION IS TYPED:

```
DRS LOADED
DIAG.  RUN-TIME SERVICES
CZDMQ-B-0
M8207 DIAG.  #2 OF 2
UNIT IS M8200,M8204,OR M8207
DR>
```

THE OPERATOR THEN PROCEEDS BY TYPING ONE OR MORE OF THE COMMANDS DESCRIBED IN THE FOLLOWING SECTION 6.3.(FOR MORE DETAILED INFORMATION, REFER TO THE DIAGNOSTIC SUPERVISOR FUNCTIONAL SPECIFICATION).

6.3 PROGRAM OPTIONS

6.3.1 START COMMAND

```
*****************************************************************
STA(RT)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:
   <FLAG-LIST>/EOP:<INCR>
*****************************************************************
```

6.3.1.1 TESTS SWITCH (/TESTS:<TEST-LIST>)

<TEST-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS. ON THIS AND ALL SWITCHES, THE ANGLE BRACKETS <> ARE PUNCTUATION USED IN THE DEFINITION ONLY, AND ARE NOT TO BE TYPED BY THE OPERATOR. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.2 PASS SWITCH (/PASS:<PASS-CNT>)

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING EXECUTION. IN THIS CASE EXIT FROM THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY OCCURANCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.3 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED

```
314                                 LOE = LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP
```

```
                        CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK
                        OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAIN-
                        ING THE ERROR
                    IER INHIBIT ERROR REPORTING
                    IBE INHIBIT BASIC ERROR REPORTS
                    IXE INHIBIT EXTENDED ERROR REPORTS
                    PRI DIRECT ALL MESSAGES TO A LINE PRINTER
                    PNT PRINT NUMBER OF TEST BEING EXECUTED
                    BOE BELL ON ERROR
                    UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL
                        INTERVENTION TESTS
                    ISR INHIBIT STATISTICAL REPORTS
                    IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
                    LOT LOOP ON TEST

                THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO  0
                ARE CLEARED.  A FLAG NOT SPECIFIED IS CLEARED.  IF THE FLAGS
                SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED.  SEE  EXAMPLE  AT
                END OF 6.3.1.5.


                6.3.1.4 END OF PASS SWITCH (/EOP:<INCR>)

                <INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF
                PASSES) IT IS DESIRED  THAT  THE  END  OF  PASS  MESSAGE  BE
                PRINTED.   THE  DEFAULT  IS  AT  THE END OF EVERY PASS.  SEE
                EXAMPLE AT END OF 6.3.1.5.


                6.3.1.5 EFFECT OF START COMMAND

                THE EFFECT OF THE START COMMAND IS TO INITIATE THE  HARDWARE
                PARAMETER  DIALOGUE,  THE  SOFTWARE  PARAMETER DIALOGUE, AND
                THEN THE DIAGNOSTIC TESTS THEMSELVES.

                THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE  QUESTION
                "# UNITS?"  TO  WHICH  THE  OPERATOR REPLIES WITH A DECIMAL
                NUMBER N FROM 1 TO 16. THE TERM "UNIT" REFERS TO THE  DEVICE
                TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED.  FOLLOWING
                THIS ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES  WILL
                BE  BUILT.  EACH P-TABLE IS A CORE-RESIDENT TABLE CONTAINING
                ALL THE HARDWARE INFORMATION FOR  ONE  UNIT.   THE  OPERATOR
                MUST  SUPPLY  N  (NUMBER OF UNITS) VALUES FOR EACH QUESTION.
                HE MAY DO THIS BY GIVING ONE ANSWER  TO  EACH  QUESTION  (IN
                WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR
                BY GIVING N VALUES, SEPARATED BY COMMAS,  TO  EACH  QUESTION
                (SERIES  WILL  BE POSED ONCE).  EACH QUESTION IS FOLLOWED BY
                THE RESPONSE RADIX (D FOR  DECIMAL,  B  FOR  BINARY,  O  FOR
                OCTAL,  L  FOR  YES/NO) IN PARENTHESES AND THE DEFAULT VALUE
                AFTER THE PARENTHESES.
```

FOLLOWING THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS TO BUILD THE SOFTWARE TABLES, WHICH DEFINE THE MODE (QUICK VERIFY ETC.) THAT THE DIAGNOSTIC WILL EXECUTE IN.

WHEN THE QUESTION "# UNITS?" IS ANSWERED, MEMORY STORAGE IS ALLOCATED FOR THE P-TABLES, AND IF THERE IS NOT ENOUGH TO ACCOMMODATE THEM THE MESSAGE "TOO MANY UNITS" IS ISSUED. IN THIS CASE THE DIAGNOSTIC MUST BE EXECUTED MORE THAN ONCE TO TEST ALL UNITS.

EXAMPLE:

STA/TESTS:1:2-4:6:8-10/PASS:3/FLAGS:IER:HOE=1:UAM:LUE

THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, EACH PASS CONSISTING OF TESTS 1,2,3,4,6,8,9, AND 10 EXECUTED AGAINST ALL UNITS. THERE IS NO DIFFERENCE BETWEEN SAYING <FLAG> AND SAYING <FLAG=1>. THE NOTATION <FLAG=0> IS MEANINGFUL ONLY ON A COMMAND OTHER THAN START TO CLEAR A FLAG THAT WAS PREVIOUSLY SET. NOTE THAT ON ALL COMMANDS ONLY THE FIRST THREE LETTERS ARE SCANNED.

6.3.2 RESTART COMMAND

```
**********  **********
RES(TART)/TESTS:<TEST-LIST>/PAS .<PASS-CNT>/FLAGS:
   <FLAG-LIST>/UNITS:<UNIT-LIS >
**********  **********
```

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

<TEST-LIST>, <PASS-CNT>, AND <FLAG-LIST> ARE AS IN THE START COMMAND.

6.3.2.2 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (0,1 ETC.) OR RANGES OF DECIMAL NUMBERS (0-5, 8-10 ETC.) THAT SPECIFY THE UNITS TO BE TESTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS MAY RANGE FROM 0 THRU N-1 (N IS THE NUMBER OF UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE HARDWARE DIAGLOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

6.3.2.3 EFFECT OF RESTART COMMAND

THE RESTART COMMAND DIFFERS FROM THE START COMMAND IN THAT THE P-TABLES FROM THE PREVIOUS START COMMAND (THERE MUST HAVE BEEN ONE) ARE USED, INSTEAD OF NEW ONES BEING BUILT. THE UNITS SWITCH GIVES THE ABILITY TO SELECT A SUBSET OF THESE. THE SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED (OPERATOR WILL BE ASKED). THE COMMAND CAN BE USED AFTER COMMAND MODE HAS BEEN REENTERED IN ANY OF THE THREE NORMAL WAYS: A) THE REQUESTED NUMBER OF PASSES HAVE BEEN MADE B) AN ERROR WAS ENCOUNTERED WITH THE HALT ON ERROR FLAG SET C) A CONTROL/C WAS ENTERED BY THE OPERATOR.

6.3.3 CONTINUE COMMAND

****
CON(TINUE)/PASS:<PASS-CNT/FLAGS:<FLAG-LIST>
****

6.3.3.1 PASS SWITCH (/PASS:<PASS-CNT>)

<PASS-CNT> IS SAME AS IN START COMMAND, BUT THE DEFAULT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART. IF NONE REMAINS, THE DEFAULT IS NON-ENDING EXECUTION.

6.3.3.2 FLAG SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS SAME AS IN START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

6.3.3.3 EFFECT OF CONTINUE COMMAND

CONTINUE MUST FOLLOW A START OR RESTART, AND COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

6.3.4 PROCEED COMMAND

****
PRO(CEED)/FLAGS:<FLAG-LIST>
****

6.3.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS AS IN THE START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

6.3.4.2 EFFECT OF PROCEED COMMAND

PROCEED MUST FOLLOW A START, RESTART, OR CONTINUE. COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE PARAMETERS MAY BE ALTERED.

6.3.5 ADD COMMAND

****
ADD/UNITS:<UNIT-LIST>
****

6.3.5.1 UNITS SWITCH (/UNITS:<UNIT-LIST>

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.5.2 EFFECT OF ADD COMMAND

THE UNITS SPECIFIED ARE ADDED TO THE TEST SEQUENCE. EACH UNIT MUST HAVE A P-TABLE IN MEMORY DUE TO AN EARLIER HARDWARE DIALOGUE. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR CONTINUE. THE UNITS SWITCH MUST BE SPECIFIED. THE ADD COMMAND IS MEANINGFUL ONLY FOR UNITS THAT WERE PREVIOUSLY DROPPED.

6.3.6 DROP COMMAND

****
DRO(P)/UNITS:<UNIT-LIST>
****

6.3.6.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.6.2 EFFECT OF DROP COMMAND

THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING. THE UNITS WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR START COMMAND. THE UNITS SWITCH MUST BE ENTERED. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.

6.3.7 PRINT COMMAND

****
PRI(NT)
****

6.3.7.1 EFFECT OF PRINT COMMAND

THE TOTAL NUMBER OF ERRORS FOR EACH UNIT SINCE THE LAST START OR RESTART COMMAND ARE PRINTED. THE ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.

6.3.8 DISPLAY COMMAND

****
DIS(PLAY)/UNITS:<UNIT-LIST>
****

6.3.8.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.8.2 EFFECT OF DISPLAY COMMAND

THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS THAT WERE DROPPED BY THE OPERATOR "DROP" COMMAND ARE SO DESIGNATED.

6.3.9 FLAGS COMMAND

****
FLA(GS)
****

6.3.9.1 EFFECT OF FLAGS COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

6.3.10 ZFLAGS COMMAND

****
ZFL(AGS)
****

6.3.10.1 EFFECT OF ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

6.3.11 CONTROL CHARACTERS

A CONTROL C (C) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES A RETURN TO COMMAND MODE.

A CONTROL Z (Z) ENTERED DURING ONE OF THE THREE OPERATOR DIALOGUES- INITIAL DIALOGUE (SEE 6.2),HARDWARE DIALOGUE (SEE 6.3.1.5), OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.

A CONTROL O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES ALL TELETYPE OUTPUT TO BE SURPRESSED FOR THE REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER O IS TYPED, WHICH RESTORES NORMAL TELETYPE OUTPUT.

6.3.12 HARDWARE PARAMETERS

THE FOLLOWING QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

1. WHICH MICRO-CPU? (0= M8200, 4= M8204, 7= M8207) (0) 7?

2. MICRO-CPU CSR ADDRESS: (0) 160170?

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SEL0) RESIDE ON THE UNIBUS. THE ALLOWABLE RANGE IS 160000-177776 (OCTAL), AND THE DEFAULT IS 160170.

3. MICRO-PROCESSOR RUN SWITCH-TYPE 1 IF ON, IF OFF: (O) 0?

THE RUN SWITCH IS E28, SWITCH 7 ON THE M8207. MORE TESTS CAN BE PERFORMED IF THE RUN SWITCH IS OFF. YOU MAY GENERATE AN ERROR IF YOU ANSWER THIS QUESTION WRONG.

6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY PART 2 OF THE STATIC LOGIC TESTS.

6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.

AS SOON AS THE QUESTION "# UNITS?" IS ANSWERED (WITH THE NUMBER N, SAY) SPACE IN CORE IS ALLOCATED FOR N P-TABLES. ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A ONE-TO ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN ALL OF THE P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN LESS THAN N EXPLICIT VALUES IN RESPONSE TO A PARTICULAR QUESTION, THESE VALUES ARE PLACED IN THE P-TABLES (ONE VALUE GOING INTO THE PROPER SLOT OF EACH P-TABLE BEGINNING WITH THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS EXHAUSTED. THE LAST VALUE IN THE STRING BECOMES THE NEW DEFAULT AND IS USED TO FILL THAT SLOT IN THE REMAINING P-TABLES.

ON SUBSEQUENT TRIPS THRU THE QUESTIONS, THE SAME PROCESS IS CARRIED OUT, EXCEPT THAT THE EARLIEST P-TABLE NOT TO HAVE RECEIVED AN EXPLICIT VALUE IN ANY OF ITS SLOTS NOW ASSUMES THE ROLE THAT TABLE NUMBER ONE PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR.

IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6-10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 16 UNITS, AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 16 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER (0,1,2,...,15) EXCEPT FOR UNIT 12, WHICH SHOULD RECEIVE THE VALUE 11. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE THE NUMBER 76 FOR THE FIRST 7 UNITS AND THE NUMBER 77 FOR THE LAST 9 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

```
# UNITS (D) ? 16

UNIT 1
<QUESTION 1> ? 75
<QUESTION 2> ? 0-6
<QUESTION 3> ? 76

UNIT 21
<QUESTION 1> ?
<QUESTION 2> ? 7-11,,13-15
<QUESTION 3> ? 77
```

THE FIRST TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75 IN ALL 16 TABLES. SLOT TWO RECEIVES THE VALUES 0,1,2,...,6 IN TABLES 0 THRU 6 AND A CONSTANT 6 IN TABLES 7 THRU 15. SLOT THREE RECEIVES A CONSTANT 76 IN ALL 16 TABLES.

THE SECOND TIME THRU THE SERIES, TABLES 16 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE THE OPERATOR IN THE FORM "UNIT XX" AT THE BEGINNING OF EACH SERIES). QUESTION 1 IS RESPONDED TO BY A <CR>, SO SLOT ONE STAYS AT CONSTANT 75 IN TABLES 7 THRU 15, SINCE NO NEW EXLICIT VALUES ARE TYPED IN. SLOT TWO GETS THE VALUES 7,8,9,10,11 IN TABLES 7 THRU 11, AND GETS A 11 IN SLOT 12, AND GETS THE VALUES 13,14,15 IN TABLES 13 THRU 15. SLOT THREE GETS THE VALUE 77 IN TABLES 7 THRU 15.

THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT 16 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION (NAMELY QUESTION 2).

```
744
745
746
747
748
749                             7.0  TEST DESCRIPTIONS
750
751
752                             ********************* TEST 1 *********************
753                             *VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
754                             *DOES NOT CAUSE A TIME OUT TRAP
755                             ***************************************************
756
757
758                             ********************* TEST 2 *********************
759                             *TEST OF BR RIGHT SHIFT
760                             *VERIFY THAT A DEST OF BR RSH (011) OF A MICRO-INSTRUCTION
761                             *SHIFTS THE RESULTING BR  DATA RIGHT ONCE.
762                             ***************************************************
763
764
765                             ********************* TEST 3 *********************
766                             *IOP CRAM WRITE/READ TEST
767                             *FLOAT A 1 THROUGH EACH CRAM LOCATION
768                             ***************************************************
769
770
771                             ********************* TEST 4 *********************
772                             *IOP CRAM WRITE/READ TEST
773                             *FLOAT A 0 THROUGH EACH CRAM LOCATION
774                             ***************************************************
775
776
777                             ********************* TEST 5 *********************
778                             *IOP CRAM DUAL ADDRESSING TEST
779                             *WRITE EACH ADDRESS INTO ITSELF, READ EACH
780                             *ADDRESS TO VERIFY CORRECT ADDRESSING
781                             ***************************************************
782
783
784                             ********************* TEST 6 *********************
785                             *IOP MAIN MEMORY TEST
786                             *FLOAT A 1 THROUGH ALL MAIN MEMORY LOCATIONS
787                             ***************************************************
788
789
790                             ********************* TEST 7 *********************
791                             *IOP MAIN MEMORY TEST
792                             *FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATIONS
793                             ***************************************************
794
795
796                             ********************* TEST 8 *********************
797                             *IOP MAIN MEMORY DUAL ADDRESSING TEST
798                             *LOAD EACH MEMORY LOCATION WITH ITS OWN ADDRESS
799                             *READ BACK EACH LOCATION TO VERIFY CORRECT ADDRESSING
```

```
800                     ********************************************************
```

```
                        ********** TEST 9 **********
                        *IOP MAR TEST
                        *PERFORM DUAL ADDRESSING TEST
                        *USING MAR AUTO-INC FEATURE
                        ****************************************


                        ********** TEST 10 **********
                        *IOP (CRAM) ODT BITS TEST
                        *LOAD MAR WITH A 0 INC MAR UNTIL IT OVERFLOWS
                        *VERIFY THAT IBUS* 10 BITS IS SET ONLY WHEN MAR BIT 8 IS A ONE
                        *AND THAT IBUS* 10 BIT6 IS SET ON MAR OVERFLOW
                        ****************************************


                        ********** TEST 11 **********
                        *CRAM TEST OF JUMP(I) NEVER MICRO-PROCESSOR INSTRUCTION.
                        *PERFORM THE JUMP INSTRUCTION
                        *VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
                        *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
                        *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
                        *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
                        *THE CRAM PC IS CORRECT.  IF THE CRAM PC IN NOT RIGHT,
                        *THEN PORT4 CONTAINS A 37
                        ****************************************


                        ********** TEST 12 **********
                        *CRAM TEST OF JUMP(I) ALWAYS MICRO-PROCESSOR INSTRUCTION.
                        *PERFORM THE JUMP INSTRUCTION
                        *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
                        *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
                        *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
                        *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
                        *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
                        *THEN PORT4 WILL CONTAIN A 37
                        ****************************************


                        ********** TEST 13 **********
                        *CRAM TEST OF JUMP(I) ON C BIT SET MICRO-PROCESSOR INSTRUCTION.
                        *SET THE C BIT, PERFORM THE JUMP INSTRUCTION.
                        *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
                        *IN THE LOCATION IT IS AT.  THIS INSTRUCITON LOADS THE
                        *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
                        *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
                        *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
                        *THEN PORT4 WILL CONTAIN A 37.
                        ****************************************
```

```
                                                                I  2
   857
```

```
858
859
860
861
862
863
864                         ******************** TEST 14 ************************
865                         *CRAM TEST OF JUMP(I) ON Z BIT SET MICRO-PROCESSOR INSTRUCTION.
866                         *SET THE Z BIT, PERFORM THE JUMP INSTRUCTION.
867                         *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
868                         *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
869                         *9R WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
870                         *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
871                         *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
872                         *THEN PORT4 WILL CONTAIN A 37
873                         ******************************************************
874
875
876                         ******************** TEST 15 ************************
877                         *CRAM TEST OF JUMP(I) ON BRO SET MICRO-PROCESSOR INSTRUCTION.
878                         *SET THE BRO BIT, PERFORM THE JUMP INSTRUCTION.
879                         *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
880                         *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
881                         *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
882                         *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
883                         *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
884                         *THEN THE PORT4 WILL CONTAIN A 37
885                         ******************************************************
886
887
888                         ******************** TEST 16 ************************
889                         *CRAM TEST OF JUMP(I) ON BR1 SET MICRO-PROCESSOR INSTRUCTION.
890                         *SET THE BR1 BIT, PERFORM THE JUMP INSTRUCTION.
891                         *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
892                         *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
893                         *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
894                         *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
895                         *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
896                         *THEN PORT4 WILL CONTAIN A 37
897                         ******************************************************
898
899
900                         ******************** TEST 17 ************************
901                         *CRAM TEST OF JUMP(I) ON BR4 SET MICRO-PROCESSOR INSTRUCTION.
902                         *SET THE BR4 BIT, PERFORM THE JUMP INSTRUCTION.
903                         *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
904                         *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
905                         *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
906                         *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
907                         *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
908                         *THEN PORT4 WILL CONTAIN A 37
909                         ******************************************************
910
911
912                         ******************** TEST 18 ************************
913                         *CRAM TEST OF JUMP(I) ON BR7 SET MICRO-PROCESSOR INSTRUCTION.
```

```
   914                          *SET THE BR7 BIT, PERFORM THE JUMP INSTRUCTION
```

```
                        *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
                        *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
                        *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
                        *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
                        *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
                        *THEN PORT4 WILL CONTAIN A 37
                        ************************************************************


                        ********************* TEST 19 *************************
                        *CRAM TEST OF JUMP(I) ON C BIT CLEAR MICRO-PROCESSOR INSTRUCTION.
                        *SET THE C BIT, PERFORM THE JUMP INSTRUCTION.
                        *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
                        *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
                        *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
                        *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
                        *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
                        *THEN PORT4 WILL CONTAIN A 37
                        ************************************************************


                        ********************* TEST 20 *************************
                        *CRAM TEST OF JUMP(I) ON Z BIT CLEAR MICRO-PROCESSOR INSTRUCTION.
                        *CLEAR THE Z BIT, PERFORM THE JUMP INSTRUCTION.
                        *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
                        *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
                        *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
                        *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
                        *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
                        *THEN PORT4 WILL CONTAIN A 37
                        ************************************************************


                        ********************* TEST 21 *************************
                        *CRAM TEST OF JUMP(I) ON BR0 CLEAR MICRO-PROCESSOR INSTRUCTION.
                        *CLEAR THE BR0 BIT, PERFORM THE JUMP INSTRUCTION.
                        *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
                        *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
                        *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  A THIS POINT
                        *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
                        *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
                        *THEN PORT4 WILL CONTAIN A 37
                        ************************************************************


                        ********************* TEST 22 *************************
                        *CRAM TEST OF JUMP(I) ON BR1 CLEAR MICRO-PROCESSOR INSTRUCTION.
                        *CLEAR THE BR1 BIT, PERFORM THE JUMP INSTRUCTION.
                        *VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
                        *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
                        *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
```

```
                                                                    M 2
   971                          *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
```

```
                        *THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
                        *THEN PORT4 WILL CONTAIN A 37
                        ************************************************************


                        ******************** TEST 23 *************************
                        *CRAM TEST OF JUMP(I) ON BR4 CLEAR MICRO-PROCESSOR INSTRUCTION.
                        *CLEAR THE BR4 BIT, PERFORM THE JUMP INSTRUCTION.
                        *VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
                        *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
                        *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
                        *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
                        *THE CRAM PC IS CORRECT, IF THE CRAM PC IS NOT RIGHT
                        *THEN PORT4 CONTAINS A 37
                        ************************************************************


                        ******************** TEST 24 *************************
                        *CRAM TEST OF JUMP(I) ON BR7 CLEAR MICRO-PROCESSOR INSTRUCTION.
                        *CLEAR THE BR7 BIT, PERFORM THE JUMP INSTRUCTION.
                        *VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
                        *IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
                        *BR WITH THE LOWEST 8 BITS OF THE CRAM PC.  AT THIS POINT
                        *THE BR DATA IS MOVED TO PORT4.  IF THIS DATA IS CORRECT
                        *THE CRAM PC IS CORRECT, IF THE CRAM PC IS NOT RIGHT
                        *THEN PORT4 CONTAINS A 37
                        ************************************************************


                        ******************** TEST 25 *************************
                        *
                        *MAIN MEMORY PAGE DUAL ADDRESS TEST.
                        *IN THIS TEST WE WILL VERIFY THAT PAGES DO
                        *NOT DUAL ADDRESS.  THIS TEST IS DIFFERENT FROM THE
                        *PREVIOUS DUAL ADDRESS TESTS IN THAT THE OTHER
                        *TEST REALLY DIDN'T CHECK PAGE DUAL ADDRESSING
                        ************************************************************


                        ******************** TEST 26 *************************
                        *
                        *JUMP FIELD,PAGE TEST
                        *
                        *IN THIS TEST WE WILL MAKE SURE A JUMP FIELD INSTRUCTION
                        *WORKS.  TO DO THIS, WE'LL PUT THE DESIRED PAGE, FIELD
                        *INFORMATION IN IBUS*<13> THEN ISSUE A JUMP FIELD
                        *THEN WE'LL READ PC REG. AND VERIFY.
                        *
                        ************************************************************
```

```
1028                    ****************** TEST 27 **************************
```

```
1029
1030
1031
1032
1033
1034                            *
1035                            *JUMP TEST, JUMP ALWAYS, JUMP CHANGE FIELD
1036                            *
1037                            *IN THIS TEST, WE WILL CHECK THE ABILITY OF THE
1038                            *MICRO-PROCESSOR TO JUMP (BRANCH AND ALWAYS INSTRUCTION)
1039                            *TO LOCATIONS, FIELDS FROM OTHER LOCATIONS FIELDS.
1040                            *WE ALREADY KNOW THAT THE BRANCH INSTR WORKS FROM
1041                            *OTHER TEST.              PROCEDURE:
1042                            *       1.  START ADDR 0, FIELD 0
1043                            *       2.  **CALCULATE NEW ADDRS, FIELD VIA INC,
1044                            *       3.  CAUSE JUMP (BRANCH) TO NEW ADDRESS
1045                            *       4.  READ PC FROM IBUS*12 AND IBUS*13
1046                            *       5.  REPEAT STEP 2-4 256.TIMES
1047                            *
1048                            *       TO CALCULATE NEW ADDRESS:
1049                            *       1.  INC LOW BYTE OF ADDRESS FOR PC ADDRESS 0-7
1050                            *       2.  INC LOW BYTE OF N ADDRESS FOR PC ADDRESS 8-11
1051                            *           BITS REPRESENTED AS BITS 0-3.  WHEN 0-3 OVERFLOWS,
1052                            *           RESTARTS AT ZERO.
1053                            *                 NET RESULT IS JUMPS FROM:
1054                            *           FIELD,PAGE                  LOC
1055                            *                    0                  0
1056                            *                    1                  1
1057                            *                    2                  2
1058                            *                    3                  3
1059                            *                   10                  7
1060                            *                   11                 11
1061                            *                   :TO                :
1062                            *                   17                377
1063                            ******************************************************
1064
1065
1066                            ******************** TEST 28 ************************
1067                            *
1068                            *JUMP TEST, JUMP ALWAYS, JUMP CHANGE FIELD
1069                            *
1070                            *IN THIS TEST, WE WILL CHECK THE ABILITY OF THE
1071                            *MICRO-PROCESSOR TO JUMP (BRANCH AND ALWAYS INSTRUCTION)
1072                            *TO LOCATIONS, FIELDS FROM OTHER LOCATIONS FIELDS.
1073                            *WE ALREADY KNOW THAT THE BRANCH INSTR WORKS FROM
1074                            *OTHER TESTS.              PROCEDURE:
1075                            *       1.  START ADDR 0, FIELD 0
1076                            *       2.  **CALCULATE NEW ADDRS, FIELD VIA DEC,
1077                            *       3.  CAUSE JUMP (BRANCH) TO NEW ADDRESS
1078                            *       4.  READ PC FROM IBUS*12 AND IBUS*13
1079                            *       5.  REPEAT STEP 2-4 256.TIMES
1080                            *
1081                            *       TO CALCULATE NEW ADDRESS:
1082                            *       1.  DEC LOW BYTE OF ADDRESS FOR PC ADDRESS 0-7
1083                            *       2.  DEC LOW BYTE OF N ADDRESS FOR PC ADDRESS 8-11
1084                            *           BITS REPRESENTED AS BITS 0-3.  WHEN 0-3 OVERFLOWS,
```

```
1085                                    *           RESTARTS AT ZERO
```

```
1086
1087
1088
1089
1090
1091                            *              NET RESULT IS JUMPS FROM:
1092                            *            FIELD,PAGE              LOC:
1093                            *                 0                  0
1094                            *                17                  377
1095                            *                16                  376
1096                            *                15                  375
1097                            *                :TO                 :
1098                            *                00                  000
1099                            ************************************************************
1100
1101
1102                            ******************** TEST 29 *************************
1103                            *
1104                            *IN THIS TEST WE'LL VERIFY THAT THE Z BIT CAN BE READ FROM
1105                            *IBUS*<13>. WE ALREADY KNOW THAT THE Z BIT WORKS PROPERLY,
1106                            *ALL WE WANT TO KNOW HERE IS THAT IT CAN BE READ.
1107                            ************************************************************
1108
1109
1110
1111                            ******************** TEST 30 *************************
1112                            *
1113                            *IN THIS TEST WE'LL VERIFY THAT THE C BIT CAN BE READ FROM
1114                            *IBUS*<13>. WE ALREADY KNOW THAT THE C BIT WORKS PROPERLY
1115                            *ALL WE WANT TO KNOW HERE IS THAT IT BE READ.
1116                            *
1117                            ************************************************************
1118
1119
1120                            ******************** TEST 31 *************************
1121                            *TEST OF PROGRAM CLOCK BIT
1122                            *DO A MASTER CLEAR, VERIFY THAT PROGRAM CLOCK IS SET
1123                            *WRITE PROGRAM CLOCK BIT TO A ONE, VERIFY THAT IT CLEARS,
1124                            *AND THEN SETS SOME TIME LATER
1125                            ************************************************************
1126
1127
1128                            ******************** TEST 32 *************************
1129                            *FORCE POWER FAIL TEST
1130                            *SET FORCE POWER FAIL BIT VERIFY THAT PROCESSOR TRAPS TO 24
1131                            *GOING DOWN AND COMING UP.  VERIFY ALSO THAT BUS INIT WAS
1132                            *BLOCKED FROM GETTING TO THE M8200,4,7 DURING THE POWER FAIL
1133                            ************************************************************
1134
1135
1136                            ******************** TEST 33 *************************
1137                            *MICRO-PROCESSOR NOISE TEST
1138                            *WRITE ALL ZERO'S THEN ALL ONE'S THEN A DATA PATTERN
1139                            *TO THE IBUS* AND IBUS REGISTERS AND TO THE SP AND MAIN MEM
1140                            *THEN GO BACK AND READ THE DATA PATTERNS TO VERIFY THAT
1141                            *READING AND WRITING OF OTHER LOCATIONS AND REGISTERS
```

```
                                                                F 3
  1142                          *DID NOT CHANGE THE DATA.
```

```
1143
1144
1145
1146
1147
1148                    ************************************************************
1149
1150
1151                    ******************** TEST 34 ***************************
1152                    *THIS TEST IS DESIGNED TO MAKE SURE THAT A NODST INSTRUCTION
1153                    *DOES NOT WRITE INTO PORT B OF THE MULTIPORT RAM.
1154                    *TO DO THIS, WE'LL PUT A 125 INTO INDAT2, THEN WE'LL PUT A
1155                    *125 INTO BOTH SP1 AND BR. LAST WE'LL DO A NODST BR, SUBOC, SP1
1156                    *IF THERE IS A WRITE INTO PORTB, INADT2 WILL CONTAIN A 377
1157                    ************************************************************
1158
1159
1160                    ******************** TEST 35 ***************************
1161                    *
1162                    *EXTENDED CRAM TEST FOR M8206.  IN THIS TEST WE WILL LOAD DATA
1163                    *THROUGHOUT THE CRAM (TEST DATA IS JUST 4K OF DIAG. CODE) AND
1164                    *THEN READ IT BACK AND VERIFY THAT IT IS CORRECT
1165                    ************************************************************
1166
1167
1168                    ******************** TEST 36 ***************************
1169                    *
1170                    *THIS TEST LOADS MICRO-CODE INTO A M8206 MCPU THEN EXECUTES IT.
1171                    *THE MICRO-CODE IS DESIGNED TO WRITE ALL ONES INTO THE SEL REGS.
1172                    *
1173                    ************************************************************
1174
1175
1176                    ******************** TEST 37 ***************************
1177                    *
1178                    *NEGATIVE ADDRESS TEST.
1179                    *       IN THIS TEST, WE'LL MAKE SURE THAT THE M8207
1180                    *       DOES NOT RESPOND TO AN ADDRESS THAT ISN'T ASSIGNED
1181                    *        TO IT
1182                    *
1183                    ************************************************************
1184
1185
1186                    ******************** TEST 38 ***************************
1187                    *
1188                    *BYTE ADDRESSING TEST
1189                    *       HERE, WE'RE GOING TO MAKE SURE THAT WE CAN
1190                    *        WRITE INTO ONLY A HIGH OR LOW BYTE OF THE MCPU.
1191                    *
1192                    ************************************************************
1193
1194
1195                    ******************** TEST 39 ***************************
1196                    *
1197                    *IN THIS TEST WE'RE GOING TO MAKE SURE THAT THE PC
1198                    *REG COUNTS UP PROPERLY.  THE PC REG SHOULD INCREMENT
```

```
1199                                    *ONCE AFTER EACH INSTRUCTION.
```

```
1200
1201
1202
1203
1204
1205                                    *
1206                                    ************************************************************
1207
1208
1209                                    ******************** TEST 40 ****************************
1210                                    *
1211                                    *!N THIS TEST WE'LL MAKE SURE THAT "BRANCH FIELD H" DOESN'T
1212                                    *GET STUCK HIGH.
1213                                    *FIRST WE'LL CLEAR THE PC HIGH REG. THEN WE'LL DO A BRANCH INSTR
1214                                    *WITH BAB BITS 11+12 SET.  IF PCR  BITS 8+9 SET THEN WE'LL KNOW
1215                                    *WE WERE SUCCESSFUL IF PCR BITS 8+9 FAIL TO SET, WE,LL KNOW
1216                                    *THAT THE MAX SELECTED THE WRONG INPUT TO BE CLOCKED INTO THE PCR.
1217                                    ************************************************************
1218
1219
1220                                    ******************** TEST 41 ****************************
1221                                    *
1222                                    *IN THIS TEST WE'RE GOING TO MAKE SURE THAT ONLY SPO
1223                                    *IS SELECTED FOR SOURCE WHEN THE DESTINATION
1224                                    *IS THE OUTBUS
1225                                    *FIRST WE'LL WRITE EACH SP ADDRS INTO ITSELF THEN WE'LL
1226                                    *MOV SP TO OBUS4.  THAT SHOULD SELECT
1227                                    *SP ADDRESS 0.  IF ANY OTHER DATA SHOWS UP, WE'LL
1228                                    *BLAME IT ON THE SELECTION OF A DIFFERENT SCRATCH PAD.
1229                                    ************************************************************
1230
1231
1232                                    ******************** TEST 42 ****************************
1233                                    *
1234                                    *IN THIS TEST WE ARE GOING TO MAKE SURE THAT THE
1235                                    *SIGNAL "MOV INST H" (AND ITS ASSOC. TRIBS) DOESN'T GET
1236                                    *STUCK HIGH.  IN ORDER TO DO THIS WE'LL CLEAR THE PC HIGH REG
1237                                    *PUT KNOWN DATA IN THE BREG AND SP1 THEN WE'LL BRANCH
1238                                    *WITH CROM BITS 0-3 SET AS WELL AS CROM BIT 9 WITH CROM BITS 8 AND 11 CLEAR.
1239                                    *IF "MOV INST H" GETS STUCK HIGH, THE PC REG HIGH WILL GET LOADED
1240                                    *WITH THE CONTENTS OF THE ALU
1241                                    ************************************************************
1242
1243
1244                                    ******************** TEST 43 ****************************
1245                                    *TEST TAHT MASTER CLEAR, CLEARS BITS IN THE NPR CONTROL REGISTER AND
1246                                    *MICROPROCESSOR MISCELLANEOUS REGISTER-FIRST WE'LL SET THE
1247                                    *PRIORITY UP SO THAT WHEN WE SET THE BUS REQUEST BIT THAT IT WON'T BUG US
1248                                    *THEN WE'LL SET ALL THE BITS IN BOTH REGS EXCEPT THE
1249                                    *NPR REQUEST. WE'LL LOOK TO SEE THAT ALL GOT SET, NEXT
1250                                    *WE'LL DO A MASTER CLEAR AND BE SURE THAT THEY ALL CLEAR.
1251                                    ************************************************************
1252
```

8.0 ERROR INFORMATION

8.1 ERROR REPORTING

ERRORS ARE REPORTED BY THE PROGRAM AS THEY OCCUR (IF NOT INHIBITED). THE REPORT CONFORMS TO THE DIAGNOSTIC SUPERVISOR ERROR REPORT FORMAT, AND CONSISTS OF A DESCRIPTION OF THE ERROR, THE TEST NUMBER, SUBTEST NUMBER, PC OF THE ERROR CALL, DEVICE ADDRESS, AND BASIC AND EXTENDED ERROR INFORMATION.

THE FOLLOWING EXAMPLES PROVIDE TYPICAL ERROR REPORTS:

```
CZDMQ DVC FTL ERR 00045 TST 027 SUB 000 PC:022572

MASTER  CLEAR  FAILED  TO  CLEAR  PC  REG,  CONTENTS=000624
CZDMQ DVC FTL ERR 00015 TST 042 SUB 000 PC:027234

UNIT=00, FAILING UNIT ADDRESS=160170
JUMP TEST ERROR
FROM ADDR     TO ADDR      BAD ADDR
000402        000000       000114
```

FOR ALL OTHER ERRORS, THE REPORT MAY BE MORE EXTENSIVE AND REQUIRE ADDITIONAL DATA TO BE REPORTED.

9.0 HISTORY

MODIFIED AUGUST 1980 FOR THE FOLLOWING REASONS:

1) CANCELL DEPO CZDMQA1
2) CANCELL DEPO CZDMQA2
3) DETECT BAD TIMEING ON INTERNAL CLOCK.

a

```
1301
1302
1303
1304
1305
```

```
1306                                    .TITLE CZDMQB0 M8207 STATIC DIAG #2
1307            002000                          .=2000
1308
1309
1310
1311
1312
1313
1314                                            .MCALL  SVC
1315    002000                                  SVC                             ; INITIALIZE SUPERVISOR MACROS
1316
1317
1318
1319
1320
1321    002000                                  BGNMOD  CZDMQ
1322
1323
1324            000000                  $LSTIN= 0
1325            000000                  $LSTTAG= 0
1326            000000                  SVCINS= 0       ; LIST INSTRUCTIONS, SHIFTED RIGHT
1327            000000                  SVCTST= 0       ; LIST TEST TAGS, SHIFTED RIGHT
1328            000000                  SVCSUB= 0       ; LIST SUBTEST TAGS, SHIFTED RIGHT
1329            000000                  SVCGBL= 0       ; LIST GLOBAL TAGS, SHIFTED RIGHT
1330            000000                  SVCTAG= 0       ; LIST OTHER TAGS, SHIFTED RIGHT
1331
1332                                    ;       CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
1333                                    ;       TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS.  CHANGE THE
1334                                    ;       SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS.  YOU MAY
1335                                    ;       CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
1336
1337
```

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

002000                                  POINTER BGNAU,BGNDU


                                        HEADER  CZDMQ,B,0,240.,0
002000                          L$NAME::                ;DIAGNOSTIC NAME
002000  103                             .ASCII /C/
002001  132                             .ASCII /Z/
002002  104                             .ASCII /D/
002003  115                             .ASCII /M/
002004  121                             .ASCII /Q/
002005  000                             .BYTE   0
002006  000                             .BYTE   0
002007  000                             .BYTE   0
002010                          L$REV::                 ;REVISION LEVEL
002010  102                             .ASCII  /B/
002011                          L$DEPO::                ;0
002011  060                             .ASCII  /0/
002012                          L$UNIT::                ;NUMBER OF UNITS
002012  000000                          .WORD   0
002014                          L$TIML::                ;LONGEST TEST TIME
002014  000360                          .WORD   240.
002016                          L$HPCP::                ;POINTER TO H.W. QUES.
002016  027340                          .WORD   L$HARD
002020                          L$SPCP::                ;POINTER TO S.W. QUES.
002020  000000                          .WORD   0
002022                          L$HPTP::                ;PTR. TO DEF. H.W. PTABLE
002022  002262                          .WORD   L$HW
002024                          L$SPTP::                ;PTR. TO S.W. PTABLE
002024  000000                          .WORD   0
002026                          L$LADP::                ;DIAG. END ADDRESS
002026  030140                          .WORD   L$LAST
002030                          L$STA::                 ;RESERVED FOR APT STATS
002030  000000                          .WORD   0
002032                          L$CO::
002032  000000                          .WORD   0
002034                          L$DTYP::                ;DIAGNOSTIC TYPE
002034  000000                          .WORD   0
002036                          L$APT::                 ;APT EXPANSION
002036  000000                          .WORD   0
002040                          L$DTP::                 ;PTR. TO DISPATCH TABLE
002040  002132                          .WORD   L$DISPATCH
002042                          L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
002042  000000                          .WORD   0
002044                          L$ENVI::                ;FLAGS DESCRIBE HOW IT WAS SETUP
002044  000000                          .WORD   0
002046                          L$EXP1::                ;EXPANSION WORD
002046  000000                          .WORD   0
002050                          L$MREV::                ;SVC REV AND EDIT #
002050  003                             .BYTE   C$REVISION
002051  003                             .BYTE   C$EDIT
```

```
1394  002052
1395  002052  000000          L$EF::          .WORD   0               ;DIAG. EVENT FLAGS
1396  002054  000000                          .WORD   0
1397  002056                  L$SPC::
1398  002056  000000                          .WORD   0
1399  002060                  L$DEVP::                                ; POINTER TO DEVICE TYPE LIST
1400  002060  002730                          .WORD   L$DVTYP
1401  002062                  L$REPP::                                ;PTR. TO REPORT CODE
1402  002062  000000                          .WORD   0
1403  002064                  L$EXP4::
1404  002064  000000                          .WORD   0
1405  002066                  L$EXP5::
1406  002066  000000                          .WORD   0
1407  002070                  L$AUT::                                 ;PTR. TO ADD UNIT CODE
1408  002070  012144                          .WORD   L$AU
1409  002072                  L$DUT::                                 ;PTR. TO DROP UNIT CODE
1410  002072  012140                          .WORD   L$DU
1411  002074                  L$LUN::                                 ;LUN FOR EXERCISERS TO FILL
1412  002074  000000                          .WORD   0
1413  002076                  L$DESP::                                ;POINTER TO DIAG. DESCRIPTION
1414  002076  002312                          .WORD   L$DESC
1415  002100                  L$LOAD::                                ;GENERATE SPECIAL AUTOLOAD EMT
1416  002100  104035                          EMT     E$LOAD
1417  002102                  L$ETP::                                 ;POINTER TO ERRTBL
1418  002102  000000                          .WORD   0
1419  002104                  L$ICP::                                 ;PTR. TO INIT CODE
1420  002104  011340                          .WORD   L$INIT
1421  002106                  L$CCP::                                 ;PTR. TO CLEAN-UP CODE
1422  002106  012134                          .WORD   L$CLEAN
1423  002110                  L$ACP::                                 ;PTR. TO AUTO CODE
1424  002110  012042                          .WORD   L$AUTO
1425  002112                  L$PRT::                                 ;PTR. TO PROTECT TABLE
1426  002112  002122                          .WORD   L$PROT
1427  002114                  L$TEST::                                ;TEST NUMBER
1428  002114  000000                          .WORD   0
1429  002116                  L$DLY::                                 ;DELAY COUNT
1430  002116  000000                          .WORD   0
1431  002120                  L$HIME::                                ;PTR. TO HIGH MEM
1432  002120  000000                          .WORD   0
1433
1434
1435  002122                          BGNPROT
1436  002122                  L$PROT::
1437  002122  177777                  .WORD   -1
1438  002124  177777                  .WORD   -1
1439  002126  177777                  .WORD   -1
1440  002130                          ENDPROT
1441
```

```
                                .SBTTL  DISPATCH TABLE

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

002130                                  DISPATCH 43
002130  000053                         .WORD   43
002132                          L$DISPATCH::
002132  012146                          .WORD   T1
002134  012256                          .WORD   T2
002136  012422                          .WORD   T3
002140  012552                          .WORD   T4
002142  012712                          .WORD   T5
002144  013154                          .WORD   T6
002146  013356                          .WORD   T7
002150  013570                          .WORD   T8
002152  014112                          .WORD   T9
002154  014470                          .WORD   T10
002156  014736                          .WORD   T11
002160  015202                          .WORD   T12
002162  015432                          .WORD   T13
002164  015676                          .WORD   T14
002166  016142                          .WORD   T15
002170  016406                          .WORD   T16
002172  016652                          .WORD   T17
002174  017116                          .WORD   T18
002176  017362                          .WORD   T19
002200  017642                          .WORD   T20
002202  020122                          .WORD   T21
002204  020376                          .WORD   T22
002206  020654                          .WORD   T23
002210  021130                          .WORD   T24
002212  021404                          .WORD   T25
002214  021576                          .WORD   T26
002216  021744                          .WORD   T27
002220  022322                          .WORD   T28
002222  022560                          .WORD   T29
002224  022774                          .WORD   T30
002226  023210                          .WORD   T31
002230  023452                          .WORD   T32
002232  024044                          .WORD   T33
002234  025116                          .WORD   T34
002236  025216                          .WORD   T35
002240  025364                          .WORD   T36
002242  026042                          .WORD   T37
002244  026202                          .WORD   T38
002246  026310                          .WORD   T39
002250  026446                          .WORD   T40
002252  026544                          .WORD   T41
002254  026644                          .WORD   T42
002256  026770                          .WORD   T43
```

```
1498
1499
1500
```

```
                        .SBTTL DEFAULT HARDWARE P-TABLE

                        ;////////////////////////////////////////////////////////////////////////////
                        ;/ THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
                        ;/ THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
                        ;/ IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
                        ;////////////////////////////////////////////////////////////////////////////

                        .ENABL  AMA
002260                          BGNHW   DFPTBL
002260  000013                  .WORD   L10001-L$HW/2
002262                  L$HW::
002262                  DFPTBL::
002262  000007                  .WORD   7               ;MICRO-CPU TYPE.
002264  160170                  .WORD   160170  ;M8200,4,7 CRS ADDRESS
002266  000300                  .WORD   300     ;M8200,4,7 VECTOR ADDRESS
002270  005000                  .WORD   5000    ;INTERRUPT PRIORITY LEVEL
002272  000003                  .WORD   3       ;LINE UNIT TYPE
002274  000056                  .WORD   56      ;SWITCH PACK #1 (DDCMP LINE #)
002276  000000                  .WORD   0       ;SWITCH PACK #2 (BM873 BOOT ADDRESS)
002300  000000                  .WORD   0       ;SWITCH PACK #3
002302  000000                  .WORD   0       ;TEST CONNECTOR INSTALLED FLAG
002304  000004                  .WORD   4       ;CONTAINS BAUD RATE 4=56K BAUD DEFAULT
                                                ;0=2.4K , 1=4.8K , 2=9.6K , 3=19.2K , 4=56K
                                                ;5=250K , 6=500K , 7=1 MEG BAUD
002306  000000                  .WORD   0       ;0=RUN SW OFF, 1=SW ON

002310                          ENDHW
002310                  L10001:
```

```
1530                                    .SBTTL  SOFTWARE P-TABLE
1531
1532                                    ;////////////////////////////////////////////////////////////////////////
1533                                    ;/ THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
1534                                    ;/ PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
1535                                    ;////////////////////////////////////////////////////////////////////////
1536
1537  002310                                    BGNSW   SFPTBL
1538  002310  000000                            .WORD   L10002-L$SW/2
1539  002312                            L$SW::
1540  002312                            SFPTBL::
1541
1542
1543  002312                                    ENDSW
1544  002312                            L10002:
1545
1546
1547
1548
1549
1550
```

```
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562                                    .SBTTL  GLOBAL EQUATES SECTION
1563
1564
1565                                    ;////////////////////////////////////////////////////////////////////////////////
1566                                    ;/      THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
1567                                    ;/      ARE USED IN MORE THAN ONE TEST.
1568                                    ;////////////////////////////////////////////////////////////////////////////////
1569
1570  002312                                    EQUALS
1571                                    ;
1572                                    ; BIT DIFINITIONS
1573                                    ;
1574            100000                  BIT15== 100000
1575            040000                  BIT14== 40000
1576            020000                  BIT13== 20000
1577            010000                  BIT12== 10000
1578            004000                  BIT11== 4000
1579            002000                  BIT10== 2000
1580            001000                  BIT09== 1000
1581            000400                  BIT08== 400
1582            000200                  BIT07== 200
1583            000100                  BIT06== 100
1584            000040                  BIT05== 40
1585            000020                  BIT04== 20
1586            000010                  BIT03== 10
1587            000004                  BIT02== 4
1588            000002                  BIT01== 2
1589            000001                  BIT00== 1
1590                                    ;
1591            001000                  BIT9==  BIT09
1592            000400                  BIT8==  BIT08
1593            000200                  BIT7==  BIT07
1594            000100                  BIT6==  BIT06
1595            000040                  BIT5==  BIT05
1596            000020                  BIT4==  BIT04
1597            000010                  BIT3==  BIT03
1598            000004                  BIT2==  BIT02
1599            000002                  BIT1==  BIT01
1600            000001                  BIT0==  BIT00
1601                                    ;
1602                                    ; EVENT FLAG DEFINITIONS
1603                                    ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
1604                                    ;
1605            000040                  EF.START==      32.                     ; START COMMAND WAS ISSUED
1606            000037                  EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
```

```
1607    000036          EF.CONTINUE==   30.             ; CONTINUE COMMAND WAS ISSUED
1608    000035          EF.NEW==        29.             ; A NEW PASS HAS BEEN STARTED
1609    000034          EF.PWR==        28.             ; A POWER-FAIL/POWER-UP OCCURRED
1610                    ;
1611                    ;
1612                    ; PRIORITY LEVEL DEFINITIONS
1613                    ;
1614    000340          PRI07== 340
1615    000300          PRI06== 300
1616    000240          PRI05== 240
1617    000200          PRI04== 200
1618    000140          PRI03== 140
1619    000100          PRI02== 100
1620    000040          PRI01== 40
1621    000000          PRI00== 0
1622                    ;
1623                    ;OPERATOR FLAG BITS
1624                    ;
1625    000004          EVL==       4
1626    000010          LOT==      10
1627    000020          ADR==      20
1628    000040          IDU==      40
1629    000100          ISR==     100
1630    000200          UAM==     200
1631    000400          BOE==     400
1632    001000          PNT==    1000
1633    002000          PRI==    2000
1634    004000          IXE==    4000
1635    010000          IBE==   10000
1636    020000          IER==   20000
1637    040000          LOE==   40000
1638    100000          HOE==  100000
1639
1640
1641
1642
1643                    ;;*****************************************************************************
1644                    ;* PROGRAM EVENT FLAG DEFINITIONS
1645                    ;;*****************************************************************************
1646
1647
1648
```

```
                                .SBTTL  GLOBAL DATA SECTION

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

                                ;;*****************************************************************************
                                ;* STORAGE FOR DEVICE REGISTERS
                                ;;*****************************************************************************
002312                                  DESCRIPT        <M8207 DIAG. #2 OF 2>
002312                          L$DESC::
002312  034115  030062  020067         .ASCIZ  /M8207 DIAG. #2 OF 2/
002320  044504  043501  020056
002326  031043  047440  020106
002334  000062

                                        .EVEN

                                ;*****************************************************************************
                                ;* PROGRAM CONTROL PARAMETERS
                                ;*****************************************************************************
002336  000000                  NEXT:   .WORD   0               ;ADDRESS OF NEXT TEST TO BE EXECUTED
002340  000000                  LOCK:   .WORD   0               ;ADDRESS FOR LOCK CURRENT DATA

                                ;*****************************************************************************
                                ;* MISCELLANEOUS STORAGE
                                ;*****************************************************************************
002342  000000                  LOGDEV: .WORD   0               ;LOGICAL DEVICE NUMBER
002344  000000                  PSTACK: .WORD   0               ;BASE LEVEL PROGRAM STACK POINTER
002346  000000                  SUBRPC: .WORD   0               ;PC OF SUBR CALL FOR ERROR REPORTS
002350  000000                  ERRFLG: .WORD   0               ;SUBROUTINE ERROR FLAG
002352  000000                  RETADR: .WORD   0               ;SUBR ERROR RETURN ADDRESS
002354  000000                  STRTSW: .WORD   0               ;SWITCHES AT START OF PROGRAM
002356  000000                  STAT:   .WORD   0               ;KM STATUS WORD STORAGE
002360  000000                  CLKX:   .WORD   0               ;
002362  000000                  MASKX:  .WORD   0               ;
002364  000000                  SAVSP:  .WORD   0               ;STACK POINTER STORAGE
002366  000000                  SAVPC:  .WORD   0               ;PROGRAM COUNTER STORAGE
002370  000000                  ZERO:   .WORD   0               ;
002372  000001                  ONE:    .WORD   1               ;
002374  000000                  MEMLIM: .WORD   0               ;HIGHEST LOCATION FOR NPR'S
002376  000001                  KMACTV: .BLKW   1               ;M8200,4,7 SELECTED ACTIVE
002400  000001                  KMNUM:  .BLKW   1               ;OCTAL NUMBER OF M8200,4,7'S
002402  000001                  SAVACT: .BLKW   1               ;ORIGINAL ACTIVE DEVICES
002404  000001                  SAVNUM: .BLKW   1               ;WORKABLE NUMBER
002406  000000                  FLAG:   .WORD   0               ;SCRATCH STORAGE
002410  000000                  RUN:    .WORD   0               ;POINTER TO RUNNING DEVICES
002412  000000                  FADR:   .WORD   0
002414  000000                  WTYPE:  .WORD   0               ;M82XX NUMBER FOR TYPE OF MICO-CPU
002416  000000                  $REG5:  .WORD   0               ;STORAGE USED FOR ERROR MSG DATA
002420  000000                  $REG4:  .WORD   0
002422  000000                  $REG3:  .WORD   0
002424  000000                  $REG2:  .WORD   0
002426  000000                  $REG1:  .WORD   0
002430  000000                  $REG0:  .WORD   0
002432  000000                  TYPE:   .WORD   0               ;=0 FOR DMP,=1 FOR M8206
```

```
1705  002434  000000                  MR0:    .WORD   0               ;MEMLOC USED INSTEAD OF R0.
1706  002436  003777                  MEMSZ:  .WORD   3777            ;INDICATES MEMORIE SIZE, LAST ADDR.
1707  002440  000000                  TEMP:   .WORD   0
1708  002442  000000                  $TEMP0: .WORD   0
1709  002444  000000                  $TMP0:  .WORD   0
1710  002446  000000                  $GDADR: .WORD   0               ;CONTAINS ADDRESS OF 'GOOD' DATA
1711  002450  000000                  $BDADR: .WORD   0               ;CONTAINS ADDRESS OF 'BAD' DATA
1712  002452  000000                  $GDDAT: .WORD   0               ;CONTAINS 'GOOD' DATA
1713  002454  000000                  $BDDAT: .WORD   0               ;CONTAINS 'BAD' DATA
1714  002456  000000                          .WORD   0               ;RESERVED--NOT TO BE USED
1715  002460  000000                          .WORD   0
1716  002462  000000                  FTIME:  .WORD   0
1717  002464  000000                  SAVE4:  .WORD   0
1718  002466  000000                  SAVE6:  .WORD   0
1719  002470  000000                  RUNB:   .WORD   0               ;0= RUN OFF, 1= RUN SW ON
1720  002472  000000                  RUNINH: .WORD   0               ;0=RUN SW OFF, 1=RUN SW ON
1721
1722                                  ;*******************************************************************************
1723                                  ;* PROGRAM CONTROL FLAGS
1724                                  ;*******************************************************************************
1725  002474     000                  INIFLG: .BYTE   0               ;PROGRAM INITIALIZING FLAG
1726          002476                          .EVEN
1727  002476     000                  LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
1728  002477     000                  QV.FLG: .BYTE   0               ;QUICK VERIFY FLAG
1729                                          .EVEN
1730
1731                                  ;*******************************************************************************
1732                                  ;* DEFINITION OF M8200,4,7 STATUS WORDS - STAT1,STAT2,STAT3
1733                                  ;*
1734                                  ;*
1735                                  ;*      STAT1 - BITS 00-08 IS M8200,4,7 VECTOR ADDRESS
1736                                  ;*              BIT15=1 LINE UNIT IS AN M8203
1737                                  ;*              BIT14=0 NO TEST CONNECTOR(S) USED
1738                                  ;*              BIT14=1 H-XXX TEST CONNECTOR WILL BE USED
1739                                  ;*              BIT13=0 LINE UNIT IS AN M8201
1740                                  ;*              BIT13=1 LINE UNIT IS AN M8202
1741                                  ;*              BIT12=1 NO LINE UNIT
1742                                  ;*              BITS 09-11 IS M8200,4,7 PRIORITY LEVEL
1743                                  ;*
1744                                  ;*      STAT2 - LOW BYTE IS SWITCH PACK #1 (DDCMP LINE NUMBER)
1745                                  ;*              HIGH BYTE IS SWITCH PACK #2 (BM873 BOOT ADDRESS)
1746                                  ;*
1747                                  ;*      STAT3 - BIT0=1 DO FREE RUNNING TESTS ON M8200,4,7
1748                                  ;*
1749                                  ;*******************************************************************************
1750  002500  000000                  STAT1:  .WORD   0
1751  002502  000000                  STAT2:  .WORD   0
1752  002504  000000                  STAT3:  .WORD   0
1753
1754                                  ;*******************************************************************************
1755                                  ;* POINTERS TO M8200,4,7 VECTORS AND REGISTERS
1756                                  ;*******************************************************************************
1757  002506  000000                  KMRVEC: 0       ;POINTER TO M8200,4,7 RCV INTRPT VECTOR
1758  002510  000000                  KMRLVL: 0       ;POINTER TO M8200,4,7 RCV INTRPT SERVICE PS
1759  002512  000000                  KMTVEC: 0       ;POINTER TO M8200,4,7 TX INTRPT VECTOR
1760  002514  000000                  KMTLVL: 0       ;POINTER TO M8200,4,7 TX INTRPT SERVICE PS
```

```
1761  002516  000000                  KMCSR:  0       ;POINTER TO M8200,4,7 CONTROL STATUS REGISTER
1762  002520  000000                  KMCSRH: 0       ;POINTER TO M8200,4,7 CONTROL STATUS REGISTER HIGH BYTE
1763  002522  000000                  KMCTL:  0       ;POINTER TO M8200,4,7 CONTROL OUT REGISTER
1764  002524  000000                  KMP04:  0       ;POINTER TO M8200,4,7 PORT REGISTER - SEL4
1765  002526  000000                  KMP06:  0       ;POINTER TO M8200,4,7 PORT REGISTER - SEL6
1766
1767                                  ;;****  PRIMARY REG ADRS STORAGE FOR THIS UNIT  ******
1768                                  ;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE
1769  002530                          REGADR:
1770
1771                                  ;;****  STACK USED FOR SUBROUTINE LINKAGE  *****
1772  002530  000100                          .BLKW   100
1773  002730                          SSTACK:
1774
1775
1776
1777
1778
1779
1780
```

```
                                          .SBTTL  GLOBAL TEXT SECTION

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

                                          ;;*********************************************************************
                                          ;* NAMES OF DEVICES SUPPORTED BY PROGRAM
                                          ;;*********************************************************************
 002730                                           DEVTYP  <M8200,M8204,OR M8207>
 002730                                   L$DVTYP::
 002730  034115  030062  026060                   .ASCIZ  /M8200,M8204,OR M8207/
 002736  034115  030062  026064
 002744  051117  046440  031070
 002752  033460     000
         002756                                   .EVEN



                                          ;
                                          ; FORMAT STATEMENTS USED IN PRINT CALLS
                                          ;
```

```
1810                            .SBTTL  GLOBAL SUBROUTINES
1811
1812                            ;////////////////////////////////////////////////////////////////////////////////
1813                            ;/      THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST
1814                            ;////////////////////////////////////////////////////////////////////////////////
1815
1816
1817
1818                            ;-------------------------------------------------------------------
1819                            ; MACRO'S NEEDED TO CALL SUBROUTINES
1820                            ;-------------------------------------------------------------------
1821
1822                            .MACRO  POPSP2
1823                                    22626
1824                            .ENDM
```

```
1825
1826
1827
1828
1829
```

```
1830
1831                    ;////////////////////////////////////////////////////////////////////////////////
1832                    ;/      THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST
1833                    ;////////////////////////////////////////////////////////////////////////////////
1834
1835
1836
1837                    ;-------------------------------------------------------------------
1838                    ; MACRO'S NEEDED TO CALL SUBROUTINES
1839                    ;-------------------------------------------------------------------
1840
1841                    .MACRO  K4ONLY  ?N2
1842                            CMP     MEMSZ,#2000
1843                            BNE     N2
1844                            EXIT    TST
1845                            .ENDM
1846                    .MACRO  ED$CALL XY
1847                            .LIST
1848                            ;*************** TEST 'XY' ***************
1849                            .NLIST
1850                            .ENDM
1851                            .MACRO  BADHEAD
1852                            .RADIX  10
1853                            ED$CALL \T$TESTNUM+1
1854                            .RADIX  8
1855                            .ENDM
```

```
1856                            .MACRO  MYINT
1857                                    .LIST
1858                                    MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
1859                                    .NLIST
1860                            .ENDM
1861
1862                            .MACRO  MACEX   ?N2
1863                                    .LIST
1864                                    ;DO NOT DO TEST IF M8200
1865                                    .NLIST
1866                                    TST     TYPE
1867                                    BNE     N2
1868                                    EXIT    TST
1869                            N2:
1870                            .ENDM   MACEX
1871                            .MACRO  MACEX2  ?N2
1872                                    .LIST
1873                                    ;DO NOT DO TEST IF M8200
1874                                    .NLIST
1875                                    CMP     WTYPE,#0
1876                                    BNE     N2
1877                                    EXIT TST
1878                            N2:
1879                            .ENDM   MACEX2
1880                            .MACRO  K4ONLY  ?N2
1881                                    .LIST
1882                                    ;DO NOT DO TEST IF M8200, OR M8204
1883                                    .NLIST
1884                                    CMP     MEMSZ,#2000
1885                                    BNE     N2
1886                                    EXIT    TST
1887                            N2:
1888                                                            ;NOTE THIS TEST IS ONLY DESIGNED FOR 4K MODULE.
1889                            .ENDM
1890
1891                            .MACRO  CLRMAR
1892                                    ROMCLK
1893                                    004000
1894                            .ENDM   CLRMAR
1895                            .MACRO  ROMCLK
1896                            .LIST
1897                                    JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
1898                            .NLIST
1899                            .ENDM
1900
1901                            .MACRO  SROMCLK
1902                                    .LIST
1903                                    JSR     R5,.SROMCLK
1904                                    .NLIST
1905                            .ENDM
1906                            .MACRO  SKIP06  NNN
1907                                    .LIST
1908                                    ;GOTO 'NNN' IF M8206
1909                                    .NLIST
1910                                    CMP     WTYPE,#6                ;SEE IF M8206
1911                                    BEQ     NNN
```

```
1912                                    .ENDM
1913                                    .MACRO  SKIP07  NNN
1914                                            .LIST
1915                                            ;GOTO 'NNN' IF M8207
1916                                            .NLIST
1917                                            CMP     WTYPE,#7                ;SEE IF M8200,4,7
1918                                            BEQ     NNN
1919                                    .ENDM
1920                                    .MACRO  SKIP04  NNN
1921                                            .LIST
1922                                            ;GOTO 'NNN' IF M8204
1923                                            .NLIST
1924                                            CMP     WTYPE,#4                ;SEE IF M8204
1925                                            BEQ     NNN
1926                                    .ENDM
1927                                    .MACRO  MSTCLR
1928                                    JSR     R5,.MSTCLR              ;CLEAR M8200,4,7
1929                                    .ENDM
1930
1931    002756                          .MSTCLR:
1932    002756  112777  000100  177534          MOVB    #BIT6,@KMCSRH   ;SET INST.
1933    002764  142777  000300  177526          BICB    #BIT6!BIT7,@KMCSRH
1934    002772  000205                          RTS     R5
1935
1936    002774  000024                  PATCH:  .BLKW   20.             ;PATCH AREA.
1937
1938
1939
1940    003044                          ENDBUG:
1941                                    ;       UNSAFE TO PATCH ANY OTHER AREA.
1942    003044                          .ROMCLK:
1943    003044  000240                          NOP
1944    003046  000240                          NOP
1945    003050  152777  000002  177442  .REGT:  BISB    #BIT1,@KMCSRH
1946    003056  012577  177444                  MOV     (R5)+,@KMPO6
1947    003062  152777  000003  177430          BISB    #BIT1!BIT0,@KMCSRH
1948    003070  142777  000007  177422          BICB    #BIT2!BIT1!BIT0,@KMCSRH
1949    003076  000205                          RTS     R5
1950
1951    003100                          .SROMCLK:
1952    003100  000240                          NOP
1953    003102  022737  000006  002414          CMP     #6,WTYPE
1954    003110  001357                          BNE     .REGT
1955    003112  152777  000002  177400          BISB    #BIT1,@KMCSRH
1956    003120  012577  177402                  MOV     (R5)+,@KMPO6
1957    003124  000240                          NOP
1958    003126  000240                          NOP
1959    003130  142777  000007  177362          BICB    #7,@KMCSRH
1960    003136                          1$:
1961    003136  152777  000001  177354          BISB    #BIT0,@KMCSRH           ;STEP INSTR.
1962    003144  142777  000007  177346          BICB    #BIT2!BIT1!BIT0,@KMCSRH
1963    003152  000240                          NOP
1964    003154  000240                          NOP
1965    003156  152777  000002  177334          BISB    #2,@KMCSRH
1966    003164                          2$:
1967    003164  000205                          RTS     R5
```

```
1968  003166                          CLRALL:
1969                                          ;CLEARS C & Z BITS AND BR
1970  003166                                  ROMCLK
1971  003166  004537  003044                 JSR     R5,..ROMCLK       ;CLOCK INSTRUCTION
1972  003172  000400                         400                       ;0 TO BR
1973  003174                                  ROMCLK
1974  003174  004537  003044                 JSR     R5,..ROMCLK       ;CLOCK INSTRUCTION
1975  003200  063220                         63220                     ;SP(0) TO BR
1976  003202                                  ROMCLK
1977  003202  004537  003044                 JSR     R5,..ROMCLK       ;CLOCK INSTRUCTION
1978  003206  060400                         60400                     ;BR,SP(0) + BR
1979  003210                                  SROMCLK
1980  003210  004537  003100                 JSR     R5,..SROMCLK
1981  003214  000000                         0
1982  003216  000207                         RTS     PC
1983
1984  003220                          SETBR0:
1985                                          ;SETS BR0 BIT
1986  003220                                  ROMCLK
1987  003220  004537  003044                 JSR     R5,..ROMCLK       ;CLOCK INSTRUCTION
1988  003224  000401                         401                       ;1 TO BR
1989  003226  000207                         RTS     PC
1990
1991  003230                          SETBR1:
1992                                          ;THIS SUBROUTINE SETS BR1 BIT
1993
1994  003230                          ROMCLK                            ;NEXT WORD IS INSTRUCTION
1995  003230  004537  003044                 JSR     R5,..ROMCLK       ;CLOCK INSTRUCTION
1996  003234  000402                         000402                    ;BR_002
1997  003236  000207                         RTS     PC
1998
1999  003240                          SETBR4:
2000                                          ;THIS SUBROUTINE SETS BR4 BIT
2001
2002  003240                                  ROMCLK                    ;NEXT WORD IS INSTRUCTION
2003  003240  004537  003044                 JSR     R5,..ROMCLK       ;CLOCK INSTRUCTION
2004  003244  000420                         420
2005  003246  000207                         RTS     PC
2006
2007  003250                          SETBR7:
2008                                          ;THIS SUBROUTINE SETS BR7 BIT
2009
2010  003250                                  ROMCLK                    ;NEXT WORD IS INSTRUCTION
2011  003250  004537  003044                 JSR     R5,..ROMCLK       ;CLOCK INSTRUCTION
2012  003254  000600                         600
2013  003256  000207                         RTS     PC
2014
2015  003260                          SETC:
2016                                          ;THIS SUBROUTINE SETS THE C BIT
2017
2018  003260                                  ROMCLK                    ;NEXT WORD IS INSTRUCTION
2019  003260  004537  003044                 JSR     R5,..ROMCLK       ;CLOCK INSTRUCTION
2020  003264  000777                         000777                    ;BR_377
2021  003266                                  ROMCLK                    ;NEXT WORD IS INSTRUCTION
2022  003266  004537  003044                 JSR     R5,..ROMCLK       ;CLOCK INSTRUCTION
2023  003272  063220                         063220                    ;SP(0)_BR
```

```
2024  003274                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION
2025  003274  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
2026  003300  060400                          060400                          ;BR SP(0)+BR
2027  003302                                  SROMCLK                         ;NOW WE MUST CLOCK THE BITS INTO IBUS <13>
2028  003302  004537  003100                  JSR     R5,.SROMCLK
2029  003306  000000                          0
2030  003310  000207                          RTS     PC
2031
2032  003312                          SETZ:
2033                                          ;THIS SUBROUTINE SETS THE Z BIT
2034
2035  003312                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION
2036  003312  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
2037  003316  000777                          000777                          ;BR 377
2038  003320                                  SROMCLK                         ;NOW CLOCK THE BITS INTO IBUS<13>
2039  003320  004537  003100                  JSR     R5,.SROMCLK
2040  003324  000777                          0777
2041  003326  000207                          RTS     PC
2042
2043  003330                          RAMDAT:
2044                                          ;THIS SUBROUTINE LOADS R4 WITH THE LOWEST
2045                                          ;8 BITS OF THE CRAM PC.
2046
2047  003330  005004                          CLR     R4
2048  003332  017605  000000                  MOV     @(SP),R5                ;GOOD DATA
2049  003336  062716  000002                  ADD     #2,(SP)                 ;ADJUST STACK
2050  003342                                  SKIP06  1$      ;IF M8206,WE'LL GET PC A DIFFERENT WAY.
2051                                          ;GOTO 1$ IF M8206
2052  003352                                  SKIP07  1$      ;IF M8200,4,7 WE'LL GET PC A DIFFERENT WAY.
2053                                          ;GOTO 1$ IF M8207
2054  003362  005011                          CLR     (R1)                    ;CLEAR BIT10
2055  003364  052711  000400                  BIS     #BIT8,(R1)              ;CLOCK INSTRUCTION IN CRAM THAT
2056                                                                          ;JUMPED TO, IT LOADS BR WITH IT
2057  003370  005011                          CLR     (R1)                    ;CLR BIT8
2058  003372                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION
2059  003372  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
2060  003376  061225                          061225                          ;MOV BR TO PORT 5
2061  003400  116104  000005                  MOVB    5(R1),R4                ;PUT "FOUND" IN R4
2062  003404  000207                          RTS     PC                      ;RETURN
2063
2064  003406                          1$:     ROMCLK                          ;READ PC LOW REG DIRECTLY.
2065  003406  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
2066  003412  121244                          121244                          ;IBUS*<12> TO PORT 4
2067  003414  116104  000004                  MOVB    4(R1),R4                ;PUT INTO R4
2068  003420  000207                          RTS PC                          ;EXIT
2069
2070  003422                          WROM:
2071                                          ;THIS SUBROUTINE WRITES THE ROMMAP INTO THE CRAM
2072
2073                                  ;       BIT     #BIT15,STAT1            ;BE SURE M8200,4,7 HAS CRAM
2074                                  ;       BEQ     2$                      ;SKIP IF NO CRAM
2075  003422                                  SKIP07  2$
2076                                          ;GOTO 2$ IF M8207
2077  003432  005000                          CLR     R0                      ;R0=CRAM ADDRESS
2078  003434  012702  012146                  MOV     #ROMMAP,R2              ;R2 POINTS TO ROMMAP
2079  003440  012711  002000          1$:     MOV     #BIT10,(R1)             ;SET ROM0
```

```
                                        MOV     R0,4(R1)        ;LOAD CRAM ADDRESS
                                        MOV     (R2)+,6(R1)     ;LOAD WORD TO BE WRITTEN
                                        BIS     #BIT13,(R1)     ;WRITE IT!
                                        INC     R0              ;NEXT ADDRESS
                                        CMP     MEMSZ,R0        ;DONE YET?
                                        BNE     1$              ;BR IF NO
                                        CLR     (R1)            ;CLEAR SEL0
                                2$:     RTS     PC              ;RETURN

                                MEMSET:
                                        ;THIS SUBROUTINE LOADS CRAM WITH SPECIAL INSTRUCTIONS
                                        ;FOR THE CRAM JUMP TEST. ALL CRAM LOCATIONS ARE LOADED
                                        ;WITH INSTRUCTIONS THAT MOVE A 37 TO THE BR, EXCEPT THE
                                        ;FOLLOWING CRAM ADDRESSES: 0,1,4,7,525,1777. THESE LOCATIONS
                                        ;CONTAIN INSTRUCTIONS WHICH LOAD THE BR WITH THE LOWEST
                                        ;8 BITS OF THAT CRAM ADDRESS.

                                        SKIP07  3$      ;IF M8200,4,7 CAN'T WRITE CRAM!
                                        ;GOTO 3$ IF M8207
                                        CLR     R0              ;R0 = CRAM ADDRESS
                                1$:     MOV     #BIT10,(R1)     ;SET ROMO
                                        MOV     R0,4(R1)        ;LOAD CRAM ADDRESS
                        000006          MOV     #437,6(R1)      ;LOAD INSTRUCTION
                                        BIS     #BIT13,(R1)     ;WRITE INSTRUCTION IN CRAM
                                        INC     R0              ;NEXT ADDRESS
                                        CMP     MEMSZ,R0        ;DONE YET?
                                        BNE     1$              ;BR IF NO
                                        CLR     R0              ;INDEX REGISTER
                                2$:     MOV     #BIT10,(R1)     ;SET ROMO
                        000004          MOV     CRAMA(R0),4(R1) ;LOAD CRAM ADDRESS IN SEL4
                        000006          MOV     INSTU(R0),6(R1) ;LOAD INSTRUCTION TO BE WRITTEN
                                        BIS     #BIT13,(R1)     ;WRITE CRAM!
                                        TST     (R0)+           ;NEXT
                                        CMP     #14,R0          ;DONE YET?
                                        BNE     2$              ;BR IF NO
                                        CLR     (R1)            ;CLEAR ALL BITS
                                3$:     RTS     PC              ;RETURN

                        000001  000004  CRAMA:  .WORD   0,1,4,7,1777,525
                        001777  000525

                                INSTU:  000400          ;BR_0
                                        000401          ;BR_1
                                        000404          ;BR_4
                                        000407          ;BR_7
                                        000777          ;BR_377
                                        000525          ;BR_125


                                        ;ROUTINE TO SAVE GENERAL REGISTERS FOR ERROR ROUTINE.
                                        ; CALL = JSR    PC,SV05
                                SV05:   MOV     R5,$REG5
                                        MOV     R4,$REG4
                                        MOV     R3,$REG3
                                        MOV     R2,$REG2
                                        MOV     R1,$REG1
```

```
 2136  003656  013737  002434  002430          MOV     MR0,$REG0
 2137  003664  000207                          RTS     PC
 2138
 2139
```

```
2140                                    .SBTTL  GLOBAL ERROR REPORT SECTION
2141
2142                                    ;////////////////////////////////////////////////////////////////
2143                                    ;/      THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
2144                                    ;/      THAT ARE USED IN MORE THAN ONE TEST.
2145                                    ;////////////////////////////////////////////////////////////////
2146
2147
2148
2149
2150
2151  003666  047045  047445  022466  TFM1:   .ASCIZ  /%N%06%S4%06%S4%04%N/
2152  003674  032123  047445  022466
2153  003702  032123  047445  022464
2154  003710  000116
2155  003712  047045  047445  022463  TFM2:   .ASCIZ  /%N%03%S7%03%N/
2156  003720  033523  047445  022463
2157  003726  000116
2158  003730  047045  047445  022463  TFM3:   .ASCIZ  /%N%03%S10%03%S4%04%N/
2159  003736  030523  022460  031517
2160  003744  051445  022464  032117
2161  003752  047045     000
2162  003755     045  022516  031517  TFM4:   .ASCIZ  /%N%03%S7%03%N/
2163  003762  051445  022467  031517
2164  003770  047045     000
2165  003773     045  022516  033117  TFM5:   .ASCIZ  /%N%06%S5%06%S3%06%N/
2166  004000  051445  022465  033117
2167  004006  051445  022463  033117
2168  004014  047045     000
2169  004017     045  022516  051101  TFM36:  .ASCIZ  /%N%AREGISTER ADDRESS ERROR,ADDRESS = %06%A,UNIT = %02/
2170  004024  043505  051511  042524
2171  004032  020122  042101  051104
2172  004040  051505  020123  051105
2173  004046  047522  026122  042101
2174  004054  051104  051505  020123
2175  004062  020075  047445  022466
2176  004070  026101  047125  052111
2177  004076  036440  022440  031117
2178  004104     000
2179  004105     045  022516  020101  TFM41:  .ASCIZ  /%N%A CSR HIGH BYTE GOT WRITTEN INTO ON A LOW BYTE XFER/
2180  004112  051503  020122  044510
2181  004120  044107  041040  052131
2182  004126  020105  047507  020124
2183  004134  051127  052111  042524
2184  004142  020116  047111  047524
2185  004150  047440  020116  020101
2186  004156  047514  020127  054502
2187  004164  042524  054040  042506
2188  004172  000122
2189  004174  047045  040445  041440  TFM42:  .ASCIZ  /%N%A CSR LOW BYTE GOT WRITTEN INTO ON A HIGH BYTE XFER/
2190  004202  051123  046040  053517
2191  004210  041040  052131  020105
2192  004216  047507  020124  051127
2193  004224  052111  042524  020116
2194  004232  047111  047524  047440
2195  004240  020116  020101  044510
```

```
2196  004246  044107  041040  052131
2197  004254  020105  043130  051105
2198  004262     000
2199  004263     045  022516  047101  TFM40:  .ASCIZ  /%N%ANEG ADDR TEST DUAL ADDR ERROR-BAD ADDR = %O6/
2200  004270  043505  040440  042104
2201  004276  020122  042524  052123
2202  004304  042040  040525  020114
2203  004312  042101  051104  042440
2204  004320  051122  051117  041055
2205  004326  042101  040440  042104
2206  004334  020122  020075  047445
2207  004342  000066
2208  004344  040445  051440  051103  TFM43:  .ASCIZ  /%A SCRATCH PAD %O3%A DUAL ADDRESS ERROR WITH SP%O2/
2209  004352  052101  044103  050040
2210  004360  042101  022440  031517
2211  004366  040445  042040  040525
2212  004374  020114  042101  051104
2213  004402  051505  020123  051105
2214  004410  047522  020122  044527
2215  004416  044124  051440  022520
2216  004424  031117     000
2217  004427     045  022524  052101  TFM44:  .ASCIZ  /%T%ATHE MAR REG, CONTENTS= %O6/
2218  004434  042510  046440  051101
2219  004442  051040  043505  020054
2220  004450  047503  052116  047105
2221  004456  051524  020075  047445
2222  004464  000066
2223  004466  052045  040445  044124  TFM45:  .ASCIZ  /%T%ATHE PC REG, CONTENTS= %O6/
2224  004474  020105  041520  051040
2225  004502  043505  020054  047503
2226  004510  052116  047105  051524
2227  004516  020075  047445  000066
2228  004524  047045  040445  047516  TFM45A: .ASCII  /%N%ANOTE: THIS ERROR MAY BE FALSELY GENERATED IF THE/
2229  004532  042524  020072  044124
2230  004540  051511  042440  051122
2231  004546  051117  046440  054501
2232  004554  041040  020105  040506
2233  004562  051514  046105  020131
2234  004570  042507  042516  040522
2235  004576  042524  020104  043111
2236  004604  052040  042510
2237  004610  047045  040445  052522          .ASCIZ  /%N%ARUN BIT (SW7 OF E28) IS ON/
2238  004616  020116  044502  020124
2239  004624  051450  033527  047440
2240  004632  020106  031105  024470
2241  004640  044440  020123  047117
2242  004646     000
2243  004647     045  047101  051120  TFM46:  .ASCIZ  ''%ANPR/MISC REGS DATA FAILURE, GOOD =%O6%A, BAD =%O6''
2244  004654  046457  051511  020103
2245  004662  042522  051507  042040
2246  004670  052101  020101  040506
2247  004676  046111  051125  026105
2248  004704  043440  047517  020104
2249  004712  022475  033117  040445
2250  004720  020054  040502  020104
2251  004726  022475  033117     000
```

```
2252  004733     045  050101  020103  TFM47:  .ASCIZ  "%APC INCR. INCORRECT: S/B= %06%A ; WAS = %06"
2253  004740  047111  051103  020056
2254  004746  047111  047503  051122
2255  004754  041505  035124  051440
2256  004762  041057  020075  047445
2257  004770  022466  020101  020073
2258  004776  040527  020123  020075
2259  005004  047445  000066
2260  005010  040515  052123  051105  TMMC:   .ASCIZ  /MASTER CLEAR FAILED TO CLEAR /
2261  005016  041440  042514  051101
2262  005024  043040  044501  042514
2263  005032  020104  047524  041440
2264  005040  042514  051101  000040
2265  005046  047045  052045  047045  FM1:    .ASCIZ  /%N%T%N/
2266  005054     000
2267
2268
2269
2270  005055     000                  EM0:    .ASCIZ  //
2271  005056  051103  046501  042040  EM1:    .ASCIZ  /CRAM DATA ERROR/
2272  005064  052101  020101  051105
2273  005072  047522  000122
2274  005076  051103  046501  042040  EM2:    .ASCIZ  /CRAM DUAL ADDRESSING ERROR/
2275  005104  040525  020114  042101
2276  005112  051104  051505  044523
2277  005120  043516  042440  051122
2278  005126  051117     000
2279  005131     112  046525  020120  EM3:    .ASCIZ  /JUMP ERROR/
2280  005136  051105  047522  000122
2281  005144  051103  046501  045040  EM4:    .ASCIZ  /CRAM JUMP TEST FAULT/
2282  005152  046525  020120  042524
2283  005160  052123  043040  052501
2284  005166  052114     000
2285  005171     111  050117  046440  EM5:    .ASCIZ  /IOP MAIN MEMORY TEST/
2286  005176  044501  020116  042515
2287  005204  047515  054522  052040
2288  005212  051505  000124
2289  005216  047511  020120  040515  EM6:    .ASCIZ  /IOP MAR TEST/
2290  005224  020122  042524  052123
2291  005232     000
2292  005233     102  020122  044522  EM7:    .ASCIZ  /BR RIGHT SHIFT ERROR/
2293  005240  044107  020124  044123
2294  005246  043111  020124  051105
2295  005254  047522  000122
2296  005260  040515  020122  052504  EM10:   .ASCIZ  /MAR DUAL ADDRESSING ERROR/
2297  005266  046101  040440  042104
2298  005274  042522  051523  047111
2299  005302  020107  051105  047522
2300  005310  000122
2301  005312  052512  050115  043040  EM11:   .ASCIZ  /JUMP FIELD ERROR/
2302  005320  042511  042114  042440
2303  005326  051122  051117     000
2304  005333     112  046525  020120  EM12:   .ASCIZ  /JUMP TEST ERROR/
2305  005340  042524  052123  042440
2306  005346  051122  051117     000
2307  005353     103  047117  044504  EM16:   .ASCIZ  /CONDITION CODE TESTING,Z & C/
```

```
2308  005360  044524  047117  041440
2309  005366  042117  020105  042524
2310  005374  052123  047111  026107
2311  005402  020132  020046  000103
2312  005410  046103  041517  020113  EMB1:   .ASCIZ  /CLOCK TIME TOO FAST/
2313  005416  044524  042515  052040
2314  005424  047517  043040  051501
2315  005432  000124
2316  005434                          EM35:
2317  005434  047506  041522  020105  EM17:   .ASCIZ  /FORCE POWER FAIL ERROR/
2318  005442  047520  042527  020122
2319  005450  040506  046111  042440
2320  005456  051122  051117     000
2321  005463     111  052502  025123  EM27:   .ASCIZ  "IBUS* WRITE/READ ERROR"
2322  005470  053440  044522  042524
2323  005476  051057  040505  020104
2324  005504  051105  047522  000122
2325
2326  005512  041111  051525  047457  EM29:   .ASCIZ  'IBUS/OBUS WRITE/READ ERROR'
2327  005520  052502  020123  051127
2328  005526  052111  027505  042522
2329  005534  042101  042440  051122
2330  005542  051117     000
2331
2332  005545     120  046507  041440  EMB50:  .ASCIZ  'PGM CLOCK WOULD NOT CLEAR'
2333  005552  047514  045503  053440
2334  005560  052517  042114  047040
2335  005566  052117  041440  042514
2336  005574  051101     000
2337  005577     120  046507  041440  EMB51:  .ASCIZ  'PGM CLOCK WOULD NOT SET'
2338  005604  047514  045503  053440
2339  005612  052517  042114  047040
2340  005620  052117  051440  052105
2341  005626     000
2342  005627     045  022516  025101  STM:    .ASCIZ  '%N%A************************************************************'
2343  005634  025052  025052  025052
2344  005642  025052  025052  025052
2345  005650  025052  025052  025052
2346  005656  025052  025052  025052
2347  005664  025052  025052  025052
2348  005672  025052  025052  025052
2349  005700  025052  025052  025052
2350  005706  025052  025052  025052
2351  005714  025052  025052  025052
2352  005722     000
2353  005723     000                  DH0:    .ASCIZ  //
2354
2355
2356  005724  054105  042520  052103  DH1:    .ASCIZ  /EXPECTED  FOUND  ADDRESS/
2357  005732  042105  020040  047506
2358  005740  047125  020104  040440
2359  005746  042104  042522  051523
2360  005754     000
2361  005755     105  050130  041505  DH2:    .ASCIZ  /EXPECTED  FOUND/
2362  005762  042524  020104  043040
2363  005770  052517  042116     000
```

```
2364  005775     106  047522  020115  DH3:    .ASCIZ  /FROM ADDR  TO ADDR  BAD ADDR/
2365  006002  042101  051104  020040
2366  006010  047524  040440  042104
2367  006016  020122  041040  042101
2368  006024  040440  042104  000122
2369
2370
2371                                  .EVEN
2372
2373
2374
2375                                  ;-------------------------------------------------------------------
2376                                  ; MACRO'S NEEDED TO REPORT ERRORS
2377                                  ;-------------------------------------------------------------------
2378
2379                                  .MACRO  MDT1
2380                                          PRINTB  #TFM1,$REG2,$REG4,$REG0
2381                                  .ENDM
2382
2383                                  .MACRO  MDT2
2384                                          PRINTB  #TFM1,$REG5,$REG4,$REG2
2385                                  .ENDM
2386
2387                                  .MACRO  MDT3
2388                                          PRINTB  #TFM2,$REG5,$REG4
2389                                  .ENDM
2390
2391                                  .MACRO  MDT4
2392                                          PRINTB  #TFM3,$REG5,$REG4,FLAG
2393                                  .ENDM
2394
2395                                  .MACRO  MDT5
2396                                          PRINTB  #TFM3,$REG5,$REG4,$REG2
2397                                  .ENDM
2398
2399                                  .MACRO  MDT0
2400                                  .ENDM
2401                                  .MACRO  MDT6
2402                                          PRINTB  #TFM4,$REG2,$REG4
2403                                  .ENDM
2404
2405                                  .MACRO  MDT7
2406                                          PRINTB  #TFM4,$REG5,$REG4
```

```
2407                            .ENDM
2408                            .MACRO  MDT8
2409                                    PRINTB #TFM5,FADR,$REG5,$REG4
2410                            .ENDM
```

```
2411
2412                            .MACRO  $MD     ERRNN   ERNB    ERHM    ERFM
2413                                    BGNMSG  ERR'ERRNN
```

```
                                        PRINTB  #FM1,#EM'ERNB
                                        PRINTB  #FM1,#DH'ERHM
                                        MDT'ERFM
                                        PRINTB  #STM
                                        ENDMSG
                                .ENDM

                                .MACRO  ERROR   ECB
                                        JSR     PC,SV05
                                        ERRDF   'ECB',EM0,ERR'ECB'
                                .ENDM


006032                                  $MD     1,1,1,1
006032                          ERR1::
006032  012746  005056                  MOV     #EM1,-(SP)
006036  012746  005046                  MOV     #FM1,-(SP)
006042  012746  000002                  MOV     #2,-(SP)
006046  010600                          MOV     SP,R0
006050  104414                          TRAP    C$PNTB
006052  062706  000006                  ADD     #6,SP
006056  012746  005724                  MOV     #DH1,-(SP)
006062  012746  005046                  MOV     #FM1,-(SP)
006066  012746  000002                  MOV     #2,-(SP)
006072  010600                          MOV     SP,R0
006074  104414                          TRAP    C$PNTB
006076  062706  000006                  ADD     #6,SP
006102  013746  002430                  MOV     $REG0,-(SP)
006106  013746  002420                  MOV     $REG4,-(SP)
006112  013746  002424                  MOV     $REG2,-(SP)
006116  012746  003666                  MOV     #TFM1,-(SP)
006122  012746  000004                  MOV     #4,-(SP)
006126  010600                          MOV     SP,R0
006130  104414                          TRAP    C$PNTB
006132  062706  000012                  ADD     #12,SP
006136  012746  005627                  MOV     #STM,-(SP)
006142  012746  000001                  MOV     #1,-(SP)
006146  010600                          MOV     SP,R0
006150  104414                          TRAP    C$PNTB
006152  062706  000004                  ADD     #4,SP
006156                          L10003:
006156  104423                          TRAP    C$MSG
006160                                  $MD     2,2,1,1
006160                          ERR2::
006160  012746  005076                  MOV     #EM2,-(SP)
006164  012746  005046                  MOV     #FM1,-(SP)
006170  012746  000002                  MOV     #2,-(SP)
006174  010600                          MOV     SP,R0
006176  104414                          TRAP    C$PNTB
006200  062706  000006                  ADD     #6,SP
006204  012746  005724                  MOV     #DH1,-(SP)
006210  012746  005046                  MOV     #FM1,-(SP)
006214  012746  000002                  MOV     #2,-(SP)
006220  010600                          MOV     SP,R0
006222  104414                          TRAP    C$PNTB
```

```
2470  006224  062706  000006                ADD     #6,SP
2471  006230  013746  002430                MOV     $REG0,-(SP)
2472  006234  013746  002420                MOV     $REG4,-(SP)
2473  006240  013746  002424                MOV     $REG2,-(SP)
2474  006244  012746  003666                MOV     #TFM1,-(SP)
2475  006250  012746  000004                MOV     #4,-(SP)
2476  006254  010600                        MOV     SP,R0
2477  006256  104414                        TRAP    C$PNTB
2478  006260  062706  000012                ADD     #12,SP
2479  006264  012746  005627                MOV     #STM,-(SP)
2480  006270  012746  000001                MOV     #1,-(SP)
2481  006274  010600                        MOV     SP,R0
2482  006276  104414                        TRAP    C$PNTB
2483  006300  062706  000004                ADD     #4,SP
2484  006304                        L10004:
2485  006304  104423                        TRAP    C$MSG
2486  006306                                $MD     3,1,1,2
2487  006306                        ERR3::
2488  006306  012746  005056                MOV     #EM1,-(SP)
2489  006312  012746  005046                MOV     #FM1,-(SP)
2490  006316  012746  000002                MOV     #2,-(SP)
2491  006322  010600                        MOV     SP,R0
2492  006324  104414                        TRAP    C$PNTB
2493  006326  062706  000006                ADD     #6,SP
2494  006332  012746  005724                MOV     #DH1,-(SP)
2495  006336  012746  005046                MOV     #FM1,-(SP)
2496  006342  012746  000002                MOV     #2,-(SP)
2497  006346  010600                        MOV     SP,R0
2498  006350  104414                        TRAP    C$PNTB
2499  006352  062706  000006                ADD     #6,SP
2500  006356  013746  002424                MOV     $REG2,-(SP)
2501  006362  013746  002420                MOV     $REG4,-(SP)
2502  006366  013746  002416                MOV     $REG5,-(SP)
2503  006372  012746  003666                MOV     #TFM1,-(SP)
2504  006376  012746  000004                MOV     #4,-(SP)
2505  006402  010600                        MOV     SP,R0
2506  006404  104414                        TRAP    C$PNTB
2507  006406  062706  000012                ADD     #12,SP
2508  006412  012746  005627                MOV     #STM,-(SP)
2509  006416  012746  000001                MOV     #1,-(SP)
2510  006422  010600                        MOV     SP,R0
2511  006424  104414                        TRAP    C$PNTB
2512  006426  062706  000004                ADD     #4,SP
2513  006432                        L10005:
2514  006432  104423                        TRAP    C$MSG
2515  006434                                $MD     4,3,2,3
2516  006434                        ERR4::
2517  006434  012746  005131                MOV     #EM3,-(SP)
2518  006440  012746  005046                MOV     #FM1,-(SP)
2519  006444  012746  000002                MOV     #2,-(SP)
2520  006450  010600                        MOV     SP,R0
2521  006452  104414                        TRAP    C$PNTB
2522  006454  062706  000006                ADD     #6,SP
2523  006460  012746  005755                MOV     #DH2,-(SP)
2524  006464  012746  005046                MOV     #FM1,-(SP)
2525  006470  012746  000002                MOV     #2,-(SP)
```

```
2526  006474  010600                        MOV     SP,R0
2527  006476  104414                        TRAP    C$PNTB
2528  006500  062706  000006                ADD     #6,SP
2529  006504  013746  002420                MOV     $REG4,-(SP)
2530  006510  013746  002416                MOV     $REG5,-(SP)
2531  006514  012746  003712                MOV     #TFM2,-(SP)
2532  006520  012746  000003                MOV     #3,-(SP)
2533  006524  010600                        MOV     SP,R0
2534  006526  104414                        TRAP    C$PNTB
2535  006530  062706  000010                ADD     #10,SP
2536  006534  012746  005627                MOV     #STM,-(SP)
2537  006540  012746  000001                MOV     #1,-(SP)
2538  006544  010600                        MOV     SP,R0
2539  006546  104414                        TRAP    C$PNTB
2540  006550  062706  000004                ADD     #4,SP
2541  006554                        L10006:
2542  006554  104423                        TRAP    C$MSG
```

```
2543  006556                                    $MD     5,4,2,3
2544  006556                              ERR5::
2545  006556  012746  005144                    MOV     #EM4,-(SP)
2546  006562  012746  005046                    MOV     #FM1,-(SP)
2547  006566  012746  000002                    MOV     #2,-(SP)
2548  006572  010600                            MOV     SP,R0
2549  006574  104414                            TRAP    C$PNTB
2550  006576  062706  000006                    ADD     #6,SP
2551  006602  012746  005755                    MOV     #DH2,-(SP)
2552  006606  012746  005046                    MOV     #FM1,-(SP)
2553  006612  012746  000002                    MOV     #2,-(SP)
2554  006616  010600                            MOV     SP,R0
2555  006620  104414                            TRAP    C$PNTB
2556  006622  062706  000006                    ADD     #6,SP
2557  006626  013746  002420                    MOV     $REG4,-(SP)
2558  006632  013746  002416                    MOV     $REG5,-(SP)
2559  006636  012746  003712                    MOV     #TFM2,-(SP)
2560  006642  012746  000003                    MOV     #3,-(SP)
2561  006646  010600                            MOV     SP,R0
2562  006650  104414                            TRAP    C$PNTB
2563  006652  062706  000010                    ADD     #10,SP
2564  006656  012746  005627                    MOV     #STM,-(SP)
2565  006662  012746  000001                    MOV     #1,-(SP)
2566  006666  010600                            MOV     SP,R0
2567  006670  104414                            TRAP    C$PNTB
2568  006672  062706  000004                    ADD     #4,SP
2569  006676                              L10007:
2570  006676  104423                            TRAP    C$MSG
```

```
                                                        $MD     6,5,1,4
                                                ERR6::
                                                        MOV     #EM5,-(SP)
                                                        MOV     #FM1,-(SP)
                                                        MOV     #2,-(SP)
                                                        MOV     SP,R0
                                                        TRAP    C$PNTB
                                                        ADD     #6,SP
                                                        MOV     #DH1,-(SP)
                                                        MOV     #FM1,-(SP)
                                                        MOV     #2,-(SP)
                                                        MOV     SP,R0
                                                        TRAP    C$PNTB
                                                        ADD     #6,SP
                                                        MOV     FLAG,-(SP)
                                                        MOV     $REG4,-(SP)
                                                        MOV     $REG5,-(SP)
                                                        MOV     #TFM3,-(SP)
                                                        MOV     #4,-(SP)
                                                        MOV     SP,R0
                                                        TRAP    C$PNTB
                                                        ADD     #12,SP
                                                        MOV     #STM,-(SP)
                                                        MOV     #1,-(SP)
                                                        MOV     SP,R0
                                                        TRAP    C$PNTB
                                                        ADD     #4,SP
                                                L10010:
                                                        TRAP    C$MSG
                                                        $MD     7,6,1,5
                                                ERR7::
                                                        MOV     #EM6,-(SP)
                                                        MOV     #FM1,-(SP)
                                                        MOV     #2,-(SP)
                                                        MOV     SP,R0
                                                        TRAP    C$PNTB
                                                        ADD     #6,SP
                                                        MOV     #DH1,-(SP)
                                                        MOV     #FM1,-(SP)
                                                        MOV     #2,-(SP)
                                                        MOV     SP,R0
                                                        TRAP    C$PNTB
                                                        ADD     #6,SP
                                                        MOV     $REG2,-(SP)
                                                        MOV     $REG4,-(SP)
                                                        MOV     $REG5,-(SP)
                                                        MOV     #TFM3,-(SP)
                                                        MOV     #4,-(SP)
                                                        MOV     SP,R0
                                                        TRAP    C$PNTB
                                                        ADD     #12,SP
                                                        MOV     #STM,-(SP)
                                                        MOV     #1,-(SP)
                                                        MOV     SP,R0
                                                        TRAP    C$PNTB
                                                        ADD     #4,SP
```

```
2627  007152
2628  007152  104423                  L10011:
2629  007154                                  TRAP    C$MSG
2630  007154                                  $MD     10,7,2,3
2631  007154  012746  005233          ERR10::
2632  007160  012746  005046                  MOV     #EM7,-(SP)
2633  007164  012746  000002                  MOV     #FM1,-(SP)
2634  007170  010600                          MOV     #2,-(SP)
2635  007172  104414                          MOV     SP,R0
2636  007174  062706  000006                  TRAP    C$PNTB
2637  007200  012746  005755                  ADD     #6,SP
2638  007204  012746  005046                  MOV     #DH2,-(SP)
2639  007210  012746  000002                  MOV     #FM1,-(SP)
2640  007214  010600                          MOV     #2,-(SP)
2641  007216  104414                          MOV     SP,R0
2642  007220  062706  000006                  TRAP    C$PNTB
2643  007224  013746  002420                  ADD     #6,SP
2644  007230  013746  002416                  MOV     $REG4,-(SP)
2645  007234  012746  003712                  MOV     $REG5,-(SP)
2646  007240  012746  000003                  MOV     #TFM2,-(SP)
2647  007244  010600                          MOV     #3,-(SP)
2648  007246  104414                          MOV     SP,R0
2649  007250  062706  000010                  TRAP    C$PNTB
2650  007254  012746  005627                  ADD     #10,SP
2651  007260  012746  000001                  MOV     #STM,-(SP)
2652  007264  010600                          MOV     #1,-(SP)
2653  007266  104414                          MOV     SP,R0
2654  007270  062706  000004                  TRAP    C$PNTB
2655  007274                                  ADD     #4,SP
2656  007274  104423                  L10012:
2657  007276                                  TRAP    C$MSG
2658  007276                                  $MD     11,10,2,6
2659  007276  012746  005260          ERR11::
2660  007302  012746  005046                  MOV     #EM10,-(SP)
2661  007306  012746  000002                  MOV     #FM1,-(SP)
2662  007312  010600                          MOV     #2,-(SP)
2663  007314  104414                          MOV     SP,R0
2664  007316  062706  000006                  TRAP    C$PNTB
2665  007322  012746  005755                  ADD     #6,SP
2666  007326  012746  005046                  MOV     #DH2,-(SP)
2667  007332  012746  000002                  MOV     #FM1,-(SP)
2668  007336  010600                          MOV     #2,-(SP)
2669  007340  104414                          MOV     SP,R0
2670  007342  062706  000006                  TRAP    C$PNTB
2671  007346  013746  002420                  ADD     #6,SP
2672  007352  013746  002424                  MOV     $REG4,-(SP)
2673  007356  012746  003755                  MOV     $REG2,-(SP)
2674  007362  012746  000003                  MOV     #TFM4,-(SP)
2675  007366  010600                          MOV     #3,-(SP)
2676  007370  104414                          MOV     SP,R0
2677  007372  062706  000010                  TRAP    C$PNTB
2678  007376  012746  005627                  ADD     #10,SP
2679  007402  012746  000001                  MOV     #STM,-(SP)
2680  007406  010600                          MOV     #1,-(SP)
2681  007410  104414                          MOV     SP,R0
2682  007412  062706  000004                  TRAP    C$PNTB
                                            ADD     #4,SP
```

```
2683  007416                          L10013:
2684  007416  104423                          TRAP    C$MSG
2685  007420                                  $MD     12,7,2,7
2686  007420                          ERR12::
2687  007420  012746  005233                  MOV     #EM7,-(SP)
2688  007424  012746  005046                  MOV     #FM1,-(SP)
2689  007430  012746  000002                  MOV     #2,-(SP)
2690  007434  010600                          MOV     SP,R0
2691  007436  104414                          TRAP    C$PNTB
2692  007440  062706  000006                  ADD     #6,SP
2693  007444  012746  005755                  MOV     #DH2,-(SP)
2694  007450  012746  005046                  MOV     #FM1,-(SP)
2695  007454  012746  000002                  MOV     #2,-(SP)
2696  007460  010600                          MOV     SP,R0
2697  007462  104414                          TRAP    C$PNTB
2698  007464  062706  000006                  ADD     #6,SP
2699  007470  013746  002420                  MOV     $REG4,-(SP)
2700  007474  013746  002416                  MOV     $REG5,-(SP)
2701  007500  012746  003755                  MOV     #TFM4,-(SP)
2702  007504  012746  000003                  MOV     #3,-(SP)
2703  007510  010600                          MOV     SP,R0
2704  007512  104414                          TRAP    C$PNTB
2705  007514  062706  000010                  ADD     #10,SP
2706  007520  012746  005627                  MOV     #STM,-(SP)
2707  007524  012746  000001                  MOV     #1,-(SP)
2708  007530  010600                          MOV     SP,R0
2709  007532  104414                          TRAP    C$PNTB
2710  007534  062706  000004                  ADD     #4,SP
2711  007540                          L10014:
2712  007540  104423                          TRAP    C$MSG
2713  007542                                  $MD     13,10,2,3
2714  007542                          ERR13::
2715  007542  012746  005260                  MOV     #EM10,-(SP)
2716  007546  012746  005046                  MOV     #FM1,-(SP)
2717  007552  012746  000002                  MOV     #2,-(SP)
2718  007556  010600                          MOV     SP,R0
2719  007560  104414                          TRAP    C$PNTB
2720  007562  062706  000006                  ADD     #6,SP
2721  007566  012746  005755                  MOV     #DH2,-(SP)
2722  007572  012746  005046                  MOV     #FM1,-(SP)
2723  007576  012746  000002                  MOV     #2,-(SP)
2724  007602  010600                          MOV     SP,R0
2725  007604  104414                          TRAP    C$PNTB
2726  007606  062706  000006                  ADD     #6,SP
2727  007612  013746  002420                  MOV     $REG4,-(SP)
2728  007616  013746  002416                  MOV     $REG5,-(SP)
2729  007622  012746  003712                  MOV     #TFM2,-(SP)
2730  007626  012746  000003                  MOV     #3,-(SP)
2731  007632  010600                          MOV     SP,R0
2732  007634  104414                          TRAP    C$PNTB
2733  007636  062706  000010                  ADD     #10,SP
2734  007642  012746  005627                  MOV     #STM,-(SP)
2735  007646  012746  000001                  MOV     #1,-(SP)
2736  007652  010600                          MOV     SP,R0
2737  007654  104414                          TRAP    C$PNTB
2738  007656  062706  000004                  ADD     #4,SP
```

```
2739  007662
2740  007662  104423                  L10015:
2741  007664                                  TRAP    C$MSG
2742  007664                                  $MD     14,11,2,6
2743  007664  012746  005312          ERR14::
2744  007670  012746  005046                  MOV     #EM11,-(SP)
2745  007674  012746  000002                  MOV     #FM1,-(SP)
2746  007700  010600                          MOV     #2,-(SP)
2747  007702  104414                          MOV     SP,R0
2748  007704  062706  000006                  TRAP    C$PNTB
2749  007710  012746  005755                  ADD     #6,SP
2750  007714  012746  005046                  MOV     #DH2,-(SP)
2751  007720  012746  000002                  MOV     #FM1,-(SP)
2752  007724  010600                          MOV     #2,-(SP)
2753  007726  104414                          MOV     SP,R0
2754  007730  062706  000006                  TRAP    C$PNTB
2755  007734  013746  002420                  ADD     #6,SP
2756  007740  013746  002424                  MOV     $REG4,-(SP)
2757  007744  012746  003755                  MOV     $REG2,-(SP)
2758  007750  012746  000003                  MOV     #TFM4,-(SP)
2759  007754  010600                          MOV     #3,-(SP)
2760  007756  104414                          MOV     SP,R0
2761  007760  062706  000010                  TRAP    C$PNTB
2762  007764  012746  005627                  ADD     #10,SP
2763  007770  012746  000001                  MOV     #STM,-(SP)
2764  007774  010600                          MOV     #1,-(SP)
2765  007776  104414                          MOV     SP,R0
2766  010000  062706  000004                  TRAP    C$PNTB
2767  010004                                  ADD     #4,SP
2768  010004  104423                  L10016:
                                            TRAP    C$MSG
```

```
2769  010006                          $MD     15,12,3,8
2770  010006                  ERR15::
2771  010006  012746  005333          MOV     #EM12,-(SP)
2772  010012  012746  005046          MOV     #FM1,-(SP)
2773  010016  012746  000002          MOV     #2,-(SP)
2774  010022  010600                  MOV     SP,R0
2775  010024  104414                  TRAP    C$PNTB
2776  010026  062706  000006          ADD     #6,SP
2777  010032  012746  005775          MOV     #DH3,-(SP)
2778  010036  012746  005046          MOV     #FM1,-(SP)
2779  010042  012746  000002          MOV     #2,-(SP)
2780  010046  010600                  MOV     SP,R0
2781  010050  104414                  TRAP    C$PNTB
2782  010052  062706  000006          ADD     #6,SP
2783  010056  013746  002420          MOV     $REG4,-(SP)
2784  010062  013746  002416          MOV     $REG5,-(SP)
2785  010066  013746  002412          MOV     FADR,-(SP)
2786  010072  012746  003773          MOV     #TFM5,-(SP)
2787  010076  012746  000004          MOV     #4,-(SP)
2788  010102  010600                  MOV     SP,R0
2789  010104  104414                  TRAP    C$PNTB
2790  010106  062706  000012          ADD     #12,SP
2791  010112  012746  005627          MOV     #STM,-(SP)
2792  010116  012746  000001          MOV     #1,-(SP)
2793  010122  010600                  MOV     SP,R0
2794  010124  104414                  TRAP    C$PNTB
2795  010126  062706  000004          ADD     #4,SP
2796  010132                  L10017:
2797  010132  104423                  TRAP    C$MSG
2798  010134                          $MD     16,16,2,7
2799  010134                  ERR16::
2800  010134  012746  005353          MOV     #EM16,-(SP)
2801  010140  012746  005046          MOV     #FM1,-(SP)
2802  010144  012746  000002          MOV     #2,-(SP)
2803  010150  010600                  MOV     SP,R0
2804  010152  104414                  TRAP    C$PNTB
2805  010154  062706  000006          ADD     #6,SP
2806  010160  012746  005755          MOV     #DH2,-(SP)
2807  010164  012746  005046          MOV     #FM1,-(SP)
2808  010170  012746  000002          MOV     #2,-(SP)
2809  010174  010600                  MOV     SP,R0
2810  010176  104414                  TRAP    C$PNTB
2811  010200  062706  000006          ADD     #6,SP
2812  010204  013746  002420          MOV     $REG4,-(SP)
2813  010210  013746  002416          MOV     $REG5,-(SP)
2814  010214  012746  003755          MOV     #TFM4,-(SP)
2815  010220  012746  000003          MOV     #3,-(SP)
2816  010224  010600                  MOV     SP,R0
2817  010226  104414                  TRAP    C$PNTB
2818  010230  062706  000010          ADD     #10,SP
2819  010234  012746  005627          MOV     #STM,-(SP)
2820  010240  012746  000001          MOV     #1,-(SP)
2821  010244  010600                  MOV     SP,R0
2822  010246  104414                  TRAP    C$PNTB
2823  010250  062706  000004          ADD     #4,SP
2824  010254                  L10020:
```

```
 2825  010254  104423                    TRAP    C$MSG
 2826
```

```
2827
2828  010256                                    $MD     17,17,0,0
2829  010256                              ERR17::
2830  010256  012746  005434                    MOV     #EM17,-(SP)
2831  010262  012746  005046                    MOV     #FM1,-(SP)
2832  010266  012746  000002                    MOV     #2,-(SP)
2833  010272  010600                            MOV     SP,R0
2834  010274  104414                            TRAP    C$PNTB
2835  010276  062706  000006                    ADD     #6,SP
2836  010302  012746  005723                    MOV     #DH0,-(SP)
2837  010306  012746  005046                    MOV     #FM1,-(SP)
2838  010312  012746  000002                    MOV     #2,-(SP)
2839  010316  010600                            MOV     SP,R0
2840  010320  104414                            TRAP    C$PNTB
2841  010322  062706  000006                    ADD     #6,SP
2842  010326  012746  005627                    MOV     #STM,-(SP)
2843  010332  012746  000001                    MOV     #1,-(SP)
2844  010336  010600                            MOV     SP,R0
2845  010340  104414                            TRAP    C$PNTB
2846  010342  062706  000004                    ADD     #4,SP
2847  010346                              L10021:
2848  010346  104423                            TRAP    C$MSG
2849  010350                                    $MD     29,29,2,3
2850  010350                              ERR29::
2851  010350  012746  005512                    MOV     #EM29,-(SP)
2852  010354  012746  005046                    MOV     #FM1,-(SP)
2853  010360  012746  000002                    MOV     #2,-(SP)
2854  010364  010600                            MOV     SP,R0
2855  010366  104414                            TRAP    C$PNTB
2856  010370  062706  000006                    ADD     #6,SP
2857  010374  012746  005755                    MOV     #DH2,-(SP)
2858  010400  012746  005046                    MOV     #FM1,-(SP)
2859  010404  012746  000002                    MOV     #2,-(SP)
2860  010410  010600                            MOV     SP,R0
2861  010412  104414                            TRAP    C$PNTB
2862  010414  062706  000006                    ADD     #6,SP
2863  010420  013746  002420                    MOV     $REG4,-(SP)
2864  010424  013746  002416                    MOV     $REG5,-(SP)
2865  010430  012746  003712                    MOV     #TFM2,-(SP)
2866  010434  012746  000003                    MOV     #3,-(SP)
2867  010440  010600                            MOV     SP,R0
2868  010442  104414                            TRAP    C$PNTB
2869  010444  062706  000010                    ADD     #10,SP
2870  010450  012746  005627                    MOV     #STM,-(SP)
2871  010454  012746  000001                    MOV     #1,-(SP)
2872  010460  010600                            MOV     SP,R0
2873  010462  104414                            TRAP    C$PNTB
2874  010464  062706  000004                    ADD     #4,SP
2875  010470                              L10022:
2876  010470  104423                            TRAP    C$MSG
2877  010472                                    $MD     35,35,2,3
2878  010472                              ERR35::
2879  010472  012746  005434                    MOV     #EM35,-(SP)
2880  010476  012746  005046                    MOV     #FM1,-(SP)
2881  010502  012746  000002                    MOV     #2,-(SP)
2882  010506  010600                            MOV     SP,R0
```

```
2883  010510  104414                          TRAP    C$PNTB
2884  010512  062706  000006                  ADD     #6,SP
2885  010516  012746  005755                  MOV     #DH2,-(SP)
2886  010522  012746  005046                  MOV     #FM1,-(SP)
2887  010526  012746  000002                  MOV     #2,-(SP)
2888  010532  010600                          MOV     SP,R0
2889  010534  104414                          TRAP    C$PNTB
2890  010536  062706  000006                  ADD     #6,SP
2891  010542  013746  002420                  MOV     $REG4,-(SP)
2892  010546  013746  002416                  MOV     $REG5,-(SP)
2893  010552  012746  003712                  MOV     #TFM2,-(SP)
2894  010556  012746  000003                  MOV     #3,-(SP)
2895  010562  010600                          MOV     SP,R0
2896  010564  104414                          TRAP    C$PNTB
2897  010566  062706  000010                  ADD     #10,SP
2898  010572  012746  005627                  MOV     #STM,-(SP)
2899  010576  012746  000001                  MOV     #1,-(SP)
2900  010602  010600                          MOV     SP,R0
2901  010604  104414                          TRAP    C$PNTB
2902  010606  062706  000004                  ADD     #4,SP
2903  010612                          L10023:
2904  010612  104423                          TRAP    C$MSG
2905
2906  010614                          BGNMSG  ERR36
2907  010614                          ERR36::
2908  010614                          PRINTB  #STM
2909  010614  012746  005627                  MOV     #STM,-(SP)
2910  010620  012746  000001                  MOV     #1,-(SP)
2911  010624  010600                          MOV     SP,R0
2912  010626  104414                          TRAP    C$PNTB
2913  010630  062706  000004                  ADD     #4,SP
2914  010634                          ENDMSG
2915  010634                          L10024:
2916  010634  104423                          TRAP    C$MSG
2917
2918  010636                          BGNMSG  ERR40
2919  010636                          ERR40::
2920  010636                          PRINTF  #TFM40,R2
2921  010636  010246                          MOV     R2,-(SP)
2922  010640  012746  004263                  MOV     #TFM40,-(SP)
2923  010644  012746  000002                  MOV     #2,-(SP)
2924  010650  010600                          MOV     SP,R0
2925  010652  104417                          TRAP    C$PNTF
2926  010654  062706  000006                  ADD     #6,SP
2927  010660                          PRINTB  #STM
2928  010660  012746  005627                  MOV     #STM,-(SP)
2929  010664  012746  000001                  MOV     #1,-(SP)
2930  010670  010600                          MOV     SP,R0
2931  010672  104414                          TRAP    C$PNTB
2932  010674  062706  000004                  ADD     #4,SP
2933  010700                          ENDMSG
2934  010700                          L10025:
2935  010700  104423                          TRAP    C$MSG
2936  010702                          BGNMSG  ERR41
2937  010702                          ERR41::
2938  010702                          PRINTF  #TFM41
```

```
2939  010702  012746  004105                    MOV     #TFM41,-(SP)
2940  010706  012746  000001                    MOV     #1,-(SP)
2941  010712  010600                           MOV     SP,R0
2942  010714  104417                           TRAP    C$PNTF
2943  010716  062706  000004                   ADD     #4,SP
2944  010722                            PRINTB  #STM
2945  010722  012746  005627                   MOV     #STM,-(SP)
2946  010726  012746  000001                   MOV     #1,-(SP)
2947  010732  010600                           MOV     SP,R0
2948  010734  104414                           TRAP    C$PNTB
2949  010736  062706  000004                   ADD     #4,SP
2950  010742                            ENDMSG
2951  010742                            L10026:
2952  010742  104423                           TRAP    C$MSG
2953  010744                            BGNMSG  ERR42
2954  010744                            ERR42::
2955  010744                            PRINTF  #TFM42
2956  010744  012746  004174                   MOV     #TFM42,-(SP)
2957  010750  012746  000001                   MOV     #1,-(SP)
2958  010754  010600                           MOV     SP,R0
2959  010756  104417                           TRAP    C$PNTF
2960  010760  062706  000004                   ADD     #4,SP
2961  010764                            PRINTB  #STM
2962  010764  012746  005627                   MOV     #STM,-(SP)
2963  010770  012746  000001                   MOV     #1,-(SP)
2964  010774  010600                           MOV     SP,R0
2965  010776  104414                           TRAP    C$PNTB
2966  011000  062706  000004                   ADD     #4,SP
2967  011004                            ENDMSG
2968  011004                            L10027:
2969  011004  104423                           TRAP    C$MSG
2970
2971  011006                            BGNMSG  ERR43
2972  011006                            ERR43::
2973  011006                            PRINTF  #TFM43,R5,R4
2974  011006  010446                           MOV     R4,-(SP)
2975  011010  010546                           MOV     R5,-(SP)
2976  011012  012746  004344                   MOV     #TFM43,-(SP)
2977  011016  012746  000003                   MOV     #3,-(SP)
2978  011022  010600                           MOV     SP,R0
2979  011024  104417                           TRAP    C$PNTF
2980  011026  062706  000010                   ADD     #10,SP
2981  011032                            PRINTB  #STM
2982  011032  012746  005627                   MOV     #STM,-(SP)
2983  011036  012746  000001                   MOV     #1,-(SP)
2984  011042  010600                           MOV     SP,R0
2985  011044  104414                           TRAP    C$PNTB
2986  011046  062706  000004                   ADD     #4,SP
2987  011052                            ENDMSG
2988  011052                            L10030:
2989  011052  104423                           TRAP    C$MSG
2990  011054                            BGNMSG  ERR44
2991  011054                            ERR44::
2992  011054                            PRINTF  #TFM44,#TMMC,R4
2993  011054  010446                           MOV     R4,-(SP)
2994  011056  012746  005010                   MOV     #TMMC,-(SP)
```

```
 2995  011062  012746  004427                    MOV     #TFM44,-(SP)
 2996  011066  012746  000003                    MOV     #3,-(SP)
 2997  011072  010600                            MOV     SP,R0
 2998  011074  104417                            TRAP    C$PNTF
 2999  011076  062706  000010                    ADD     #10,SP
 3000  011102                          PRINTB  #STM
 3001  011102  012746  005627                    MOV     #STM,-(SP)
 3002  011106  012746  000001                    MOV     #1,-(SP)
 3003  011112  010600                            MOV     SP,R0
 3004  011114  104414                            TRAP    C$PNTB
 3005  011116  062706  000004                    ADD     #4,SP
 3006  011122                          ENDMSG
 3007  011122                          L10031:
 3008  011122  104423                            TRAP    C$MSG
 3009  011124                          BGNMSG  ERR45
 3010  011124                          ERR45::
 3011  011124                          PRINTF  #TFM45,#TMMC,R4
 3012  011124  010446                            MOV     R4,-(SP)
 3013  011126  012746  005010                    MOV     #TMMC,-(SP)
 3014  011132  012746  004466                    MOV     #TFM45,-(SP)
 3015  011136  012746  000003                    MOV     #3,-(SP)
 3016  011142  010600                            MOV     SP,R0
 3017  011144  104417                            TRAP    C$PNTF
 3018  011146  062706  000010                    ADD     #10,SP
 3019  011152                          PRINTB  #TFM45A
 3020  011152  012746  004524                    MOV     #TFM45A,-(SP)
 3021  011156  012746  000001                    MOV     #1,-(SP)
 3022  011162  010600                            MOV     SP,R0
 3023  011164  104414                            TRAP    C$PNTB
 3024  011166  062706  000004                    ADD     #4,SP
 3025  011172                          PRINTB  #STM
 3026  011172  012746  005627                    MOV     #STM,-(SP)
 3027  011176  012746  000001                    MOV     #1,-(SP)
 3028  011202  010600                            MOV     SP,R0
 3029  011204  104414                            TRAP    C$PNTB
 3030  011206  062706  000004                    ADD     #4,SP
 3031  011212                          ENDMSG
 3032  011212                          L10032:
 3033  011212  104423                            TRAP    C$MSG
 3034  011214                          BGNMSG  ERR46
 3035  011214                          ERR46::
 3036  011214                          PRINTF  #TFM46,$GDDAT,R4
 3037  011214  010446                            MOV     R4,-(SP)
 3038  011216  013746  002452                    MOV     $GDDAT,-(SP)
 3039  011222  012746  004647                    MOV     #TFM46,-(SP)
 3040  011226  012746  000003                    MOV     #3,-(SP)
 3041  011232  010600                            MOV     SP,R0
 3042  011234  104417                            TRAP    C$PNTF
 3043  011236  062706  000010                    ADD     #10,SP
 3044  011242                          PRINTB  #STM
 3045  011242  012746  005627                    MOV     #STM,-(SP)
 3046  011246  012746  000001                    MOV     #1,-(SP)
 3047  011252  010600                            MOV     SP,R0
 3048  011254  104414                            TRAP    C$PNTB
 3049  011256  062706  000004                    ADD     #4,SP
 3050  011262                          ENDMSG
```

```
3051  011262                        L10033:
3052  011262  104423                      TRAP    C$MSG
3053
3054  011264                        BGNMSG  ERR47
3055  011264                        ERR47::
3056  011264                        PRINTF  #TFM47,R5,R4
3057  011264  010446                      MOV     R4,-(SP)
3058  011266  010546                      MOV     R5,-(SP)
3059  011270  012746  004733              MOV     #TFM47,-(SP)
3060  011274  012746  000003              MOV     #3,-(SP)
3061  011300  010600                      MOV     SP,R0
3062  011302  104417                      TRAP    C$PNTF
3063  011304  062706  000010              ADD     #10,SP
3064  011310                        PRINTB  #STM
3065  011310  012746  005627              MOV     #STM,-(SP)
3066  011314  012746  000001              MOV     #1,-(SP)
3067  011320  010600                      MOV     SP,R0
3068  011322  104414                      TRAP    C$PNTB
3069  011324  062706  000004              ADD     #4,SP
3070  011330                        ENDMSG
3071  011330                        L10034:
3072  011330  104423                      TRAP    C$MSG
3073
```

```
3074
3075
3076
                        .SBTTL  REPORT CODING SECTION
```

```
3077                                    ;++
```

```
3078                                    ; THE REPORT CODING SECTION CONTAINS THE
3079                                    ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
3080                                    ;--
3081
3082  011332                                    BGNRPT
3083  011332                            L$RPT::
3084
3085
3086  011332                                    EXIT    RPT
3087  011332  000167                            .WORD   J$JMP
3088  011334  000000                            .WORD   L10035-2-.
3089
```

```
 3090
 3091  011336                          ENDRPT
 3092  011336                  L10035:
 3093  011336  104425                  TRAP    C$RPT
 3094
```

```
3095
3096
```

```
3097
3098
```

```
                                .SBTTL  INITIALIZE SECTION
```

```
3100
3101                                    ;////////////////////////////////////////////////////////////////////////
3102                                    ;/ THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
3103                                    ;/ AT THE BEGINNING OF EACH PASS.
3104                                    ;////////////////////////////////////////////////////////////////////////
3105
3106  011340                                    BGNINIT
3107  011340                            L$INIT::
3108
3109                                    ;INITIALIZE SUBROUTINE STACK
3110  011340  012705  002730                    MOV     #SSTACK,R5
3111                                    ;STORE BASE LEVEL PROGRAM STACK POINTER
3112  011344  010637  002344                    MOV     SP,PSTACK
3113  011350  005737  002462                    TST     FTIME
3114  011354  001011                            BNE     1$
3115  011356  013737  000004  002464           MOV     @#4,SAVE4
3116  011364  013737  000006  002466           MOV     @#6,SAVE6
3117  011372  012737  000001  002462           MOV     #1,FTIME
3118  011400  013737  002464  000004  1$:      MOV     SAVE4,@#4
3119  011406  013737  002466  000006           MOV     SAVE6,@#6
3120
3121                                    ;SEE IF PROGRAM JUST STARTED, BR IF YES
3122  011414                                    READEF  #EF.START
3123  011414  012700  000040                    MOV     #EF.START,R0
3124  011420  104447                            TRAP    C$REFG
3125  011422                                    BCOMPLETE       NEWST
3126  011422  103414                            BCS     NEWST
3127                                    ;SEE IF THIS IS A NEW PASS, BR IF YES
3128  011424                                    READEF  #EF.NEW
3129  011424  012700  000035                    MOV     #EF.NEW,R0
3130  011430  104447                            TRAP    C$REFG
3131  011432                                    BCOMPLETE       NEWST
3132  011432  103410                            BCS     NEWST
3133                                    ;SEE IF PROGRAM WAS JUST CONTINUED
3134  011434                                    READEF  #EF.CONTINUE
3135  011434  012700  000036                    MOV     #EF.CONTINUE,R0
3136  011440  104447                            TRAP    C$REFG
3137  011442                                    BCOMPLETE       ENDIT
3138  011442  103576                            BCS     ENDIT
3139                                    ;SEE IF PROGRAM JUST RESTARTED, BR IF NOT
3140  011444                                    READEF  #EF.RESTART
3141  011444  012700  000037                    MOV     #EF.RESTART,R0
3142  011450  104447                            TRAP    C$REFG
3143  011452                                    BNCOMPLETE      GETPRM
3144  011452  103003                            BCC     GETPRM
3145
3146  011454                            NEWST:
3147                                    ;RESET LOGICAL DEVICE TO -1
3148  011454  012737  177777  002342            MOV     #-1,LOGDEV
3149
3150                                    ;GET UNIBUS ADRS, VECTOR, PRIORITY LEVEL, LINE UNIT,SWITCH
3151                                    ;    PACKS, TEST CONNECTOR INFO. FOR THIS M8200,4,7 (CURRENT LOGICAL
3152                                    ;    DEVICE).
3153  011462                            GETPRM:
3154  011462  005237  002342                    INC     LOGDEV
3155  011466  023737  002342  002012            CMP     LOGDEV,L$UNIT
```

```
3156  011474  002367                          BGE     NEWST
3157  011476                                  GPHARD  LOGDEV,R1
3158  011476  013700  002342                  MOV     LOGDEV,R0
3159  011502  104442                          TRAP    C$GPHRD
3160  011504  010001                          MOV     R0,R1
3161  011506                                  BNCOMPLETE      GETPRM
3162  011506  103365                          BCC     GETPRM
3163  011510  012137  002414                  MOV     (R1)+,WTYPE
3164                                  ;GET ADDRESS OF M8200,4,7
3165  011514  011137  002516                  MOV     (R1),KMCSR
3166                                  ;GET POINTER TO M8200,4,7 CSR HI BYTE
3167  011520  011137  002520                  MOV     (R1),KMCSRH
3168  011524  005237  002520                  INC     KMCSRH
3169                                  ;GET POINTER TO M8200,4,7 CTL OUT REG
3170  011530  011137  002522                  MOV     (R1),KMCTL
3171  011534  062737  000002  002522          ADD     #2,KMCTL
3172                                  ;GET POINTER TO M8200,4,7 PORT REG - SEL 4
3173  011542  011137  002524                  MOV     (R1),KMPO4
3174  011546  062737  000004  002524          ADD     #4,KMPO4
3175                                  ;GET POINTER TO M8200,4,7 PORT REG - SEL 6
3176  011554  012137  002526                  MOV     (R1)+,KMPO6
3177  011560  062737  000006  002526          ADD     #6,KMPO6
3178                                  ;GET POINTER TO RCV VECTOR
3179  011566  011137  002506                  MOV     (R1),KMRVEC
3180                                  ;GET POINTER TO RCV PRIORITY LEVEL
3181  011572  011137  002510                  MOV     (R1),KMRLVL
3182  011576  062737  000002  002510          ADD     #2,KMRLVL
3183                                  ;GET POINTER TO TX VECTOR
3184  011604  011137  002512                  MOV     (R1),KMTVEC
3185  011610  062737  000004  002512          ADD     #4,KMTVEC
3186                                  ;GET POINTER TO TX PRIORITY LEVEL
3187  011616  011137  002514                  MOV     (R1),KMTLVL
3188  011622  062737  000006  002514          ADD     #6,KMTLVL
3189                                  ;PUT VECTOR INTO STAT1
3190  011630  016137  000020  002472          MOV     20(R1),RUNINH
3191  011636  012137  002500                  MOV     (R1)+,STAT1
3192                                  ;PUT PRIORITY INTO STAT1
3193  011642  052137  002500                  BIS     (R1)+,STAT1
3194                                  ;SEE IF NO LINE UNIT, SET BIT IF YES
3195  011646  005711                          TST     (R1)
3196  011650  001004                          BNE     50000$
3197  011652  052737  010000  002500          BIS     #BIT12,STAT1
3198  011660  000416                          BR      4$
3199  011662                                  50000$:
3200                                  ;SEE IF M8201 LINE UNIT, SET BIT IF YES
3201  011662  021127  000001                  CMP     (R1),#1
3202  011666  001001                          BNE     50001$
3203  011670  000412                          BR      4$
3204  011672                                  50001$:
3205                                  ;SEE IF M8202 LINE UNIT, SET BIT IF YES
3206  011672  021127  000002                  CMP     (R1),#2
3207  011676  001004                          BNE     50002$
3208  011700  052737  020000  002500          BIS     #BIT13,STAT1
3209  011706  000403                          BR      4$
3210  011710                                  50002$:
3211                                  ;SET BIT FOR M8203 LINE UNIT
```

```
3212  011710  052737  100000  002500          BIS     #BIT15,STAT1
3213  011716                              4$:
3214                                      ;SET BIT IN STAT1 FOR TEST CONNECTOR
3215  011716  056137  000006  002500          BIS     6(R1),STAT1
3216  011724  062701  000002                  ADD     #2,R1
3217                                      ;SET SWITCH PACK #1 IN STAT2 LOW BYTE
3218  011730  012137  002502                  MOV     (R1)+,STAT2
3219                                      ;SET SWITCH PACK #2 IN STAT2 HIGH BYTE
3220  011734  111137  002503                  MOVB    (R1),STAT2+1
3221
3222                                      ;INCREMENT LOGICAL UNIT (DEVICE) NUMBER
3223                                      ;       INC     LOGDEV
3224  011740  000240                          NOP
3225  011742  000240                          NOP
3226
3227  011744  012737  002000  002436          MOV     #2000,MEMSZ
3228  011752  005037  002432                  CLR     TYPE
3229  011756  123727  002414  000000          CMPB    WTYPE,#0
3230  011764  001425                          BEQ     ENDIT
3231  011766  123727  002414  000004          CMPB    WTYPE,#4        ;KMC?
3232  011774  001004                          BNE     5$
3233  011776  012737  000001  002432          MOV     #1,TYPE
3234  012004  000415                          BR      ENDIT
3235  012006  012737  007777  002436  5$:     MOV     #7777,MEMSZ
3236  012014  123727  002414  000006          CMPB    WTYPE,#6
3237  012022  001003                          BNE     7$
3238  012024  012737  000001  002432          MOV     #1,TYPE
3239  012032  013737  002472  002470  7$:     MOV     RUNINH,RUNB
3240  012040                          6$:
3241  012040                          ENDIT:
3242  012040                                  ENDINIT
3243  012040                          L10036:
3244  012040  104411                          TRAP    C$INIT
3245
3246                                  .EVEN
3247  012042                                  BGNAUTO
3248  012042                          L$AUTO::
3249                                          ;DEVICE DOES NOT HAVE A "READY"
3250  012042  013701  002516                  MOV     KMCSR,R1        ;R1 CONTAINS BASE M8200,4,7 ADDRESS
3251  012046  012705  000004                  MOV     #4,R5           ;4 REGISTERS TO BE TESTED
3252  012052  012737  012104  000004          MOV     #2$,4           ;SET OUT TIMEOUT TRAP
3253  012060  012737  000240  000006          MOV     #240,6          ;LEVEL 7
3254  012066  005711                  1$:     TST     (R1)            ;REFERENCE DEVICE REGISTERS
3255  012070  000240                          NOP
3256  012072  062701  000002                  ADD     #2,R1           ;NEXT REGISTER
3257  012076  005305                          DEC     R5              ;DEC REGISTER COUNT
3258  012100  001372                          BNE     1$              ;BR IF NOT LAST REGISTER
3259  012102  000405                          BR      3$
3260
3261  012104  062706  000004          2$:     ADD     #4,SP
3262  012110                                  DODU    LOGDEV
3263  012110  013700  002342                  MOV     LOGDEV,R0
3264  012114  104451                          TRAP    C$DODU
3265
3266  012116  013737  002464  000004  3$:     MOV     SAVE4,4
3267  012124  013737  002466  000006          MOV     SAVE6,6
```

```
 3268  012132                          ENDAUTO
 3269  012132                  L10037:
 3270  012132  104461                  TRAP    C$AUTO
 3271
```

```
                                .SBTTL  CLEANUP CODING SECTION

                                ;////////////////////////////////////////////////////////////////////
                                ;/ THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ;/ AT THE END OF EACH PASS.
                                ;////////////////////////////////////////////////////////////////////

012134                                  BGNCLN
012134                          L$CLEAN::
012134                                  BRESET
012134  104433                          TRAP    C$RESET

012136                                  ENDCLN
012136                          L10040:
012136  104412                          TRAP    C$CLEAN
```

```
                        .SBTTL  DROP UNIT SECTION

                        ;////////////////////////////////////////////////////////////////////////////
                        ;/ THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                        ;/ TO NO LONGER BE TESTED.
                        ;////////////////////////////////////////////////////////////////////////////

012140                          BGNDU
012140                  L$DU::
                        ;ISSUE UNIBUS RESET TO CLEAN UP
012140                          BRESET
012140  104433                  TRAP    C$RESET
012142                          ENDDU
012142                  L10041:
012142  104453                  TRAP    C$DU
```

```
                                .SBTTL  ADD UNIT SECTION

                                ;////////////////////////////////////////////////////////////////////
                                ;/ THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ;/  TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING.  IF
                                ;/  "EF.AUNIT" IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
                                ;////////////////////////////////////////////////////////////////////

012144                                  BGNAU
012144                          L$AU::
012144                                  ENDAU
012144                          L10042:
012144  104452                          TRAP    C$AU
```

```
                        .SBTTL  HARDWARE TESTS



                        ;START OF CODE BLOCK WHICH IS USED AS DATA
012146                  ROMMAP:

                                BADHEAD
                                ;************** TEST 1 ***************
                                ;*VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
                                ;*DOES NOT CAUSE A TIME OUT TRAP
012146                          BADHEAD
                                ;************** TEST 1 ***************

012146                  BGNTST
012146                  T1::
012146  013701  002516          MOV     KMCSR,R1        ;R1 CONTAINS BASE M8200,4,7 ADDRESS
012152  012705  000004          MOV     #4,R5           ;4 REGISTERS TO BE TESTED
012156  012737  012214  000004  MOV     #2$,4           ;SET OUT TIMEOUT TRAP
012164  012737  000240  000006  MOV     #240,6          ;LEVEL 7
012172  005711          1$:     TST     (R1)            ;REFERENCE DEVICE REGISTERS
012174  000240                  NOP
012176                          ESCAPE  TST
012176  104410                  TRAP    C$ESCAPE
012200  000054                  .WORD   L10043-.
012202  062701  000002          ADD     #2,R1           ;NEXT REGISTER
012206  005305                  DEC     R5              ;DEC REGISTER COUNT
012210  001370                  BNE     1$              ;BR IF NOT LAST REGISTER
012212  000410                  BR      3$

012214  062706  000004  2$:     ADD     #4,SP
012220                          ERROR   36              ;TIME OUT ERROR
012224  104455                  TRAP    C$ERDF
012226  000044                  .WORD   36
012230  005055                  .WORD   EMO
012232  010614                  .WORD   ERR36

012234  013737  002464  000004  3$:     MOV     SAVE4,4
012242  013737  002466  000006          MOV     SAVE6,6
012250                          ESCAPE  TST
012250  104410                  TRAP    C$ESCAPE
012252  000002                  .WORD   L10043-.

012254                  ENDTST
012254                  L10043:
012254  104401                  TRAP    C$ETST
                        .EVEN



012256                          BADHEAD
                                ;************** TEST 2 ***************
                                ;*TEST OF BR RIGHT SHIFT
                                ;*VERIFY THAT A DEST OF BR RSH (011) OF A MICRO-INSTRUCTION
                                ;*SHIFTS THE RESULTING BR DATA RIGHT ONCE.
012256                          BADHEAD
                                ;************** TEST 2 ***************
```

```
3387
3388  012256                                BGNTST
3389  012256                                T2::
3390
3391  012256                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
3392  012262  013701  002516                        MOV     KMCSR,R1        ;R1 = M8200,4,7 BASE ADDRESS
3393  012266  005011                                CLR     (R1)            ;CLEAR SELO
3394  012270  012705  052525                        MOV     #52525,R5       ;START WITH 125
3395  012274  010561  000004                        MOV     R5,4(R1)        ;PORT4 125
3396  012300                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION
3397  012300  004537  003044                        JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3398  012304  120500                                120500                  ;PORT4 TO BR-REG
3399  012306                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION
3400  012306  004537  003044                        JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3401  012312  061620                                061620                  ;BR RSH BR, SHIFT BR RIGHT
3402  012314                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION
3403  012314  004537  003044                        JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3404  012320  061225                                061225                  ;PORT5 BR
3405  012322  006005                                ROR     R5              ;R5 = "EXPECTED"
3406  012324  005004                                CLR     R4
3407  012326  116104  000005                        MOVB    5(R1),R4        ;R4 = "FOUND"
3408  012332  120504                                CMPB    R5,R4           ;DID BR SHIFT RIGHT ONCE?
3409  012334  001410                                BEQ     1$              ;BR IF YES
3410  012336                                        ERROR   12              ;BR RIGHT SHIFT ERROR
3411  012342  104455                                TRAP    C$ERDF
3412  012344  000014                                .WORD   12
3413  012346  005055                                .WORD   EM0
3414  012350  007420                                .WORD   ERR12
3415
3416  012352                                        ESCAPE TST              ;SHOULD BE 52
3417  012352  104410                                TRAP    C$ESCAPE
3418  012354  000044                                .WORD   L10044-.
3419  012356                                1$:
3420  012356                                        ROMCLK                  ;NEXT WORD IS INSTRUCTION
3421  012356  004537  003044                        JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3422  012362  061620                                061620                  ;BR RSH BR, SHFT BR RIGHT AGAIN
3423  012364                                        ROMCLK                  ;NE4XT WORD IS INSTRUCTION
3424  012364  004537  003044                        JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3425  012370  061225                                061225                  ;PORT5 BR
3426  012372  006005                                ROR     R5              ;R5 = "EXPECTED"
3427  012374  116104  000005                        MOVB    5(R1),R4        ;R4 = "FOUND"
3428  012400  120504                                CMPB    R5,R4           ;DID BR SHIFT RIGHT?
3429  012402  001406                                BEQ     2$              ;BR IF YES
3430  012404                                        ERROR   12              ;BR RIGHT SHIFT ERROR
3431  012410  104455                                TRAP    C$ERDF
3432  012412  000014                                .WORD   12
3433  012414  005055                                .WORD   EM0
3434  012416  007420                                .WORD   ERR12
3435                                                                        ;S/B 25
3436  012420                                2$:
3437  012420                                ENDTST
3438  012420                                L10044:
3439  012420  104401                                TRAP    C$ETST
3440
3441  012422                                        BADHEAD
3442                                                ;************** TEST 3 **************
```

```
                                        ;*IOP CRAM WRITE/READ TEST
                                        ;*FLOAT A 1 THROUGH EACH CRAM LOCATION
012422                                  BADHEAD
                                        ;*************** TEST 3 ***************

012422                          BGNTST
012422                          T3::
012422                                  MACEX
                                        ;DO NOT DO TEST IF M8200
012430  104432                          TRAP    C$EXIT
012432  000116                          .WORD   L10045-.
012434                                  MYINT
012434  013701  002516                  MOV     KMCSR,R1            ;RECORD DEVICE ADDR.
                                                                    ;R1 CONTAINS BASE M8200,4,7 ADDRESS
012440  005037  002434                  CLR     MR0                 ;MR0 = CRAM ADDRESS
012444  012702  000001          ADR4:   MOV     #1,R2               ;R2 = WRITE DATA
012450                          ADR5:
012450                                  BGNSEG
012450  104404                          TRAP    C$BSEG
012452  012711  002000          3$:     MOV     #BIT10,(R1)         ;SET ROM0
012456  013761  002434  000004          MOV     MR0,4(R1)           ;WRITE ADDRESS TO SEL4
012464  010261  000006                  MOV     R2,6(R1)            ;LOAD SEL6 WITH WRITE DATA
012470  052711  020000                  BIS     #BIT13,(R1)         ;WRITE SEL6 INTO CRAM
012474  016104  000006                  MOV     6(R1),R4            ;READ CRAM INTO "FOUND"
012500  020204                          CMP     R2,R4               ;IS DATA CORRECT?
012502  001410                          BEQ     4$                  ;BR IF OK
012504                                  ERROR   1                   ;ERROR
012510  104455                          TRAP    C$ERDF
012512  000001                          .WORD   1
012514  005055                          .WORD   EM0
012516  006032                          .WORD   ERR1
012520                                  ESCAPE SEG
012520  104410                          TRAP    C$ESCAPE
012522  000002                          .WORD   10000$-.
012524                          4$:     ENDSEG
012524                          10000$:
012524  104405                          TRAP    C$ESEG
012526  000241                          CLC                         ;CLEAR CARRY
012530  006102                          ROL     R2                  ;SHIFT WRITE DATA
012532  001346                          BNE     ADR5                ;BSR IF NOT DONE THIS ADDRESS
012534  005237  002434                  INC     MR0                 ;BUMP TO NEXT CRAM ADDRESS
012540  023737  002436  002434          CMP     MEMSZ,MR0           ;DONE YET?
012546  001336                          BNE     ADR4                ;BR IF NO
012550                          5$:
012550                          ENDTST
012550                          L10045:
012550  104401                          TRAP    C$ETST

012552                                  BADHEAD
                                        ;*************** TEST 4 ***************
                                        ;*IOP CRAM WRITE/READ TEST
                                        ;*FLOAT A 0 THROUGH EACH CRAM LOCATION
012552                                  BADHEAD
                                        ;*************** TEST 4 ***************

012552                          BGNTST
```

```
3499  012552                          T4::
3500  012552                          MACEX
3501                                  ;DO NOT DO TEST IF M8200
3502  012560  104432                  TRAP    C$EXIT
3503  012562  000126                  .WORD   L10046-.
3504  012564                          MYINT
3505  012564  013701  002516          MOV     KMCSR,R1            ;RECORD DEVICE ADDR.
3506  012570                          MSTCLR                      ;MASTER CLEAR M8200,4,7
3507  012574  005037  002434          CLR     MR0                 ;MR0 = CRAM ADDRESS
3508  012600  012702  000001  ADR1:   MOV     #1,R2               ;R2 = WRITE DATA
3509  012604                  ADR2:
3510  012604                          BGNSEG
3511  012604  104404                  TRAP    C$BSEG
3512  012606  005102                  COM     R2                  ;MAKE IT A FLOATING ZERO
3513  012610  012711  002000  3$:     MOV     #BIT10,(R1)         ;SET ROM0
3514  012614  013761  002434  000004 MOV     MR0,4(R1)           ;WRITE ADDRESS TO SEL4
3515  012622  010261  000006          MOV     R2,6(R1)            ;LOAD SEL6 WITH WRITE DATA
3516  012626  052711  020000          BIS     #BIT13,(R1)         ;WRITE SEL6 INTO CRAM
3517  012632  016104  000006          MOV     6(R1),R4            ;READ CRAM INTO "FOUND"
3518  012636  020204                  CMP     R2,R4               ;IS DATA CORRECT?
3519  012640  001410                  BEQ     4$                  ;BR IF OK
3520  012642                          ERROR   1                   ;ERROR
3521  012646  104455                  TRAP    C$ERDF
3522  012650  000001                  .WORD   1
3523  012652  005055                  .WORD   EM0
3524  012654  006032                  .WORD   ERR1
3525  012656                          ESCAPE SEG
3526  012656  104410                  TRAP    C$ESCAPE
3527  012660  000002                  .WORD   10000$-.
3528  012662                  4$:     ENDSEG
3529  012662                  10000$:
3530  012662  104405                  TRAP    C$ESEG
3531  012664  005102                  COM     R2                  ;BACK TO FLOATING ONE
3532  012666  000241                  CLC                         ;CLEAR CARRY
3533  012670  006102                  ROL     R2                  ;SHIFT WRITE DATA
3534  012672  001344                  BNE     ADR2                ;BR IF NOT DONE THIS ADDRESS
3535  012674  005237  002434          INC     MR0                 ;BUMP TO NEXT CRAM ADDRESS
3536  012700  023737  002436  002434 CMP     MEMSZ,MR0           ;DONE YET?
3537  012706  001334                  BNE     ADR1                ;BR IF NO
3538  012710                  5$:
3539  012710                  ENDTST
3540  012710                  L10046:
3541  012710  104401                  TRAP    C$ETST
3542
3543  012712                          BADHEAD
3544                                  ;*************** TEST 5 ***************
3545                                  ;*IOP CRAM DUAL ADDRESSING TEST
3546                                  ;*WRITE EACH ADDRESS INTO ITSELF, READ EACH
3547                                  ;*ADDRESS TO VERIFY CORRECT ADDRESSING
3548  012712                          BADHEAD
3549                                  ;*************** TEST 5 ***************
3550
3551  012712                  BGNTST
3552  012712                  T5::
3553  012712                  MACEX
3554                                  ;DO NOT DO TEST IF M8200
```

```
3555  012720  104432                    TRAP    C$EXIT
3556  012722  000230                    .WORD   L10047-.
3557  012724                            MYINT
3558  012724  013701  002516            MOV     KMCSR,R1        ;RECORD DEVICE ADDR.
3559                                                              ;R1 CONTAINS BASE M8200,4,7 ADDRESS
3560  012730                            MSTCLR                  ;MASTER CLEAR M8200,4,7
3561  012734  005037  002434            CLR     MR0             ;MR0 =CRAM ADDRESS
3562  012740                            BGNSEG
3563  012740  104404                    TRAP    C$BSEG
3564  012742  013702  002434      1$:   MOV     MR0,R2          ;SAVE R2 FOR TYPEOUT
3565  012746  012711  002000            MOV     #BIT10,(R1)     ;SET ROM0
3566  012752  013761  002434  000004    MOV     MR0,4(R1)       ;WRITE ADDRESS TO SEL4
3567  012760  013761  002434  000006    MOV     MR0,6(R1)       ;LOAD SEL6 WITH WRITE DATA
3568  012766  052711  020000            BIS     #BIT13,(R1)     ;WRITE CRAM
3569  012772                            SKIP06  15$             ;IF M8206,SKIP NEXT INSTR.
3570                                    ;GOTO 15$ IF M8206
3571  013002  005061  000006            CLR     6(R1)           ;CLEAR SEL 6
3572  013006                      15$:
3573  013006  016104  000006            MOV     6(R1),R4        ;SHOULD READ BACK OWN ADDRESS
3574  013012  023704  002434            CMP     MR0,R4          ;IS DATA CORRECT?
3575  013016  001410                    BEQ     2$              ;BR IF YES
3576  013020                            ERROR   1               ;DATA ERROR
3577  013024  104455                    TRAP    C$ERDF
3578  013026  000001                    .WORD   1
3579  013030  005055                    .WORD   EM0
3580  013032  006032                    .WORD   ERR1
3581  013034                            ESCAPE SEG
3582  013034  104410                    TRAP    C$ESCAPE
3583  013036  000002                    .WORD   10000$-.
3584  013040                      2$:   ENDSEG
3585  013040                      10000$:
3586  013040  104405                    TRAP    C$ESEG
3587  013042                            BGNSEG
3588  013042  104404                    TRAP    C$BSEG
3589  013044  005237  002434            INC     MR0             ;BUMP TO NEXT ADDRESS
3590  013050  023737  002436  002434    CMP     MEMSZ,MR0       ;DONE WRITING YET?
3591  013056  001331                    BNE     1$              ;BR IF NO
3592  013060  005037  002434            CLR     MR0             ;RESTART AT ADDRESS 0
3593  013064  013702  002434      3$:   MOV     MR0,R2          ;SAVE R2 FOR TYPEOUT
3594  013070  012711  002000            MOV     #BIT10,(R1)     ;SET ROM0
3595  013074  013761  002434  000004    MOV     MR0,4(R1)       ;SEL4 = CRAM ADDRESS
3596  013102  016104  000006            MOV     6(R1),R4        ;READ CRAM INTO ''FOUND''
3597  013106  023704  002434            CMP     MR0,R4          ;IS DATA CORRECT?
3598  013112  001411                    BEQ     4$              ;BR IF YES
3599  013114                            ERROR   2               ;DUAL ADDRESSING ERROR
3600  013120  104455                    TRAP    C$ERDF
3601  013122  000002                    .WORD   2
3602  013124  005055                    .WORD   EM0
3603  013126  006160                    .WORD   ERR2
3604  013130                            ESCAPE SEG
3605  013130  104410                    TRAP    C$ESCAPE
3606  013132  000002                    .WORD   10001$-.
3607  013134                            ENDSEG
3608  013134                      10001$:
3609  013134  104405                    TRAP    C$ESEG
3610  013136                      4$:                           ;LOOP TO 3$ IF SW09=1
```

```
 3611  013136  005237  002434                  INC     MR0             ;BUMP TO NEXT ADDRESS
 3612  013142  023737  002436  002434          CMP     MEMSZ,MR0       ;DONE WRITING YET?
 3613  013150  001345                          BNE     3$              ;BR IF NO
 3614  013152                          5$:
 3615  013152                          ENDTST
 3616  013152                          L10047:
 3617  013152  104401                          TRAP    C$ETST
 3618
 3619
 3620  013154                                  BADHEAD
 3621                                          ;*************** TEST 6 ***************
 3622                                          ;*IOP MAIN MEMORY TEST
 3623                                          ;*FLOAT A 1 THROUGH ALL MAIN MEMORY LOCATIONS
 3624  013154                                  BADHEAD
 3625                                          ;*************** TEST 6 ***************
 3626
```

```
3627  013154                                BGNTST
3628  013154                                T6::
3629  013154                                      MYINT
3630  013154  013701  002516                      MOV     KMCSR,R1          ;RECORD DEVICE ADDR.
3631                                                                          ;R1 CONTAINS BASE M8200,4,7 ADDRESS
3632  013160                                      MSTCLR                      ;MASTER CLEAR M8200,4,7
3633  013164  005037  002406                      CLR     FLAG              ;START WITH ADDRESS 0
3634  013170  012737  000001  002434  1$:         MOV     #1,MR0            ;START WITH BIT 0
3635  013176  042737  003777  013232  65$:        BIC     #3777,66$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
```

```
  3636  013204  042737  000037  013240          BIC     #37,68$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
```

```
3637  013212  153737  002406  013232          BISB    FLAG,66$        ;ADD ADDRESS TO INSTRUCTION7
3638  013220  153737  002407  013240          BISB    FLAG+1,68$      ;ADD ADDRESS TO INSTRUCTION
3639  013226                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3640  013226  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3641  013232  010000                  66$:    010000
3642  013234                                  ROMCLK
3643  013234  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3644  013240  004000                  68$:    004000                  ;LOAD MAR HI
3645  013242  013761  002434  000004          MOV     MR0,4(R1)       ;WRITE PATTERN IN PORT4
3646  013250                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3647  013250  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3648  013254  122500                          122500                  ;MOVE PORT4 TO MEMORY
3649  013256                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3650  013256  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3651  013262  040620                          040620                  ;MOVE MEMORY TO BR
3652  013264                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3653  013264  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3654  013270  061225                          61225                   ;MOVE BR TO PORT5
3655  013272  013705  002434                  MOV     MR0,R5          ;PUT "EXPECTED" IN R5
3656  013276  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" IN R4
3657  013302  120504                          CMPB    R5,R4           ;DATA CORRECT?
3658  013304  001410                          BEQ     67$             ;BR IF YES
3659  013306                                  ERROR   6               ;DATA ERROR
3660  013312  104455                          TRAP    C$ERDF
3661  013314  000006                          .WORD   6
3662  013316  005055                          .WORD   EM0
3663  013320  006700                          .WORD   ERR6
3664  013322                                  ESCAPE TST
3665  013322  104410                          TRAP    C$ESCAPE
3666  013324  000030                          .WORD   L10050-.
3667  013326                          67$:                            ;SW09=1?
3668  013326  000241                          CLC                     ;CLEAR CARRY
3669  013330  106137  002434                  ROLB    MR0             ;SHIFT BIT IN MR0
3670  013334  001320                          BNE     65$             ;DONE IF MR0=0
3671  013336                                  BREAK
3672  013336  104422                          TRAP    C$BRK
3673  013340  005237  002406                  INC     FLAG            ;NEXT ADDRESS
3674  013344  023737  002436  002406          CMP     MEMSZ,FLAG      ;LAST ADDRESS?
3675  013352  001306                          BNE     1$              ;BR IF NO
3676  013354                          2$:
3677  013354                          ENDTST
3678  013354                          L10050:
3679  013354  104401                          TRAP    C$ETST
3680
3681  013356                                  BADHEAD
3682                                          ;*************** TEST 7 ***************
3683                                          ;*IOP MAIN MEMORY TEST
3684                                          ;*FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATIONS
3685  013356                                  BADHEAD
3686                                          ;*************** TEST 7 ***************
3687
3688  013356                          BGNTST
3689  013356                          T7::
3690  013356                                  MYINT
3691  013356  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
3692                                                                  ;R1 CONTAINS BASE M8200,4,7 ADDRESS
```

```
3693  013362                                    MSTCLR                  ;MASTER CLEAR M8200,4,7
3694  013366  005037  002406                    CLR     FLAG            ;START WITH ADDRESS 0
3695  013372  012737  000001  002434  1$:       MOV     #1,MR0          ;START WITH BIT 0
3696  013400  005137  002434          64$:      COM     MR0             ;CHANGE TO FLOATING 0
3697  013404  042737  003777  013440  65$:      BIC     #3777,66$       ;CLEAR ADDRESS FIELD OF INSTRUCTION
3698  013412  042737  000037  013446            BIC     #37,68$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
3699  013420  153737  002406  013440            BISB    FLAG,66$        ;ADD ADDRESS TO INSTRUCTION
3700  013426  153737  002407  013446            BISB    FLAG+1,68$      ;ADD ADDRESS TO INSTRUCTION
3701  013434                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3702  013434  004537  003044                    JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3703  013440  010000                  66$:      010000                  ;LOAD MAR LO WITH ADDRESS IN FLAG
3704  013442                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3705  013442  004537  003044                    JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3706  013446  004000                  68$:      004000                  ;LOAD MAR HI
3707  013450  013761  002434  000004            MOV     MR0,4(R1)       ;WRITE PATTERN IN PORT4
3708  013456                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3709  013456  004537  003044                    JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3710  013462  122500                            122500                  ;MOVE PORT4 TO MEMORY
3711  013464                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3712  013464  004537  003044                    JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3713  013470  040620                            040620                  ;MOVE MEMORY TO BR
3714  013472                                    ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3715  013472  004537  003044                    JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3716  013476  061225                            61225                   ;MOVE BR TO PORT5
3717  013500  013705  002434                    MOV     MR0,R5          ;PUT "EXPECTED" IN R5
3718  013504  116104  000005                    MOVB    5(R1),R4        ;PUT "FOUND" IN R4
3719  013510  120504                            CMPB    R5,R4           ;DATA CORRECT?
3720  013512  001406                            BEQ     67$             ;BR IF YES
3721  013514                                    ERROR   6               ;DATA ERROR
3722  013520  104455                            TRAP    C$ERDF
3723  013522  000006                            .WORD   6
3724  013524  005055                            .WORD   EM0
3725  013526  006700                            .WORD   ERR6
3726  013530                          67$:      ESCAPE  TST
3727  013530  104410                            TRAP    C$ESCAPE
3728  013532  000034                            .WORD   L10051-.
3729  013534  005137  002434                    COM     MR0             ;CHANGE TO FLOATING 1
3730  013540  000241                            CLC                     ;CLEAR CARRY
3731  013542  106137  002434                    ROLB    MR0             ;SHIFT BIT IN MR0
3732  013546  001314                            BNE     64$             ;DONE IF MR0=0
3733  013550                                    BREAK
3734  013550  104422                            TRAP    C$BRK
3735  013552  005237  002406                    INC     FLAG            ;NEXT ADDRESS
3736  013556  023737  002436  002406            CMP     MEMSZ,FLAG      ;LAST ADDRESS?
3737  013564  001302                            BNE     1$              ;BR IF NO
3738  013566                          2$:
3739  013566                          ENDTST
3740  013566                          L10051:
3741  013566  104401                            TRAP    C$ETST
3742
3743  013570                                    BADHEAD
3744                                            ;************** TEST 8 ***************
3745                                            ;*IOP MAIN MEMORY DUAL ADDRESSING TEST
3746                                            ;*LOAD EACH MEMORY LOCATION WITH ITS OWN ADDRESS
3747                                            ;*READ BACK EACH LOCATION TO VERIFY CORRECT ADDRESSING
3748  013570                                    BADHEAD
```

```
                                        ;*************** TEST 8 ****************

                                BGNTST
                                T8::
                                        MYINT
013570  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
                                                                        ;R1 CONTAINS BASE M8200,4,7 ADDRESS
013574                                  MSTCLR                          ;MASTER CLEAR M8200,4,7
013600  005037  002406                  CLR     FLAG                    ;START WITH ADDRESS 0
013604  013702  002406          1$:     MOV     FLAG,R2                 ;PUT DATA IN R2
013610  042737  003777  013644          BIC     #3777,2$                ;CLEAR ADDRESS FIELD OF INSTRUCTION
013616  042737  000037  013652          BIC     #37,7$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
013624  153737  002406  013644          BISB    FLAG,2$                 ;ADD ADDRESS TO INSTRUCTION
013632  153737  002407  013652          BISB    FLAG+1,7$               ;ADD ADDRESS TO INSTRUCTION
013640                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,
013640  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
013644  010000                  2$:     010000                          ;LOAD MAR LO
013646                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,
013646  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
013652  004000                  7$:     004000                          ;LOAD MAR HI
013654  010261  000004                  MOV     R2,4(R1)
013660                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,
013660  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
013664  122500                          122500                          ;MOVE PORT4 TO MEMORY
013666                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,
013666  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
013672  040620                          040620                          ;MOVE MEMORY TO THE BR
013674                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,
013674  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
013700  061225                          61225                           ;MOVE BR TO PORT5
013702  010205                          MOV     R2,R5                   ;PUT "EXPECTED" IN R5
013704  116104  000005                  MOVB    5(R1),R4                ;PUT "FOUND" IN R4
013710  120504                          CMPB    R5,R4                   ;DATA CORRECT?
013712  001406                          BEQ     3$                      ;BR IF YES
013714                                  ERROR   6                       ;DATA ERROR
013720  104455                          TRAP    C$ERDF
013722  000006                          .WORD   6
013724  005055                          .WORD   EMO
013726  006700                          .WORD   ERR6
013730                          3$:     ESCAPE TST
013730  104410                          TRAP    C$ESCAPE
013732  000156                          .WORD   L10052-.
013734  005237  002406                  INC     FLAG                    ;NEXT ADDRESS
013740  023737  002436  002406          CMP     MEMSZ,FLAG              ;LAST ADDRESS?
013746  001316                          BNE     1$                      ;BR IF NO
013750  012737  013762  002340          MOV     #4$,LOCK                ;NEW SCOPE 1
013756  005037  002406                  CLR     FLAG                    ;RESTART AT ADDRESS 0
013762  013702  002406          4$:     MOV     FLAG,R2                 ;PUT DATA IN R2
013766  042737  003777  014014          BIC     #3777,5$                ;CLEAR ADDRESS FIELD OF INSTRUCTION
013774  042737  000037  014022          BIC     #37,8$                  ;CLEAR ADDRESS FIELD OF INSTRUCTION
014002  153737  002406  014014          BISB    FLAG,5$                 ;ADD ADDRESS TO INSTRUCTION
014010                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION, ROMCLK PC+5304
014010  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
014014  010000                  5$:     010000                          ;LOAD THE MAR LO
014016                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,
014016  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
```

```
3805  014022  004000                  8$:     004000                  ;LOAD MAR HI
3806  014024                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3807  014024  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3808  014030  040620                          040620                  ;MOVE MEMORY TO THE BR
3809  014032                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3810  014032  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3811  014036  061225                          61225                   ;MOV BR TO PORT5
3812  014040  010205                          MOV     R2,R5           ;PUT "EXPECTED" IN R5
3813  014042  116104  000005                  MOVB    5(R1),R4        ;PUT "FOUND" IN R4
3814  014046  120504                          CMPB    R5,R4           ;DATA CORRECT?
3815  014050  001406                          BEQ     6$              ;BR IF YES
3816  014052                                  ERROR   6               ;ADDRESSING ERROR
3817  014056  104455                          TRAP    C$ERDF
3818  014060  000006                          .WORD   6
3819  014062  005055                          .WORD   EM0
3820  014064  006700                          .WORD   ERR6
3821  014066                          6$:     ESCAPE TST
3822  014066  104410                          TRAP    C$ESCAPE
3823  014070  000020                          .WORD   L10052-.
3824  014072                                  BREAK
3825  014072  104422                          TRAP    C$BRK
3826  014074  005237  002406                  INC     FLAG            ;NEXT ADDRESS
3827  014100  023737  002436  002406          CMP     MEMSZ,FLAG      ;IS IT THE LAST
3828  014106  001325                          BNE     4$              ;BR IF NO
3829  014110                          9$:
3830  014110                          ENDTST
3831  014110                          L10052:
3832  014110  104401                          TRAP    C$ETST
3833
3834  014112                                  BADHEAD
3835                                          ;*************** TEST 9 ***************
3836                                          ;*IOP MAR TEST
3837                                          ;*PERFORM DUAL ADDRESSING TEST
3838                                          ;*USING MAR AUTO-INC FEATURE
3839  014112                                  BADHEAD
3840                                          ;*************** TEST 9 ***************
3841
3842  014112                  BGNTST
3843  014112                  T9::
3844  014112                          K4ONLY                          ;FOR 4K CPUS ONLY.
3845                                  ;DO NOT DO TEST IF M8200, OR M8204
3846  014122  104432                  TRAP    C$EXIT
3847  014124  000342                  .WORD   L10053-.
3848  014126                          MYINT
3849  014126  013701  002516          MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
3850                                                          ;R1 CONTAINS BASE M8200,4,7 ADDRESS
3851  014132                          MSTCLR                  ;MASTER CLEAR M8200,4,7
3852  014136  005002                  CLR     R2              ;START WITH A ZERO
3853  014140  013703  002436          MOV     MEMSZ,R3        ;GET MEMORY SIZE
3854  014144  005203                  INC     R3              ;STOP ADDR=MEMSZ+1
3855  014146                          ROMCLK                  ;NEXT WORD IS INSTRUCTION,
3856  014146  004537  003044          JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3857  014152  010000                  010000                  ;LOAD MAR WITH A ZERO
3858  014154                          CLRMAR
3859  014154  004537  003044          JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
3860  014162  010261  000004  1$:     MOV     R2,4(R1)        ;WRITE DATA TO PORT4
```

```
3861  014166                                ROMCLK                          ;NEXT WORD IS INSTRUCTION,
3862  014166  004537  003044                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
3863  014172  136500                        136500                          ;MEM_PORT4, AUTO-INC MAR
3864  014174  005202                        INC     R2                      ;INCREMENT DATA
3865  014176  020302                        CMP     R3,R2   ;DONE YET?
3866  014200  001370                        BNE     1$                      ;BR IF NO
3867  014202  005002                        CLR     R2                      ;RESTART WITH A ZERO
3868  014204                                ROMCLK                          ;NEXT WORD IS INSTRUCTION,
3869  014204  004537  003044                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
3870  014210  010000                        010000                          ;LOAD MAR WITH A ZERO
3871  014212                                CLRMAR
3872  014212  004537  003044                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
3873  014220                          2$:
3874  014220                                SROMCLK                         ;NEXT WORD IS INSTRUCTION,
3875  014220  004537  003100                JSR     R5,.SROMCLK
3876  014224  055224                        055224                          ;MOVE MEM TO PORT4
3877  014226  010205                        MOV     R2,R5                   ;PUT "EXPECTED" IN R5
3878  014230  116104  000004                MOVB    4(R1),R4                ;PUT "FOUND" IN R4
3879  014234  120504                        CMPB    R5,R4                   ;DATA CORRECT?
3880  014236  001406                        BEQ     3$                      ;BR IF YES
3881  014240                                ERROR   11                      ;MAR ERROR
3882  014244  104455                        TRAP    C$ERDF
3883  014246  000013                        .WORD   11
3884  014250  005055                        .WORD   EMO
3885  014252  007276                        .WORD   ERR11
3886  014254                          3$:   SROMCLK
3887  014254  004537  003100                JSR     R5,.SROMCLK
3888  014260  000000                        0                               ;DUMP NOP INSTR. TO CLK AUTO INC IN MAR.
3889  014262  005004                        CLR     R4
3890  014264                                ROMCLK          ;READ IBUS* <15> (MAR HIGH)
3891  014264  004537  003044                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
3892  014270  121325                        121325          ;MAR HIGH _POT 5
3893
3894  014272                                ROMCLK          ;READ IBUS* <14> (MAR LOW)
3895  014272  004537  003044                JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
3896  014276  121304                        121304
3897  014300  016104  000004                MOV     4(R1),R4                ;ADD TO MAR HIGH.
3898  014304  042704  160000                BIC     #160000,R4
3899  014310  005202                        INC     R2
3900  014312  020237  002436                CMP     R2,MEMSZ
3901  014316  001002                        BNE     35$
3902  014320  052702  010000                BIS     #10000,R2               ;IF AT HIGH LIMIT,ADD IN OVERFLOW BIT.
3903  014324                          35$:
3904  014324  020204                        CMP     R2,R4   ;ADDR. OK?
3905  014326  001406                        BEQ     4$
3906  014330                                ERROR   11                      ;ERROR MAR ADDR. BAD IN IBUS <14>AND <15>
3907  014334  104455                        TRAP    C$ERDF
3908  014336  000013                        .WORD   11
3909  014340  005055                        .WORD   EMO
3910  014342  007276                        .WORD   ERR11
3911                                                                        ;EXPECTED (R4) IS COMBINATION OF
3912                                                                        ;IBUS* <14> AND <15>
3913  014344                          4$:
3914  014344                                ESCAPE  TST
3915  014344  104410                        TRAP    C$ESCAPE
3916  014346  000120                        .WORD   L10053-.
```

```
3917  014350                                BREAK
3918  014350  104422                        TRAP    C$BRK
3919  014352  032702  010000                BIT     #10000,R2       ;DONE YET?
3920  014356  001720                        BEQ     2$
3921                                        ;*
3922                                        ;*THIS SECTION OF CODE ADDED TO MAKE SURE
3923                                        ;*THAT MASTER CLEAR, CLEARS THE MAR
3924                                        ;*
3925
3926  014360                                SKIP06  40$
3927                                        ;GOTO 40$ IF M8206
3928  014370  005737  002470                TST     RUNB
3929  014374  001034                        BNE     40$
3930  014376  005737  002472                TST     RUNINH
3931  014402  001031                        BNE     40$
3932  014404  052711  040000                BIS     #40000,(R1)     ;SET MASTER CLEAR
3933  014410  005011                        CLR     (R1)            ;CLEAR MASTER CLEAR
3934  014412                                ROMCLK                  ;WE MUST FIRST CLOCK
3935  014412  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3936  014416  121325                        121325                  ;THE MAR LATCH REGS
3937  014420                                ROMCLK                  ;BEFORE WE CAN READ THEM
3938  014420  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3939  014424  121304                        121304
3940  014426                                ROMCLK                  ;READ IBUS* <15> PUT IN PORT5
3941  014426  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3942  014432  121325                        121325                  ;MAR HIGH
3943  014434                                ROMCLK                  ;READ IBUS* <14>, PUT IN PORT4
3944  014434  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
3945  014440  121304                        121304                  ;MAR LOW
3946  014442  005002                        CLR     R2              ;EXPECT MAR CLEAR
3947  014444  016104  000004                MOV     4(R1),R4        ;READ PORTS 4&5. THEY CONTAIN
3948                                                                ;THE CONTENTS OF THE MAR
3949                                                                ;MASTER CLEAR SHOULD HAVE
3950                                                                ;CLEARED THE MAR
3951                                                                ;BRANCH END TST IF CLEAR
3952  014450  001406                        BEQ     40$
3953  014452                                ERROR   44
3954  014456  104455                        TRAP    C$ERDF
3955  014460  000054                        .WORD   44
3956  014462  005055                        .WORD   EM0
3957  014464  011054                        .WORD   ERR44
3958  014466                        40$:
3959  014466                        ENDTST
3960  014466                        L10053:
3961  014466  104401                        TRAP    C$ETST
3962
3963  014470                                BADHEAD
3964                                        ;*************** TEST 10 ***************
3965                                        ;*IOP (CRAM) ODT BITS TEST
3966                                        ;*LOAD MAR WITH A 0 INC MAR UNTIL IT OVERFLOWS
3967                                        ;*VERIFY THAT IBUS* 10 BITS IS SET ONLY WHEN MAR BIT 8 IS A ONE
3968                                        ;*AND THAT IBUS* 10 BIT6 IS SET ON MAR OVERFLOW
3969  014470                                BADHEAD
3970                                        ;*************** TEST 10 ***************
3971
3972  014470                        BGNTST
```

```
3973  014470                          T10::
3974  014470                                MACEX
3975                                        ;DO NOT DO TEST IF M8200
3976  014476  104432                        TRAP    C$EXIT
3977  014500  000234                        .WORD   L10054-.
3978  014502                                MYINT
3979  014502  013701  002516                MOV     KMCSR,R1            ;RECORD DEVICE ADDR.
3980                                                                    ;R1 CONTAINS BASE M8200,4,7 ADDRESS
3981  014506                                MSTCLR                      ;MASTER CLEAR M8200,4,7
3982  014512  005002                        CLR     R2                  ;R2=SAME AS MAR CONTENTS
3983  014514                                ROMCLK                      ;NEXT WORD IS INSTRUCTION,
3984  014514  004537  003044                JSR     R5,.ROMCLK          ;CLOCK INSTRUCTION
3985  014520  010000                        010000                      ;MAR_0
3986  014522                          1$:
3987  014522                                ROMCLK                      ;NEXT WORD IS INSTRUCTION,
3988  014522  004537  003044                JSR     R5,.ROMCLK          ;CLOCK INSTRUCTION
3989  014526  121204                        121204                      ;PORT4=IBUS*10
3990  014530  005005                        CLR     R5                  ;R5=''EXPECTED''
3991  014532  032702  000400                BIT     #BIT8,R2            ;IS BIT8 SET IN MAR?
3992  014536  001402                        BEQ     .+6                 ;BR IF NO
3993  014540  012705  000040                MOV     #BIT5,R5            ;IF YES THEN SET BIT5
3994  014544  016104  000004                MOV     4(R1),R4            ;R4=''FOUND''
3995  014550  042704  177637                BIC     #177637,R4          ;CLEAR UNWANTED BITS
3996  014554  020504                        CMP     R5,R4               ;BITS 5&6 SHOULD BE CLEAR
3997  014556  001410                        BEQ     15$                 ;BR IF OK
3998  014560                                ERROR   7                   ;ERROR BITS 5&6 NOT CLEAR
3999  014564  104455                        TRAP    C$ERDF
4000  014566  000007                        .WORD   7
4001  014570  005055                        .WORD   EM0
4002  014572  007026                        .WORD   ERR7
4003  014574                                ESCAPE  TST
4004  014574  104410                        TRAP    C$ESCAPE
4005  014576  000136                        .WORD   L10054-.
4006  014600                          15$:
4007  014600                                ROMCLK                      ;NEXT WORD IS INSTRUCTION,
4008  014600  004537  003044                JSR     R5,.ROMCLK          ;CLOCK INSTRUCTION
4009  014604  014000                        014000                      ;INC MAR
4010  014606  005202                        INC     R2                  ;BUMP MEM ADDRESS
4011  014610  022702  002000                CMP     #2000,R2            ;OVERFLOWED YET?(OVFL PAGE BITS).
4012  014614  001342                        BNE     1$                  ;BR IF NO
4013  014616                                ROMCLK                      ;NEXT WORD IS INSTRUCTION,
4014  014616  004537  003044                JSR     R5,.ROMCLK          ;CLOCK INSTRUCTION
4015  014622  121204                        121204                      ;PART4 IBUS* 10
4016  014624  012705  000100                MOV     #BIT6,R5            ;R5=''EXPECTED''
4017  014630  016104  000004                MOV     4(R1),R4            ;R4=''FOUND''
4018  014634  042704  177627                BIC     #177627,R4          ;CLEAR UNWANTED BITS
4019  014640  020504                        CMP     R5,R4               ;BIT6 SHOULD BE SET
4020  014642  001406                        BEQ     17$                 ;BR IF OK
4021  014644                                ERROR   7                   ;ERROR, BIT6 NOT SET
4022  014650  104455                        TRAP    C$ERDF
4023  014652  000007                        .WORD   7
4024  014654  005055                        .WORD   EM0
4025  014656  007026                        .WORD   ERR7
4026  014660                          17$:  ROMCLK                      ;NEXT WORD IS INSTRUCTION,
4027  014660  004537  003044                JSR     R5,.ROMCLK          ;CLOCK INSTRUCTION
4028  014664  010000                        010000                      ;MAR_0
```

```
4029  014666
4030  014666  004537  003044          ROMCLK                  ;NEXT WORD IS INSTRUCTION,
4031  014672  004000                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
4032  014674                          004000                  ;MAR HI 0
4033  014674  004537  003044          ROMCLK                  ;NEXT WORD IS INSTRUCTION,
4034  014700  121204                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
4035  014702  005005                  121204                  ;PORT4 IBUS* 10
4036  014704  016104  000004          CLR     R5              ;R5="EXPECTED"
4037  014710  042704  177637          MOV     4(R1),R4        ;R4="FOUND"
4038  014714  020504                  BIC     #177637,R4      ;CLEAR UNWANTED BITS
4039  014716  001406                  CMP     R5,R4           ;BITS 5&6 SHOULD BE CLEAR
4040  014720                          BEQ     2$              ;BR IF OK
4041  014724  104455                  ERROR   7               ;ERROR 5&6 NOT BOTH CLEAR
4042  014726  000007                  TRAP    C$ERDF
4043  014730  005055                  .WORD   7
4044  014732  007026                  .WORD   EM0
4045  014734                          .WORD   ERR7
4046  014734                  2$:
4047  014734                  ENDTST
4048  014734  104401          L10054:
4049                                  TRAP    C$ETST
4050  014736
4051                                  BADHEAD
4052                                  ;************** TEST 11 ***************
4053                                  ;*CRAM TEST OF JUMP(I) NEVER MICRO-PROCESSOR INSTRUCTION.
4054                                  ;*PERFORM THE JUMP INSTRUCTION
4055                                  ;*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
4056                                  ;*IN THE LOCATION IT IS AT.  THIS INSTRUCTION LOADS THE
4057                                  ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4058                                  ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4059                                  ;*THE CRAM PC IS CORRECT. IF THE CRAM PC IS NOT RIGHT,
4060  014736                          ;*THEN PORT4 CONTAINS A 37
4061                                  BADHEAD
4062                                  ;************** TEST 11 ***************
4063  014736
4064  014736                  BGNTST
4065  014736                  T11::
4066                                  SKIP04  10$
4067  014746                          ;GOTO 10$ IF M8204
4068  014746  104432                  EXIT    TST                     ;CAN'T DO IF ROM,4K
4069  014750  000230                  TRAP    C$EXIT
4070  014752                          .WORD   L10055-.
4071  014752                  10$:
4072  014752  013701  002516          MYINT
4073                                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
4074  014756                                                  ;R1 CONTAINS BASE M8200,4,7 ADDRESS
4075  014762                          MSTCLR                  ;MASTER CLEAR M8200,4,7
4076  014762  104404                  BGNSEG
4077  014764  004737  003474          TRAP    C$BSEG
4078  014770                          JSR     PC,MEMSET       ;SET MEM AND RAM
4079  014770  004737  003166  1$:
4080  014774                          JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
4081  014774  004537  003100          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4082  015000  100400                  JSR     R5,.SROMCLK
4083  015002                          100400                  ;START AT ROM PC=0
4084  015002  004537  003100          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
                                      JSR     R5,.SROMCLK
```

```
4085  015006  114377                    114377!<400*0>          ;JUMP TO ROM PC OF 1777
4086  015010  004737  003330            JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4087  015014  000001                    1                       ;EXPECTED DATA
4088  015016  120504                    CMPB    R5,R4           ;IS ROM PC CORRECT?
4089  015020  001406                    BEQ     2$              ;BR IF NO
4090  015022                            ERROR   5               ;ERROR, CRAM PC IS WRONG
4091  015026  104455                    TRAP    C$ERDF
4092  015030  000005                    .WORD   5
4093  015032  005055                    .WORD   EM0
4094  015034  006556                    .WORD   ERR5
4095  015036                    2$:     ESCAPE SEG
4096  015036  104410                    TRAP    C$ESCAPE
4097  015040  000002                    .WORD   10000$-.
4098  015042                            ENDSEG
4099  015042                    10000$:
4100  015042  104405                    TRAP    C$ESEG
4101  015044                            BGNSEG
4102  015044  104404                    TRAP    C$BSEG
4103  015046  004737  003166            JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
4104  015052                            SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4105  015052  004537  003100            JSR     R5,.SROMCLK
4106  015056  100403                    100403                  ;START AT ROM PC=3
4107  015060                            SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4108  015060  004537  003100            JSR     R5,.SROMCLK
4109  015064  100000                    100000!<400*0>  ;JUMP TO ROM PC OF 0
4110  015066  004737  003330            JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4111  015072  000004                    4                       ;EXPECTED DATA
4112  015074  120504                    CMPB    R5,R4           ;IS ROM PC CORRECT?
4113  015076  001406                    BEQ     4$              ;BR IF YES
4114  015100                            ERROR   5               ;ERROR, CROM PC IS WRONG
4115  015104  104455                    TRAP    C$ERDF
4116  015106  000005                    .WORD   5
4117  015110  005055                    .WORD   EM0
4118  015112  006556                    .WORD   ERR5
4119  015114                    4$:     ESCAPE SEG
4120  015114  104410                    TRAP    C$ESCAPE
4121  015116  000002                    .WORD   10001$-.
4122  015120                            ENDSEG
4123  015120                    10001$:
4124  015120  104405                    TRAP    C$ESEG
4125  015122                            BGNSEG
4126  015122  104404                    TRAP    C$BSEG
4127  015124  004737  003166            JSR     PC,CLRALL       ;CLEAR ALL CONDITINS
4128  015130                            SROMCLK                 ;NEXT WORD IS INSTRUCION,
4129  015130  004537  003100            JSR     R5,.SROMCLK
4130  015134  100406                    100406                  ;START AT ROM PC=6
4131  015136                            SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4132  015136  004537  003100            JSR     R5,.SROMCLK
4133  015142  104125                    104125!<400*0>          ;JUMP TO ROM PC OF 525
4134  015144  004737  003330            JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4135  015150  000007                    7                       ;EXPECTED DATA
4136  015152  120504                    CMPB    R5,R4           ;IS ROM PC CORRECT?
4137  015154  001406                    BEQ     6$              ;BR IF YES
4138  015156                            ERROR   5               ;ERROR, CRAM PC IS WRONG
4139  015162  104455                    TRAP    C$ERDF
4140  015164  000005                    .WORD   5
```

```
4141  015166  005055                        .WORD   F%0
4142  015170  006556                        .WORD   ERR5
4143  015172                        6$:     ESCAPE SEG
4144  015172  104410                        TRAP    C$ESCAPE
4145  015174  000002                        .WORD   10002$-.
4146  015176                                ENDSEG
4147  015176                        10002$:
4148  015176  104405                        TRAP    C$ESEG
4149  015200                        ENDTST
4150  015200                        L10055:
4151  015200  104401                        TRAP    C$ETST
4152
4153  015202                                BADHEAD
4154                                        ;************** TEST 12 ***************
4155                                        ;*CRAM TEST OF JUMP(I) ALWAYS MICRO-PROCESSOR INSTRUCTION.
4156                                        ;*PERFORM THE JUMP INSTRUCTION
4157                                        ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
4158                                        ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4159                                        ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4160                                        ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4161                                        ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4162                                        ;*THEN PORT4 WILL CONTAIN A 37
4163  015202                                BADHEAD
4164                                        ;************** TEST 12 ***************
4165
4166  015202                        BGNTST
4167  015202                        T12::
4168  015202                                MACEX2                  ;DON'T DO IF M8200
4169                                        ;DO NOT DO TEST IF M8200
4170  015212  104432                        TRAP    C$EXIT
4171  015214  000214                        .WORD   L10056-.
4172  015216                                MYINT
4173  015216  013701  002516                MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
4174                                                                ;R1 CONTAINS BASE M8200,4,7 ADDRESS
4175  015222                                MSTCLR                  ;MASTER CLEAR M8200,4,7
4176  015226  004737  003474                JSR     PC,MEMSET       ;SET MEM AND RAM
4177  015232                        1$:     BGNSEG
4178  015232  104404                        TRAP    C$BSEG
4179  015234                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4180  015234  004537  003100                JSR     R5,.SROMCLK
4181  015240  100400                        100400                  ;START AT ROM PC=0
4182  015242                                SROMCLK                 ;NEXT WORD IS INSTRUCION,
4183  015242  004537  003100                JSR     R5,.SROMCLK
4184  015246  114777                        114377!<400*1>          ;JUMP TO ROM PC OF 1777
4185  015250  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4186  015254  000377                        377                     ;EXPECTED DATA
4187  015256  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
4188  015260  001406                        BEQ     2$              ;BR IF YES
4189  015262                                ERROR   5               ;ERROR, CRAM PC IS WRONG
4190  015266  104455                        TRAP    C$ERDF
4191  015270  000005                        .WORD   5
4192  015272  005055                        .WORD   EM0
4193  015274  006556                        .WORD   ERR5
4194  015276                        2$:     ESCAPE SEG
4195  015276  104410                        TRAP    C$ESCAPE
4196  015300  000002                        .WORD   10000$-.
```

```
4197  015302                                ENDSEG
4198  015302                          10000$:
4199  015302  104405                        TRAP    C$ESEG
4200  015304                                BGNSEG
4201  015304  104404                        TRAP    C$BSEG
4202  015306                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4203  015306  004537  003100                JSR     R5,.SROMCLK
4204  015312  100403                        100403                  ;START AT ROM PC=3
4205  015314                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4206  015314  004537  003100                JSR     R5,.SROMCLK
4207  015320  100400                        100000!<400*1>  ;JUMP TO ROM PC OF 0
4208  015322  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4209  015326  000000                        0                       ;EXPECTED DATA
4210  015330  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
4211  015332  001406                        BEQ     4$              ;BR IF YES
4212  015334                                ERROR   5       ;ERROR, CRAM PC IS WRONG
4213  015340  104455                        TRAP    C$ERDF
4214  015342  000005                        .WORD   5
4215  015344  005055                        .WORD   EMO
4216  015346  006556                        .WORD   ERR5
4217  015350                          4$:   ESCAPE SEG
4218  015350  104410                        TRAP    C$ESCAPE
4219  015352  000002                        .WORD   10001$-.
4220  015354                                ENDSEG
4221  015354                          10001$:
4222  015354  104405                        TRAP    C$ESEG
4223  015356                                BGNSEG
4224  015356  104404                        TRAP    C$BSEG
4225  015360                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4226  015360  004537  003100                JSR     R5,.SROMCLK
4227  015364  100406                        100406                  ;START AT ROM PC=6
4228  015366                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4229  015366  004537  003100                JSR     R5,.SROMCLK
4230  015372  104525                        104125!<400*1>          ;JUMP TO ROM PC OF 525
4231  015374  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4232  015400  000125                        125                     ;EXPECTED DATA
4233  015402  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
4234  015404  001406                        BEQ     6$              ;BR IF YES
4235  015406                                ERROR   5               ;ERROR, CRAM PC IS WRONG
4236  015412  104455                        TRAP    C$ERDF
4237  015414  000005                        .WORD   5
4238  015416  005055                        .WORD   EMO
4239  015420  006556                        .WORD   ERR5
4240  015422                          6$:   ESCAPE SEG
4241  015422  104410                        TRAP    C$ESCAPE
4242  015424  000002                        .WORD   10002$-.
4243  015426                                ENDSEG
4244  015426                          10002$:
4245  015426  104405                        TRAP    C$ESEG
4246  015430                          ENDTST
4247  015430                          L10056:
4248  015430  104401                        TRAP    C$ETST
4249
4250  015432                                BADHEAD
4251                                        ;************** TEST 13 ***************
4252                                        ;*CRAM TEST OF JUMP(1) ON C BIT SET MICRO-PROCESSOR INSTRUCTION.
```

```
                                        ;*SET THE C BIT, PERFORM THE JUMP INSTRUCTION.
                                        ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
                                        ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
                                        ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
                                        ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
                                        ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
                                        ;*THEN PORT4 WILL CONTAIN A 37
015432                                  BADHEAD
                                        ;*************** TEST 13 ***************

015432                          BGNTST
015432                          T13::
015432                          MACEX2                          ;DON'T DO IF M8200
                                ;DO NOT DO TEST IF M8200
015442  104432                  TRAP    C$EXIT
015444  000230                  .WORD   L10057-.
015446                          MYINT
015446  013701  002516          MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
                                                                ;R1 CONTAINS BASE M8200,4,7 ADDRESS
015452                          MSTCLR                          ;MASTER CLEAR M8200,4,7
015456  004737  003474          JSR     PC,MEMSET               ;SET MEM AND RAM
015462                          1$:     BGNSEG
015462  104404                  TRAP    C$BSEG
015464  004737  003260          JSR     PC,SETC                 ;SET THE C BIT'
015470                          SROMCLK                         ;NEXT WORD IS INSTRUCTION,
015470  004537  003100          JSR     R5,.SROMCLK
015474  100400                  100400                          ;START AT ROM PC=0
015476                          SROMCLK                         ;NEXT WORD IS INSTRUCION,
015476  004537  003100          JSR     R5,.SROMCLK
015502  115377                  114377!<400*2>                  ;JUMP TO ROM PC OF 1777
015504  004737  003330          JSR     PC,RAMDAT               ;R4=CRAM PC (LSB 8 BITS)
015510  000377                  377                             ;EXPECTED DATA
015512  120504                  CMPB    R5,R4                   ;IS ROM PC CORRECT?
015514  001406                  BEQ     2$                      ;BR IF YES
015516                          ERROR   5                       ;ERROR, CRAM PC IS WRONG
015522  104455                  TRAP    C$ERDF
015524  000005                  .WORD   5
015526  005055                  .WORD   EM0
015530  006556                  .WORD   ERR5
015532                          2$:                             ;LOOP TO 1$ IF SW09=1
015532                          ESCAPE SEG
015532  104410                  TRAP    C$ESCAPE
015534  000002                  .WORD   10000$-.
015536                          ENDSEG
015536                          10000$:
015536  104405                  TRAP    C$ESEG
015540                          BGNSEG
015540  104404                  TRAP    C$BSEG
015542  004737  003260          JSR     PC,SETC                 ;SET THE C BIT'
015546                          SROMCLK                         ;NEXT WORD IS INSTRUCTION,
015546  004537  003100          JSR     R5,.SROMCLK
015552  100403                  100403                          ;START AT ROM PC=3
015554                          SROMCLK                         ;NEXT WORD IS INSTRUCTION,
015554  004537  003100          JSR     R5,.SROMCLK
015560  101000                  100000!<400*2>                  ;JUMP TO ROM PC OF 0
015562  004737  003330          JSR     PC,RAMDAT               ;R4=CRAM PC (LSB 8 BITS)
```

```
4309  015566  000000                  0                       ;EXPECTED DATA
4310  015570  120504                  CMPB    R5,R4           ;IS ROM PC CORRECT?
4311  015572  001406                  BEQ     4$              ;BR IF YES
4312  015574                          ERROR   5               ;ERROR, CRAM PC IS WRONG
4313  015600  104455                  TRAP    C$ERDF
4314  015602  000005                  .WORD   5
4315  015604  005055                  .WORD   EM0
4316  015606  006556                  .WORD   ERR5
4317  015610                  4$:                             ;LOOP TO 3$ IF SW09=1
4318  015610                          ESCAPE SEG
4319  015610  104410                  TRAP    C$ESCAPE
4320  015612  000002                  .WORD   10001$-.
4321  015614                          ENDSEG
4322  015614                  10001$:
4323  015614  104405                  TRAP    C$ESEG
4324  015616                          BGNSEG
4325  015616  104404                  TRAP    C$BSEG
4326  015620  004737  003260          JSR     PC,SETC         ;SET THE C BIT'
4327  015624                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4328  015624  004537  003100          JSR     R5,.SROMCLK
4329  015630  100406                  100406                  ;START AT ROM PC=6
4330  015632                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4331  015632  004537  003100          JSR     R5,.SROMCLK
4332  015636  105125                  104125!<400*2>          ;JUMP TO ROM PC OF 525
4333  015640  004737  003330          JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4334  015644  000125                  125                     ;EXPECTED DATA
4335  015646  120504                  CMPB    R5,R4           ;IS ROM PC CORRECT?
4336  015650  001406                  BEQ     6$              ;BR IF YES
4337  015652                          ERROR   5               ;ERROR, CRAM PC IS WRONG
4338  015656  104455                  TRAP    C$ERDF
4339  015660  000005                  .WORD   5
4340  015662  005055                  .WORD   EM0
4341  015664  006556                  .WORD   ERR5
4342  015666                  6$:     ESCAPE SEG
4343  015666  104410                  TRAP    C$ESCAPE
4344  015670  000002                  .WORD   10002$-.
4345  015672                          ENDSEG
4346  015672                  10002$:
4347  015672  104405                  TRAP    C$ESEG
4348  015674                  ENDTST
4349  015674                  L10057:
4350  015674  104401                  TRAP    C$ETST
4351
4352  015676                          BADHEAD
4353                                  ;*************** TEST 14 ***************
4354                                  ;*CRAM TEST OF JUMP(I) ON Z BIT SET MICRO-PROCESSOR INSTRUCTION.
4355                                  ;*SET THE Z BIT, PERFORM THE JUMP INSTRUCTION.
4356                                  ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
4357                                  ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4358                                  ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4359                                  ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4360                                  ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4361                                  ;*THEN PORT4 WILL CONTAIN A 37
4362  015676                          BADHEAD
4363                                  ;*************** TEST 14 ***************
4364
```

```
4365  015676                          BGNTST
4366  015676                          T14::
4367  015676                                MACEX2                    ;DON'T DO IF M8200.
4368                                        ;DO NOT DO TEST IF M8200
4369  015706  104432                       TRAP    C$EXIT
4370  015710  000230                       .WORD   L10060-.
4371  015712                                MYINT
4372  015712  013701  002516                MOV     KMCSR,R1                  ;RECORD DEVICE ADDR.
4373                                                                    ;R1 CONTAINS BASE M8200,4,7 ADDRESS
4374  015716                                MSTCLR                    ;MASTER CLEAR M8200,4,7
4375  015722  004737  003474                JSR     PC,MEMSET         ;SET MEM AND RAM
4376  015726                          1$:   BGNSEG
4377  015726  104404                       TRAP    C$BSEG
4378  015730  004737  003312                JSR     PC,SETZ           ;SET THE Z BIT'
4379  015734                                SROMCLK                   ;NEXT WORD IS INSTRUCTION,
4380  015734  004537  003100                JSR     R5,.SROMCLK
4381  015740  100400                       100400                    ;START AT ROM PC=0
4382  015742                                SROMCLK                   ;NEXT WORD IS INSTRUCION,
4383  015742  004537  003100                JSR     R5,.SROMCLK
4384  015746  115777                       114377!<400*3>            ;JUMP TO ROM PC OF 1777
4385  015750  004737  003330                JSR     PC,RAMDAT         ;R4=CRAM PC (LSB 8 BITS)
4386  015754  000377                       377                       ;EXPECTED DATA
4387  015756  120504                       CMPB    R5,R4             ;IS ROM PC CORRECT?
4388  015760  001406                       BEQ     2$                ;BR IF YES
4389  015762                                ERROR   5                 ;ERROR, CRAM PC IS WRONG
4390  015766  104455                       TRAP    C$ERDF
4391  015770  000005                       .WORD   5
4392  015772  005055                       .WORD   EM0
4393  015774  006556                       .WORD   ERR5
4394  015776                          2$:   ESCAPE SEG
4395  015776  104410                       TRAP    C$ESCAPE
4396  016000  000002                       .WORD   10000$-.
4397  016002                                ENDSEG
4398  016002                          10000$:
4399  016002  104405                       TRAP    C$ESEG
4400  016004                                BGNSEG
4401  016004  104404                       TRAP    C$BSEG
4402  016006  004737  003312                JSR     PC,SETZ           ;SET THE Z BIT'
4403  016012                                SROMCLK                   ;NEXT WORD IS INSTRUCTION,
4404  016012  004537  003100                JSR     R5,.SROMCLK
4405  016016  100403                       100403                    ;START AT ROM PC=3
4406  016020                                SROMCLK                   ;NEXT WORD IS INSTRUCTION,
4407  016020  004537  003100                JSR     R5,.SROMCLK
4408  016024  101400                       100000!<400*3>            ;JUMP TO ROM PC OF 0
4409  016026  004737  003330                JSR     PC,RAMDAT         ;R4=CRAM PC (LSB 8 BITS)
4410  016032  000000                       0                         ;EXPECTED DATA
4411  016034  120504                       CMPB    R5,R4             ;IS ROM PC CORRECT?
4412  016036  001406                       BEQ     4$                ;BR IF YES
4413  016040                                ERROR   5                 ;ERROR, CRAM PC IS WRONG
4414  016044  104455                       TRAP    C$ERDF
4415  016046  000005                       .WORD   5
4416  016050  005055                       .WORD   EM0
4417  016052  006556                       .WORD   ERR5
4418  016054                          4$:   ESCAPE SEG
4419  016054  104410                       TRAP    C$ESCAPE
4420  016056  000002                       .WORD   10001$-.
```

```
4421  016060                                        ENDSEG
4422  016060                                10001$:
4423  016060  104405                                TRAP    C$ESEG
4424  016062                                        BGNSEG
4425  016062  104404                                TRAP    C$BSEG
4426  016064  004737  003312                        JSR     PC,SETZ         ;SET THE Z BIT'
4427  016070                                        SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4428  016070  004537  003100                        JSR     R5,..SROMCLK
4429  016074  100406                                100406                  ;START AT ROM PC=6
4430  016076                                        SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4431  016076  004537  003100                        JSR     R5,..SROMCLK
4432  016102  105525                                104125!<400*3>          ;JUMP TO ROM PC OF 525
4433  016104  004737  003330                        JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4434  016110  000125                                125                     ;EXPECTED DATA
4435  016112  120504                                CMPB    R5,R4           ;IS ROM PC CORRECT?
4436  016114  001406                                BEQ     6$              ;BR IF YES
4437  016116                                        ERROR   5               ;ERROR, CRAM PC IS WRONG
4438  016122  104455                                TRAP    C$ERDF
4439  016124  000005                                .WORD   5
4440  016126  005055                                .WORD   EM0
4441  016130  006556                                .WORD   ERR5
4442  016132                                6$:     ESCAPE  SEG
4443  016132  104410                                TRAP    C$ESCAPE
4444  016134  000002                                .WORD   10002$-.
4445  016136                                        ENDSEG
4446  016136                                10002$:
4447  016136  104405                                TRAP    C$ESEG
4448  016140                                ENDTST
4449  016140                                L10060:
4450  016140  104401                                TRAP    C$ETST
4451
4452  016142                                        BADHEAD
4453                                                ;*************** TEST 15 ***************
4454                                                ;*CRAM TEST OF JUMP(I) ON BR0 SET MICRO-PROCESSOR INSTRUCTION.
4455                                                ;*SET THE BR0 BIT, PERFORM THE JUMP INSTRUCTION.
4456                                                ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
4457                                                ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4458                                                ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4459                                                ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4460                                                ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4461                                                ;*THEN PORT4 WILL CONTAIN A 37
4462  016142                                        BADHEAD
4463                                                ;*************** TEST 15 ***************
4464
4465  016142                                BGNTST
4466  016142                                T15::
4467  016142                                        MACEX2                  ;DON'T DO IF M8200.
4468                                                ;DO NOT DO TEST IF M8200
4469  016152  104432                                TRAP    C$EXIT
4470  016154  000230                                .WORD   L10061-.
4471  016156                                        MYINT
4472  016156  013701  002516                        MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
4473                                                                        ;R1 CONTAINS BASE M8200,4,7 ADDRESS
4474  016162                                        MSTCLR                  ;MASTER CLEAR M8200,4,7
4475  016166  004737  003474                        JSR     PC,MEMSET       ;SET MEM AND RAM
4476  016172                                1$:
```

```
 4477  016172
 4478  016172  104404                    BGNSEG
 4479  016174  004737  003220            TRAP    C$BSEG
 4480  016200                            JSR     PC,SETBRO       ;SET THE BRO BIT'
 4481  016200  004537  003100            SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 4482  016204  100400                    JSR     R5,.SROMCLK
 4483  016206                            100400                  ;START AT ROM PC=0
 4484  016206  004537  003100            SROMCLK                 ;NEXT WORD IS INSTRUCION,
 4485  016212  116377                    JSR     R5,.SROMCLK
 4486  016214  004737  003330            114377!<400*4>          ;JUMP TO ROM PC OF 1777
 4487  016220  000377                    JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
 4488  016222  120504                    377                     ;EXPECTED DATA
 4489  016224  001406                    CMPB    R5,R4           ;IS ROM PC CORRECT?
 4490  016226                            BEQ     2$              ;BR IF YES
 4491  016232  104455                    ERROR   5               ;ERROR, CRAM PC IS WRONG
 4492  016234  000005                    TRAP    C$ERDF
 4493  016236  005055                    .WORD   5
 4494  016240  006556                    .WORD   EM0
 4495  016242                            .WORD   ERR5
 4496  016242  104410            2$:     ESCAPE SEG
 4497  016244  000002                    TRAP    C$ESCAPE
 4498  016246                            .WORD   10000$-.
 4499  016246                            ENDSEG
 4500  016246  104405            10000$:
 4501  016250                            TRAP    C$ESEG
 4502  016250  104404                    BGNSEG
 4503  016252  004737  003220            TRAP    C$BSEG
 4504  016256                            JSR     PC,SETBRO               ;SET THE BRO BIT'
 4505  016256  004537  003100            SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 4506  016262  100403                    JSR     R5,.SROMCLK
 4507  016264                            100403                  ;START AT ROM PC=3
 4508  016264  004537  003100            SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 4509  016270  102000                    JSR     R5,.SROMCLK
 4510  016272  004737  003330            100000!<400*4>          ;JUMP TO ROM PC OF 0
 4511  016276  000000                    JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
 4512  016300  120504                    0                       ;EXPECTED DATA
 4513  016302  001406                    CMPB    R5,R4           ;IS ROM PC CORRECT?
 4514  016304                            BEQ     4$              ;BR IF YES
 4515  016310  104455                    ERROR   5               ;ERROR, CRAM PC IS WRONG
 4516  016312  000005                    TRAP    C$ERDF
 4517  016314  005055                    .WORD   5
 4518  016316  006556                    .WORD   EM0
 4519  016320                            .WORD   ERR5
 4520  016320  104410            4$:     ESCAPE SEG
 4521  016322  000002                    TRAP    C$ESCAPE
 4522  016324                            .WORD   10001$-.
 4523  016324                            ENDSEG
 4524  016324  104405            10001$:
 4525  016326                            TRAP    C$ESEG
 4526  016326  104404                    BGNSEG
 4527  016330  004737  003220            TRAP    C$BSEG
 4528  016334                            JSR     PC,SETBRO               ;SET THE BRO BIT'
 4529  016334  004537  003100            SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 4530  016340  100406                    JSR     R5,.SROMCLK
 4531  016342                            100406                  ;START AT ROM PC=6
 4532  016342  004537  003100            SROMCLK                 ;NEXT WORD IS INSTRUCTION,
                                         JSR     R5,.SROMCLK
```

```
4533  016346  106125                  104125!<400*4>          ;JUMP TO ROM PC OF 525
4534  016350  004737  003330          JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4535  016354  000125                  125                     ;EXPECTED DATA
4536  016356  120504                  CMPB    R5,R4           ;IS ROM PC CORRECT?
4537  016360  001406                  BEQ     6$              ;BR IF YES
4538  016362                          ERROR   5               ;ERROR, CRAM PC IS WRONG
4539  016366  104455                  TRAP    C$ERDF
4540  016370  000005                  .WORD   5
4541  016372  005055                  .WORD   EM0
4542  016374  006556                  .WORD   ERR5
4543  016376                  6$:     ESCAPE SEG
4544  016376  104410                  TRAP    C$ESCAPE
4545  016400  000002                  .WORD   10002$-.
4546  016402                          ENDSEG
4547  016402                  10002$:
4548  016402  104405                  TRAP    C$ESEG
4549  016404                  ENDTST
4550  016404                  L10061:
4551  016404  104401                  TRAP    C$ETST
4552
4553  016406                          BADHEAD
4554                                  ;************** TEST 16 ***************
4555                                  ;*CRAM TEST OF JUMP(I) ON BR1 SET MICRO-PROCESSOR INSTRUCTION.
4556                                  ;*SET THE BR1 BIT, PERFORM THE JUMP INSTRUCTION.
4557                                  ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
4558                                  ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4559                                  ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4560                                  ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4561                                  ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4562                                  ;*THEN PORT4 WILL CONTAIN A 37
4563  016406                          BADHEAD
4564                                  ;************** TEST 16 ***************
4565
4566  016406                  BGNTST
4567  016406                  T16::
4568  016406                          MACEX2                  ;DON'T DO IF M8200.
4569                                  ;DO NOT DO TEST IF M8200
4570  016416  104432                  TRAP    C$EXIT
4571  016420  000230                  .WORD   L10062-.
4572  016422                          MYINT
4573  016422  013701  002516          MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
                                                                ;R1 CONTAINS BASE M8200,4,7 ADDRESS
4574
4575  016426                          MSTCLR                  ;MASTER CLEAR M8200,4,7
4576  016432  004737  003474          JSR     PC,MEMSET       ;SET MEM AND RAM
4577  016436                  1$:
4578  016436                          BGNSEG
4579  016436  104404                  TRAP    C$BSEG
4580  016440  004737  003230          JSR     PC,SETBR1       ;SET THE BR1 BIT'
4581  016444                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4582  016444  004537  003100          JSR     R5,.SROMCLK
4583  016450  100400                  100400                  ;START AT ROM PC=0
4584  016452                          SROMCLK                 ;NEXT WORD IS INSTRUCION,
4585  016452  004537  003100          JSR     R5,.SROMCLK
4586  016456  116777                  114377!<400*5>          ;JUMP TO ROM PC OF 1777
4587  016460  004737  003330          JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4588  016464  000377                  377                     ;EXPECTED DATA
```

```
4589  016466  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
4590  016470  001406                        BEQ     2$              ;BR IF YES
4591  016472                                ERROR   5               ;ERROR, CRAM PC IS WRONG
4592  016476  104455                        TRAP    C$ERDF
4593  016500  000005                        .WORD   5
4594  016502  005055                        .WORD   EM0
4595  016504  006556                        .WORD   ERR5
4596  016506                          2$:   ESCAPE SEG
4597  016506  104410                        TRAP    C$ESCAPE
4598  016510  000002                        .WORD   10000$-.
4599  016512                                ENDSEG
4600  016512                          10000$:
4601  016512  104405                        TRAP    C$ESEG
4602  016514                                BGNSEG
4603  016514  104404                        TRAP    C$BSEG
4604  016516  004737  003230                JSR     PC,SETBR1               ;SET THE BR1 BIT'
4605  016522                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4606  016522  004537  003100                JSR     R5,.SROMCLK
4607  016526  100403                        100403                  ;START AT ROM PC=3
4608  016530                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4609  016530  004537  003100                JSR     R5,.SROMCLK
4610  016534  102400                        100000!<400*5>          ;JUMP TO ROM PC OF 0
4611  016536  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4612  016542  000000                        0                       ;EXPECTED DATA
4613  016544  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
4614  016546  001406                        BEQ     4$              ;BR IF YES
4615  016550                                ERROR   5               ;ERROR, CRAM PC IS WRONG
4616  016554  104455                        TRAP    C$ERDF
4617  016556  000005                        .WORD   5
4618  016560  005055                        .WORD   EM0
4619  016562  006556                        .WORD   ERR5
4620  016564                          4$:   ESCAPE SEG
4621  016564  10441C                        TRAP    C$ESCAPE
4622  016566  000002                        .WORD   10001$-.
4623  016570                                ENDSEG
4624  016570                          10001$:
4625  016570  104405                        TRAP    C$ESEG
4626  016572                                BGNSEG
4627  016572  104404                        TRAP    C$BSEG
4628  016574  004737  003230                JSR     PC,SETBR1               ;SET THE BR1 BIT'
4629  016600                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4630  016600  004537  003100                JSR     R5,.SROMCLK
4631  016604  100406                        100406                  ;START AT ROM PC=6
4632  016606                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4633  016606  004537  003100                JSR     R5,.SROMCLK
4634  016612  106525                        104125!<400*5>          ;JUMP TO ROM PC OF 525
4635  016614  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4636  016620  000125                        125                     ;EXPECTED DATA
4637  016622  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
4638  016624  001406                        BEQ     6$              ;BR IF YES
4639  016626                                ERROR   5               ;ERROR, CRAM PC IS WRONG
4640  016632  104455                        TRAP    C$ERDF
4641  016634  000005                        .WORD   5
4642  016636  005055                        .WORD   EM0
4643  016640  006556                        .WORD   ERR5
4644  016642                          6$:   ESCAPE SEG
```

```
4645  016642  104410                  TRAP    C$ESCAPE
4646  016644  000002                  .WORD   10002$-.
4647  016646                          ENDSEG
4648  016646                  10002$:
4649  016646  104405                  TRAP    C$ESEG
4650  016650                  ENDTST
4651  016650                  L10062:
4652  016650  104401                  TRAP    C$ETST
4653
4654  016652                          BADHEAD
4655                                  ;*************** TEST 17 ***************
4656                                  ;*CRAM TEST OF JUMP(I) ON BR4 SET MICRO-PROCESSOR INSTRUCTION.
4657                                  ;*SET THE BR4 BIT, PERFORM THE JUMP INSTRUCTION.
4658                                  ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
4659                                  ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
4660                                  ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
4661                                  ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
4662                                  ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
4663                                  ;*THEN PORT4 WILL CONTAIN A 37
4664  016652                          BADHEAD
4665                                  ;*************** TEST 17 ***************
4666
4667  016652                  BGNTST
4668  016652                  T17::
4669  016652                          MACEX2                  ;DON'T DO IF M8200.
4670                                  ;DO NOT DO TEST IF M8200
4671  016662  104432                  TRAP    C$EXIT
4672  016664  000230                  .WORD   L10063-.
4673  016666                          MYINT
4674  016666  013701  002516          MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
4675  016672                          MSTCLR                  ;MASTER CLEAR M8200,4,7
4676  016676  004737  003474          JSR     PC,MEMSET       ;SET MEM AND RAM
4677  016702                  1$:
4678  016702                          BGNSEG
4679  016702  104404                  TRAP    C$BSEG
```

```
 4680  016704  004737  003240                  JSR     PC,SETBR4       ;SET THE BR4 BIT'
 4681  016710                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 4682  016710  004537  003100                  JSR     R5,..SROMCLK
 4683  016714  100400                          100400                  ;START AT ROM PC=0
 4684  016716                                  SROMCLK                 ;NEXT WORD IS INSTRUCION,
 4685  016716  004537  003100                  JSR     R5,..SROMCLK
```

```
 4686  016722  117377                    114377!<400*6>          ;JUMP TO ROM PC OF 1777
 4687  016724  004737  003330             JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
```

```
 4688  016730  000377                    377                     ;EXPECTED DATA
 4689  016732  120504                    CMPB    R5,R4           ;IS ROM PC CORRECT?
```

```
  4690  016734  001406                    BEQ     2$              ;BR IF YES
```

```
 4691  016736                          ERROR   5               ;ERROR, CRAM PC IS WRONG
 4692  016742  104455                  TRAP    C$ERDF
 4693  016744  000005                  .WORD   5
 4694  016746  005055                  .WORD   EM0
 4695  016750  006556                  .WORD   ERR5
```

```
4696  016752                          2$:     ESCAPE SEG
4697  016752  104410                          TRAP    C$ESCAPE
4698  016754  000002                          .WORD   10000$-.
4699  016756                                  ENDSEG
4700  016756                          10000$:
4701  016756  104405                          TRAP    C$ESEG
4702  016760                                  BGNSEG
4703  016760  104404                          TRAP    C$BSEG
4704  016762  004737  003240                  JSR     PC,SETBR4               ;SET THE BR4 BIT'
4705  016766                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4706  016766  004537  003100                  JSR     R5,.SROMCLK
4707  016772  100403                          100403                  ;START AT ROM PC=3
4708  016774                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4709  016774  004537  003100                  JSR     R5,.SROMCLK
4710  017000  103000                          100000!<400*6>          ;JUMP TO ROM PC OF 0
4711  017002  004737  003330                  JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4712  017006  000000                          0                       ;EXPECTED DATA
4713  017010  120504                          CMPB    R5,R4           ;IS ROM PC CORRECT?
4714  017012  001406                          BEQ     4$              ;BR IF YES
4715  017014                                  ERROR   5               ;ERROR, CRAM PC IS WRONG
4716  017020  104455                          TRAP    C$ERDF
4717  017022  000005                          .WORD   5
4718  017024  005055                          .WORD   EM0
4719  017026  006556                          .WORD   ERR5
4720  017030                          4$:     ESCAPE SEG
4721  017030  104410                          TRAP    C$ESCAPE
4722  017032  000002                          .WORD   10001$-.
4723  017034                                  ENDSEG
4724  017034                          10001$:
4725  017034  104405                          TRAP    C$ESEG
4726  017036                                  BGNSEG
4727  017036  104404                          TRAP    C$BSEG
4728  017040  004737  003240                  JSR     PC,SETBR4               ;SET THE BR4 BIT'
4729  017044                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4730  017044  004537  003100                  JSR     R5,.SROMCLK
4731  017050  100406                          100406                  ;START AT ROM PC=6
4732  017052                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
4733  017052  004537  003100                  JSR     R5,.SROMCLK
4734  017056  107125                          104125!<400*6>          ;JUMP TO ROM PC OF 525
4735  017060  004737  003330                  JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
4736  017064  000125                          125                     ;EXPECTED DATA
4737  017066  120504                          CMPB    R5,R4           ;IS ROM PC CORRECT?
4738  017070  001406                          BEQ     6$              ;BR IF YES
4739  017072                                  ERROR   5               ;ERROR, CRAM PC IS WRONG
4740  017076  104455                          TRAP    C$ERDF
4741  017100  000005                          .WORD   5
4742  017102  005055                          .WORD   EM0
4743  017104  006556                          .WORD   ERR5
4744  017106                          6$:     ESCAPE SEG
4745  017106  104410                          TRAP    C$ESCAPE
4746  017110  000002                          .WORD   10002$-.
4747  017112                                  ENDSEG
4748  017112                          10002$:
4749  017112  104405                          TRAP    C$ESEG
4750  017114                          ENDTST
4751  017114                          L10063:
```

```
 4752  017114  104401                        TRAP    C$ETST
 4753
 4754  017116                                BADHEAD
 4755                                        ;************** TEST 18 ***************
 4756                                        ;*CRAM TEST OF JUMP(I) ON BR7 SET MICRO-PROCESSOR INSTRUCTION.
 4757                                        ;*SET THE BR7 BIT, PERFORM THE JUMP INSTRUCTION.
 4758                                        ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
 4759                                        ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
 4760                                        ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
 4761                                        ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
 4762                                        ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
 4763                                        ;*THEN PORT4 WILL CONTAIN A 37
 4764  017116                                BADHEAD
 4765                                        ;************** TEST 18 ***************
 4766
 4767  017116                          BGNTST
 4768  017116                          T18::
 4769  017116                                MACEX2                  ;DON'T DO IF M8200.
 4770                                        ;DO NOT DO TEST IF M8200
 4771  017126  104432                        TRAP    C$EXIT
 4772  017130  000230                        .WORD   L10064-.
 4773  017132                                MYINT
 4774  017132  013701  002516                MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
 4775                                                                  ;R1 CONTAINS BASE M8200,4,7 ADDRESS
 4776  017136                                MSTCLR                  ;MASTER CLEAR M8200,4,7
 4777  017142  004737  003474                JSR     PC,MEMSET       ;SET MEM AND RAM
 4778  017146                          1$:   BGNSEG
 4779  017146  104404                        TRAP    C$BSEG
 4780  017150  004737  003250                JSR     PC,SETBR7       ;SET THE BR7 BIT'
 4781  017154                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 4782  017154  004537  003100                JSR     R5,..SROMCLK
 4783  017160  100400                        100400                  ;START AT ROM PC=0
 4784  017162                                SROMCLK                 ;NEXT WORD IS INSTRUCION,
 4785  017162  004537  003100                JSR     R5,..SROMCLK
 4786  017166  117777                        114377!<400*7>          ;JUMP TO ROM PC OF 1777
 4787  017170  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
 4788  017174  000377                        377                     ;EXPECTED DATA
 4789  017176  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
 4790  017200  001406                        BEQ     2$              ;BR IF YES
 4791  017202                                ERROR   5               ;ERROR, CRAM PC IS WRONG
 4792  017206  104455                        TRAP    C$ERDF
 4793  017210  000005                        .WORD   5
 4794  017212  005055                        .WORD   EMO
 4795  017214  006556                        .WORD   ERR5
 4796  017216                          2$:   ESCAPE SEG
 4797  017216  104410                        TRAP    C$ESCAPE
 4798  017220  000002                        .WORD   10000$-.
 4799  017222                                ENDSEG
 4800  017222                          10000$:
 4801  017222  104405                        TRAP    C$ESEG
 4802  017224                                BGNSEG
 4803  017224  104404                        TRAP    C$BSEG
 4804  017226  004737  003250                JSR     PC,SETBR7               ;SET THE BR7 BIT'
 4805  017232                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 4806  017232  004537  003100                JSR     R5,..SROMCLK
 4807  017236  100403                        100403                  ;START AT ROM PC=3
```

```
 4808  017240                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 4809  017240  004537  003100          JSR     R5,.SROMCLK
 4810  017244  103400                  100000!<400*7>          ;JUMP TO ROM PC OF 0
 4811  017246  004737  003330          JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
 4812  017252  000000                  0                       ;EXPECTED DATA
 4813  017254  120504                  CMPB    R5,R4           ;IS ROM PC CORRECT?
 4814  017256  001406                  BEQ     4$              ;BR IF YES
 4815  017260                          ERROR   5               ;ERROR, CRAM PC IS WRONG
 4816  017264  104455                  TRAP    C$ERDF
 4817  017266  000005                  .WORD   5
 4818  017270  005055                  .WORD   EM0
 4819  017272  006556                  .WORD   ERR5
 4820  017274                  4$:     ESCAPE SEG
 4821  017274  104410                  TRAP    C$ESCAPE
 4822  017276  000002                  .WORD   10001$-.
 4823  017300                          ENDSEG
 4824  017300                  10001$:
 4825  017300  104405                  TRAP    C$ESEG
 4826  017302                          BGNSEG
 4827  017302  104404                  TRAP    C$BSEG
 4828  017304  004737  003250          JSR     PC,SETBR7               ;SET THE BR7 BIT'
 4829  017310                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 4830  017310  004537  003100          JSR     R5,.SROMCLK
 4831  017314  100406                  100406                  ;START AT ROM PC=6
 4832  017316                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 4833  017316  004537  003100          JSR     R5,.SROMCLK
 4834  017322  107525                  104125!<400*7>          ;JUMP TO ROM PC OF 525
 4835  017324  004737  003330          JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
 4836  017330  000125                  125                     ;EXPECTED DATA
 4837  017332  120504                  CMPB    R5,R4           ;IS ROM PC CORRECT?
 4838  017334  001406                  BEQ     6$              ;BR IF YES
 4839  017336                          ERROR   5               ;ERROR, CRAM PC IS WRONG
 4840  017342  104455                  TRAP    C$ERDF
 4841  017344  000005                  .WORD   5
 4842  017346  005055                  .WORD   EM0
 4843  017350  006556                  .WORD   ERR5
 4844  017352                  6$:     ESCAPE SEG
 4845  017352  104410                  TRAP    C$ESCAPE
 4846  017354  000002                  .WORD   10002$-.
 4847  017356                          ENDSEG
 4848  017356                  10002$:
 4849  017356  104405                  TRAP    C$ESEG
 4850  017360                  ENDTST
 4851  017360                  L10064:
 4852  017360  104401                  TRAP    C$ETST
 4853
 4854  017362                          BADHEAD
 4855                                  ;*************** TEST 19 ****************
 4856                                  ;*CRAM TEST OF JUMP(I) ON C BIT CLEAR MICRO-PROCESSOR INSTRUCTION.
 4857                                  ;*SET THE C BIT, PERFORM THE JUMP INSTRUCTION.
 4858                                  ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
 4859                                  ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
 4860                                  ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
 4861                                  ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
 4862                                  ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
 4863                                  ;*THEN PORT4 WILL CONTAIN A 37
```

```
 4864  017362                                  BADHEAD
 4865                                          ;*************** TEST 19 ***************
 4866
 4867  017362                            BGNTST
 4868  017362                            T19::
 4869  017362                                  MACEX2                          ;DON'T DO IF M8200.
 4870                                          ;DO NOT DO TEST IF M8200
 4871  017372  104432                          TRAP    C$EXIT
 4872  017374  000244                          .WORD   L10065-.
 4873  017376                                  MYINT
 4874  017376  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
 4875  017402                                  MSTCLR                          ;MASTER CLEAR M8200,4,7
 4876  017406  004737  003474                  JSR     PC,MEMSET               ;SET MEM AND RAM
 4877  017412                            1$:   BGNSEG
 4878  017412  104404                          TRAP    C$BSEG
 4879  017414  004737  003260                  JSR     PC,SETC
 4880  017420  004737  003166                  JSR     PC,CLRALL
 4881  017424                                  SROMCLK                         ;NEXT WORD IS INSTRUCTION,
 4882  017424  004537  003100                  JSR     R5,.SROMCLK
 4883  017430  100400                          100400                          ;START AT ROM PC=0
 4884  017432                                  SROMCLK                         ;NEXT WORD IS INSTRUCION,
 4885  017432  004537  003100                  JSR     R5,.SROMCLK
 4886  017436  115377                          114377!<400*2>                  ;JUMP TO ROM PC OF 1777
 4887  017440  004737  003330                  JSR     PC,RAMDAT               ;R4=CRAM PC (LSB 8 BITS)
 4888  017444  000001                          1                               ;EXPECTED DATA
 4889  017446  120504                          CMPB    R5,R4                   ;IS ROM PC CORRECT?
 4890  017450  001406                          BEQ     2$                      ;BR IF YES
 4891  017452                                  ERROR   5                       ;ERROR, CRAM PC IS WRONG
 4892  017456  104455                          TRAP    C$ERDF
 4893  017460  000005                          .WORD   5
 4894  017462  005055                          .WORD   EMO
 4895  017464  006556                          .WORD   ERR5
 4896  017466                            2$:   ESCAPE SEG
 4897  017466  104410                          TRAP    C$ESCAPE
 4898  017470  000002                          .WORD   10000$-.
 4899  017472                                  ENDSEG
 4900  017472                            10000$:
 4901  017472  104405                          TRAP    C$ESEG
 4902  017474                                  BGNSEG
 4903  017474  104404                          TRAP    C$BSEG
 4904  017476                                  SKIP06  6$
 4905                                          ;GOTO 6$ IF M8206
 4906  017506  004737  003166                  JSR     PC,CLRALL               ;CLEAR ALL CONDITIONS
 4907  017512                                  SROMCLK                         ;NEXT WORD OF INSTRUCTION
 4908  017512  004537  003100                  JSR     R5,.SROMCLK
 4909  017516  100403                          100403                          ;START AT ROM PC=3
 4910  017520                                  SROMCLK                         ;NEXT WORD OF INSTRUCTION
 4911  017520  004537  003100                  JSR     R5,.SROMCLK
 4912  017524  101000                          100000!<400*2>                  ;JUMP TO ROM PC OF 0
 4913  017526  004737  003330                  JSR     PC,RAMDAT               ;R4=CRAM PC(LSB 8 BITS)
 4914  017532  000004                          4                               ;EXPECTED DATA
 4915  017534  120504                          CMPB    R5,R4                   ;IS ROM PC CORRECT?
 4916  017536  001406                          BEQ     4$                      ;BR IF YES
 4917  017540                                  ERROR   5                       ;ERROR, CRAM PC IS WRONG
 4918  017544  104455                          TRAP    C$ERDF
 4919  017546  000005                          .WORD   5
```

```
 4920  017550  005055                        .WORD   EM0
 4921  017552  006556                        .WORD   ERR5
 4922  017554                        4$:     ESCAPE SEG
 4923  017554  104410                        TRAP    C$ESCAPE
 4924  017556  000002                        .WORD   10001$-.
 4925  017560                                ENDSEG
 4926  017560                        10001$:
 4927  017560  104405                        TRAP    C$ESEG
 4928  017562                                BGNSEG
 4929  017562  104404                        TRAP    C$BSEG
 4930  017564  004737  003166                JSR     PC,CLRALL               ;CLEAR ALL CONDITIONS
 4931  017570                                SROMCLK                         ;NEXT WORD IS INSTRUCTION,
 4932  017570  004537  003100                JSR     R5,.SROMCLK
 4933  017574  100406                        100406                          ;START AT ROM PC=6
 4934  017576                                SROMCLK                         ;NEXT WORD IS INSTRUCTION,
 4935  017576  004537  003100                JSR     R5,.SROMCLK
 4936  017602  105125                        104125!<400*2>                  ;JUMP TO ROM PC OF 525
 4937  017604  004737  003330                JSR     PC,RAMDAT               ;R4=CRAM PC (LSB 8 BITS)
 4938  017610  000007                        7                               ;EXPECTED DATA
 4939  017612  120504                        CMPB    R5,R4                   ;IS ROM PC CORRECT?
 4940  017614  001406                        BEQ     6$                      ;BR IF YES
 4941  017616                                ERROR   5                       ;ERROR, CRAM PC IS WRONG
 4942  017622  104455                        TRAP    C$ERDF
 4943  017624  000005                        .WORD   5
 4944  017626  005055                        .WORD   EM0
 4945  017630  006556                        .WORD   ERR5
 4946  017632                        6$:     ESCAPE SEG
 4947  017632  104410                        TRAP    C$ESCAPE
 4948  017634  000002                        .WORD   10002$-.
 4949  017636                                ENDSEG
 4950  017636                        10002$:
 4951  017636  104405                        TRAP    C$ESEG
 4952  017640                        ENDTST
 4953  017640                        L10065:
 4954  017640  104401                        TRAP    C$ETST
 4955
 4956  017642                                BADHEAD
 4957                                        ;*************** TEST 20 ***************
 4958                                        ;*CRAM TEST OF JUMP(I) ON Z BIT CLEAR MICRO-PROCESSOR INSTRUCTION.
 4959                                        ;*CLEAR THE Z BIT, PERFORM THE JUMP INSTRUCTION.
 4960                                        ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
 4961                                        ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
 4962                                        ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
 4963                                        ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
 4964                                        ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
 4965                                        ;*THEN PORT4 WILL CONTAIN A 37
 4966  017642                                BADHEAD
 4967                                        ;*************** TEST 20 ***************
 4968
 4969  017642                        BGNTST
 4970  017642                        T20::
 4971  017642                                MACEX2                          ;DON'T DO IF M8200.
 4972                                        ;DO NOT DO TEST IF M8200
 4973  017652  104432                        TRAP    C$EXIT
 4974  017654  000244                        .WORD   L10066-.
 4975  017656                                MYINT
```

```
 4976  017656  013701  002516                 MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
 4977  017662                                 MSTCLR                          ;MASTER CLEAR M8200,4,7
 4978  017666  004737  003474                 JSR     PC,MEMSET               ;SET MEM AND RAM
 4979  017672                           1$:   BGNSEG
 4980  017672  104404                         TRAP    C$BSEG
 4981  017674  004737  003312                 JSR     PC,SETZ
 4982  017700  004737  003166                 JSR     PC,CLRALL               ; CLEAR CONDITION CODES ;*** B0
 4983  017704                                 SROMCLK                         ;NEXT WORD IS INSTRUCTION,
 4984  017704  004537  003100                 JSR     R5,.SROMCLK
 4985  017710  100400                         100400                          ;START AT ROM PC=0
 4986  017712                                 SROMCLK                         ;NEXT WORD IS INSTRUCION,
 4987  017712  004537  003100                 JSR     R5,.SROMCLK
 4988  017716  115777                         114377!<400*3>                  ;JUMP TO ROM PC OF 1777
 4989  017720  004737  003330                 JSR     PC,RAMDAT               ;R4=CRAM PC (LSB 8 BITS)
 4990  017724  000001                         1                               ;EXPECTED DATA
 4991  017726  120504                         CMPB    R5,R4                   ;IS ROM PC CORRECT?
 4992  017730  001406                         BEQ     2$                      ;BR IF YES
 4993  017732                                 ERROR   5                       ;ERROR, CRAM PC IS WRONG
 4994  017736  104455                         TRAP    C$ERDF
 4995  017740  000005                         .WORD   5
 4996  017742  005055                         .WORD   EM0
 4997  017744  006556                         .WORD   ERR5
 4998  017746                           2$:   ESCAPE SEG
 4999  017746  104410                         TRAP    C$ESCAPE
 5000  017750  000002                         .WORD   10000$-.
 5001  017752                                 ENDSEG
 5002  017752                           10000$:
 5003  017752  104405                         TRAP    C$ESEG
 5004  017754                                 BGNSEG
 5005  017754  104404                         TRAP    C$BSEG
 5006  017756                                 SKIP06  6$
 5007                                         ;GOTO 6$ IF M8206
 5008  017766  004737  003166                 JSR     PC,CLRALL               ;CLEAR ALL CONDITIONS
 5009  017772                                 SROMCLK                         ;NEXT WORD IS INSTRUCTION,
 5010  017772  004537  003100                 JSR     R5,.SROMCLK
 5011  017776  100403                         100403                          ;START AT ROM PC=3
 5012  020000                                 SROMCLK                         ;NEXT WORD IS INSTRUCTION,
 5013  020000  004537  003100                 JSR     R5,.SROMCLK
 5014  020004  101400                         100000!<400*3>                  ;JUMP TO ROM PC OF 0
 5015  020006  004737  003330                 JSR     PC,RAMDAT               ;R4=CRAM PC (LSB 8 BITS)
 5016  020012  000004                         4                               ;EXPECTED DATA
 5017  020014  120504                         CMPB    R5,R4                   ;IS ROM PC CORRECT?
 5018  020016  001406                         BEQ     4$                      ;BR IF YES
 5019  020020                                 ERROR   5                       ;ERROR, CRAM PC IS WRONG
 5020  020024  104455                         TRAP    C$ERDF
 5021  020026  000005                         .WORD   5
 5022  020030  005055                         .WORD   EM0
 5023  020032  006556                         .WORD   ERR5
 5024  020034                           4$:   ESCAPE SEG
 5025  020034  104410                         TRAP    C$ESCAPE
 5026  020036  000002                         .WORD   10001$-.
 5027  020040                                 ENDSEG
 5028  020040                           10001$:
 5029  020040  104405                         TRAP    C$ESEG
 5030  020042                                 BGNSEG
 5031  020042  104404                         TRAP    C$BSEG
```

```
5032  020044  004737  003166                  JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5033  020050                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5034  020050  004537  003100                  JSR     R5,.SROMCLK
5035  020054  100406                          100406                  ;START AT ROM PC=6
5036  020056                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5037  020056  004537  003100                  JSR     R5,.SROMCLK
5038  020062  105525                          104125!<400*3>          ;JUMP TO ROM PC OF 525
5039  020064  004737  003330                  JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5040  020070  000007                          7                       ;EXPECTED DATA
5041  020072  120504                          CMPB    R5,R4           ;IS ROM PC CORRECT?
5042  020074  001406                          BEQ     6$              ;BR IF YES
5043  020076                                  ERROR   5               ;ERROR, CRAM PC IS WRONG
5044  020102  104455                          TRAP    C$ERDF
5045  020104  000005                          .WORD   5
5046  020106  005055                          .WORD   EMO
5047  020110  006556                          .WORD   ERR5
5048  020112                          6$:     ESCAPE SEG
5049  020112  104410                          TRAP    C$ESCAPE
5050  020114  000002                          .WORD   10002$-.
5051  020116                                  ENDSEG
5052  020116                          10002$:
5053  020116  104405                          TRAP    C$ESEG
5054  020120                          ENDTST
5055  020120                          L10066:
5056  020120  104401                          TRAP    C$ETST
5057
5058  020122                                  BADHEAD
5059                                          ;*************** TEST 21 ***************
5060                                          ;*CRAM TEST OF JUMP(I) ON BRO CLEAR MICRO-PROCESSOR INSTRUCTION.
5061                                          ;*CLEAR THE BRO BIT, PERFORM THE JUMP INSTRUCTION.
5062                                          ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
5063                                          ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
5064                                          ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
5065                                          ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
5066                                          ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
5067                                          ;*THEN PORT4 WILL CONTAIN A 37
5068  020122                                  BADHEAD
5069                                          ;*************** TEST 21 ***************
5070
5071  020122                          BGNTST
5072  020122                          T21::
5073  020122                                  MACEX2                  ;DON'T DO IF M8200.
5074                                          ;DO NOT DO TEST IF M8200
5075  020132  104432                          TRAP    C$EXIT
5076  020134  000240                          .WORD   L10067-.
5077  020136                                  MYINT
5078  020136  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
5079  020142                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
5080  020146  004737  003474                  JSR     PC,MEMSET       ;SET MEM AND RAM
5081  020152                          1$:     BGNSEG
5082  020152  104404                          TRAP    C$BSEG
5083  020154  004737  003166                  JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5084  020160                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5085  020160  004537  003100                  JSR     R5,.SROMCLK
5086  020164  100400                          100400                  ;START AT ROM PC=0
5087  020166                                  SROMCLK                 ;NEXT WORD IS INSTRUCION,
```

```
5088  020166  004537  003100                JSR     R5,.SROMCLK
5089  020172  116377                        114377!<400*4>          ;JUMP TO ROM PC OF 1777
5090  020174  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5091  020200  000001                        1                       ;EXPECTED DATA
5092  020202  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
5093  020204  001406                        BEQ     2$              ;BR IF YES
5094  020206                                ERROR   5               ;ERROR, CRAM PC IS WRONG
5095  020212  104455                        TRAP    C$ERDF
5096  020214  000005                        .WORD   5
5097  020216  005055                        .WORD   EM0
5098  020220  006556                        .WORD   ERR5
5099  020222                        2$:     ESCAPE SEG
5100  020222  104410                        TRAP    C$ESCAPE
5101  020224  000002                        .WORD   10000$-.
5102  020226                                ENDSEG
5103  020226                        10000$:
5104  020226  104405                        TRAP    C$ESEG
5105  020230                                BGNSEG
5106  020230  104404                        TRAP    C$BSEG
5107  020232                                SKIP06  6$
5108                                        ;GOTO 6$ IF M8206
5109  020242  004737  003166                JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5110  020246                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5111  020246  004537  003100                JSR     R5,.SROMCLK
5112  020252  100403                        100403                  ;START AT ROM PC=3
5113  020254                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5114  020254  004537  003100                JSR     R5,.SROMCLK
5115  020260  102000                        100000!<400*4>          ;JUMP TO ROM PC OF 0
5116  020262  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5117  020266  000004                        4                       ;EXPECTED DATA
5118  020270  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
5119  020272  001406                        BEQ     4$              ;BR IF YES
5120  020274                                ERROR   5               ;ERROR, CRAM PC IS WRONG
5121  020300  104455                        TRAP    C$ERDF
5122  020302  000005                        .WORD   5
5123  020304  005055                        .WORD   EM0
5124  020306  006556                        .WORD   ERR5
5125  020310                        4$:     ESCAPE SEG
5126  020310  104410                        TRAP    C$ESCAPE
5127  020312  000002                        .WORD   10001$-.
5128  020314                                ENDSEG
5129  020314                        10001$:
5130  020314  104405                        TRAP    C$ESEG
5131  020316                                BGNSEG
5132  020316  104404                        TRAP    C$BSEG
5133  020320  004737  003166                JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5134  020324                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5135  020324  004537  003100                JSR     R5,.SROMCLK
5136  020330  100406                        100406                  ;START AT ROM PC=6
5137  020332                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5138  020332  004537  003100                JSR     R5,.SROMCLK
5139  020336  106125                        104125!<400*4>          ;JUMP TO ROM PC OF 525
5140  020340  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5141  020344  000007                        7                       ;EXPECTED DATA
5142  020346  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
5143  020350  001406                        BEQ     6$              ;BR IF YES
```

```
5144  020352                                ERROR   5               ;ERROR, CRAM PC IS WRONG
5145  020356  104455                        TRAP    C$ERDF
5146  020360  000005                        .WORD   5
5147  020362  005055                        .WORD   EM0
5148  020364  006556                        .WORD   ERR5
5149  020366                          6$:   ESCAPE SEG
5150  020366  104410                        TRAP    C$ESCAPE
5151  020370  000002                        .WORD   10002$-.
5152  020372                                ENDSEG
5153  020372                          10002$:
5154  020372  104405                        TRAP    C$ESEG
5155  020374                          ENDTST
5156  020374                          L10067:
5157  020374  104401                        TRAP    C$ETST
5158
5159  020376                                BADHEAD
5160                                        ;*************** TEST 22 ***************
5161                                        ;*CRAM TEST OF JUMP(I) ON BR1 CLEAR MICRO-PROCESSOR INSTRUCTION.
5162                                        ;*CLEAR THE BR1 BIT, PERFORM THE JUMP INSTRUCTION.
5163                                        ;*VERIFY THE JUMP DID OCCUR BY CLOCKING THE INSTRUCTION
5164                                        ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
5165                                        ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
5166                                        ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
5167                                        ;*THE JUMP WAS SUCCESSFUL, IF THE JUMP WAS UNSUCCESSFUL
5168                                        ;*THEN PORT4 WILL CONTAIN A 37
5169  020376                                BADHEAD
5170                                        ;*************** TEST 22 ***************
5171
5172  020376                          BGNTST
5173  020376                          T22::
5174  020376                                MACEX2                  ;DON'T DO IF M8200.
5175                                        ;DO NOT DO TEST IF M8200
5176  020406  104432                        TRAP    C$EXIT
5177  020410  000240                        .WORD   L10070-.
5178  020412                                MYINT
5179  020412  013701  002516                MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
5180  020416                                MSTCLR                  ;MASTER CLEAR M8200,4,7
5181  020422  004737  003474                JSR     PC,MEMSET       ;SET MEM AND RAM
5182  020426                          1$:   BGNSEG
5183  020426  104404                        TRAP    C$BSEG
5184  020430  004737  003166                JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5185  020434                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5186  020434  004537  003100                JSR     R5,.SROMCLK
5187  020440  100400                        100400                  ;START AT ROM PC=0
5188  020442                                SROMCLK                 ;NEXT WORD IS INSTRUCION,
5189  020442  004537  003100                JSR     R5,.SROMCLK
5190  020446  116777                        114377!<400*5>          ;JUMP TO ROM PC OF 1777
5191  020450  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5192  020454  000001                        1                       ;EXPECTED DATA
5193  020456  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
5194  020460  001406                        BEQ     2$              ;BR IF YES
5195  020462                                ERROR   5               ;ERROR, CRAM PC IS WRONG
5196  020466  104455                        TRAP    C$ERDF
5197  020470  000005                        .WORD   5
5198  020472  005055                        .WORD   EM0
5199  020474  006556                        .WORD   ERR5
```

```
5200  020476                          2$:     ESCAPE SEG
5201  020476  104410                          TRAP    C$ESCAPE
5202  020500  000002                          .WORD   10000$-.
5203  020502                                  ENDSEG
5204  020502                          10000$:
5205  020502  104405                          TRAP    C$ESEG
5206  020504                                  BGNSEG
5207  020504  104404                          TRAP    C$BSEG
5208  020506                                  SKIP06  6$
5209                                          ;GOTO 6$ IF M8206
5210  020516  004737  003166                  JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5211  020522                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5212  020522  004537  003100                  JSR     R5,.SROMCLK
5213  020526  100403                          100403                  ;START AT ROM PC=3
5214  020530                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5215  020530  004537  003100                  JSR     R5,.SROMCLK
5216  020534  102400                          100000!<400*5>          ;JUMP TO ROM PC OF 0
5217  020536  004737  003330                  JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5218  020542  000004                          4                       ;EXPECTED DATA
5219  020544  120504                          CMPB    R5,R4           ;IS ROM PC CORRECT?
5220  020546  001406                          BEQ     4$              ;BR IF YES
5221  020550                                  ERROR   5               ;ERROR, CRAM PC IS WRONG
5222  020554  104455                          TRAP    C$ERDF
5223  020556  000005                          .WORD   5
5224  020560  005055                          .WORD   EM0
5225  020562  006556                          .WORD   ERR5
5226  020564                          4$:     ESCAPE SEG
5227  020564  104410                          TRAP    C$ESCAPE
5228  020566  000002                          .WORD   10001$-.
5229  020570                                  ENDSEG
5230  020570                          10001$:
5231  020570  104405                          TRAP    C$ESEG
5232  020572                                  BGNSEG
5233  020572  104404                          TRAP    C$BSEG
5234  020574  004737  003166                  JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5235  020600                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5236  020600  004537  003100                  JSR     R5,.SROMCLK
5237  020604  100406                          100406                  ;START AT ROM PC=6
5238  020606                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5239  020606  004537  003100                  JSR     R5,.SROMCLK
5240  020612  106525                          104125!<400*5>          ;JUMP TO ROM PC OF 525
5241  020614  004737  003330                  JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5242  020620  000007                          7                       ;EXPECTED DATA
5243  020622  120504                          CMPB    R5,R4           ;IS ROM PC CORRECT?
5244  020624  001406                          BEQ     6$              ;BR IF YES
5245  020626                                  ERROR   5               ;ERROR, CRAM PC IS WRONG
5246  020632  104455                          TRAP    C$ERDF
5247  020634  000005                          .WORD   5
5248  020636  005055                          .WORD   EM0
5249  020640  006556                          .WORD   ERR5
5250  020642                          6$:     ESCAPE SEG
5251  020642  104410                          TRAP    C$ESCAPE
5252  020644  000002                          .WORD   10002$-.
5253  020646                                  ENDSEG
5254  020646                          10002$:
5255  020646  104405                          TRAP    C$ESEG
```

```
5256  020650                          ENDTST
5257  020650                          L10070:
5258  020650  104401                          TRAP    C$ETST
5259
5260  020652                                  BADHEAD
5261                                          ;************** TEST 23 ***************
5262                                          ;*CRAM TEST OF JUMP(I) ON BR4 CLEAR MICRO-PROCESSOR INSTRUCTION.
5263                                          ;*CLEAR THE BR4 BIT, PERFORM THE JUMP INSTRUCTION.
5264                                          ;*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
5265                                          ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
5266                                          ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
5267                                          ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
5268  020652  020652                  .
5269                                          ;*THE CRAM PC IS CORRECT, IF THE CRAM PC IS NOT RIGHT.
5270                                          ;*THEN PORT4 CONTAINS A 37
5271  020654                                  BADHEAD
5272                                          ;************** TEST 23 ***************
5273
```

```
 5274  020654                          BGNTST
 5275  020654                          T23::
 5276  020654                                  MACEX2                  ;DON'T DO IF M8200.
 5277                                          ;DO NOT DO TEST IF M8200
 5278  020664  104432                          TRAP    C$EXIT
 5279  020666  000240                          .WORD   L10071-.
 5280  020670                                  MYINT
 5281  020670  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
 5282  020674                                  MSTCLR                  ;MASTER CLEAR M8200,4,7
 5283  020700  004737  003474                  JSR     PC,MEMSET       ;SET MEM AND RAM
 5284  020704                          1$:     BGNSEG
 5285  020704  104404                          TRAP    C$BSEG
 5286  020706  004737  003166                  JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
 5287  020712                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 5288  020712  004537  003100                  JSR     R5,.SROMCLK
 5289  020716  100400                          100400                  ;START AT ROM PC=0
 5290  020720                                  SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 5291  020720  004537  003100                  JSR     R5,.SROMCLK
 5292  020724  117377                          114377!<400*6>          ;JUMP TO ROM PC OF 1777
 5293  020726  004737  003330                  JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
```

```
5294  020732  000001                  1                       ;EXPECTED DATA
5295  020734  120504                  CMPB    R5,R4           ;IS ROM PC CORRECT?
5296  020736  001406                  BEQ     2$              ;BR IF YES
5297  020740                          ERROR   5               ;ERROR, CRAM PC IS WRONG
5298  020744  104455                  TRAP    C$ERDF
5299  020746  000005                  .WORD   5
5300  020750  005055                  .WORD   EM0
5301  020752  006556                  .WORD   ERR5
5302  020754                  2$:     ESCAPE SEG
5303  020754  104410                  TRAP    C$ESCAPE
5304  020756  000002                  .WORD   10000$-.
5305  020760                          ENDSEG
5306  020760                  10000$:
5307  020760  104405                  TRAP    C$ESEG
5308  020762                          BGNSEG
5309  020762  104404                  TRAP    C$BSEG
5310  020764                          SKIP06  6$
5311                                  ;GOTO 6$ IF M8206
5312  020774  004737  003166          JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5313  021000                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5314  021000  004537  003100          JSR     R5,.SROMCLK
5315  021004  100403                  100403                  ;START AT ROM PC=3
5316  021006                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5317  021006  004537  003100          JSR     R5,.SROMCLK
5318  021012  103000                  100000!<400*6>  ;JUMP TO ROM PC OF 0
5319  021014  004737  003330          JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5320  021020  000004                  4                       ;EXPECTED DATA
```

```
 5321  021022  120504                    CMPB    R5,R4           ;IS ROM PC CORRECT?
 5322  021024  001406                    BEQ     4$              ;BSR IF YES
```

```
 5323  021026                          ERROR   5               ;ERROR, CRAM PC IS WRONG
 5324  021032  104455                  TRAP    C$ERDF
 5325  021034  000005                  .WORD   5
 5326  021036  005055                  .WORD   EM0
 5327  021040  006556                  .WORD   ERR5
```

```
5328  021042
5329  021042  104410                    4$:     ESCAPE SEG
5330  021044  000002                            TRAP    C$ESCAPE
5331  021046                                    .WORD   10001$-.
5332  021046                                    ENDSEG
5333  021046  104405                    10001$:
5334  021050                                    TRAP    C$ESEG
5335  021050  104404                            BGNSEG
5336  021052  004737  003166                    TRAP    C$BSEG
5337  021056                                    JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
5338  021056  004537  003100                    SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5339  021062  100406                            JSR     R5,.SROMCLK
5340  021064                                    100406                  ;START AT ROM PC=6
5341  021064  004537  003100                    SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5342  021070  107125                            JSR     R5,.SROMCLK
5343  021072  004737  003330                    104125!<400*6>  ;JUMP TO ROM PC OF 525
5344  021076  000007                            JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5345  021100  120504                            7                       ;EXPECTED DATA
5346  021102  001406                            CMPB    R5,R4           ;IS ROM PC CORRECT?
5347  021104                                    BEQ     6$              ;BR IF YES
5348  021110  104455                            ERROR   5               ;ERROR, CRAM PC IS WRONG
5349  021112  000005                            TRAP    C$ERDF
5350  021114  005055                            .WORD   5
5351  021116  006556                            .WORD   EM0
5352  021120                                    .WORD   ERR5
5353  021120  104410                    6$:     ESCAPE SEG
5354  021122  000002                            TRAP    C$ESCAPE
5355  021124                                    .WORD   10002$-.
5356  021124                                    ENDSEG
5357  021124  104405                    10002$:
5358  021126                                    TRAP    C$ESEG
5359  021126                            ENDTST
5360  021126  104401                    L10071:
5361                                            TRAP    C$ETST
5362  021130                                    BADHEAD
5363                                            ;************** TEST 24 ***************
5364                                            ;*CRAM TEST OF JUMP(I) ON BR7 CLEAR MICRO-PROCESSOR INSTRUCTION.
5365                                            ;*CLEAR THE BR7 BIT, PERFORM THE JUMP INSTRUCTION.
5366                                            ;*VERIFY THE JUMP DID NOT OCCUR BY CLOCKING THE INSTRUCTION
5367                                            ;*IN THE LOCATION IT IS AT. THIS INSTRUCTION LOADS THE
5368                                            ;*BR WITH THE LOWEST 8 BITS OF THE CRAM PC. AT THIS POINT
5369                                            ;*THE BR DATA IS MOVED TO PORT4. IF THIS DATA IS CORRECT
5370                                            ;*THE CRAM PC IS CORRECT, IF THE CRAM PC IS NOT RIGHT.
5371                                            ;*THEN PORT4 CONTAINS A 37
5372  021130                                    BADHEAD
5373                                            ;************** TEST 24 ***************
5374
5375  021130                            BGNTST
5376  021130                            T24::
5377  021130                                    MACEX2                  ;DON'T DO IF M8200.
5378                                            ;DO NOT DO TEST IF M8200
5379  021140  104432                            TRAP    C$EXIT
5380  021142  000240                            .WORD   L10072-.
5381  021144                                    MYINT
5382  021144  013701  002516                    MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
5383  021150                                    MSTCLR                  ;MASTER CLEAR M8200,4,7
```

```
 5384  021154  004737  003474          JSR     PC,MEMSET       ;SET MEM AND RAM
 5385  021160                  1$:     BGNSEG
 5386  021160  104404                  TRAP    C$BSEG
 5387  021162  004737  003166          JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
 5388  021166                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 5389  021166  004537  003100          JSR     R5,.SROMCLK
 5390  021172  100400                  100400          ;START AT ROM PC=0
 5391  021174                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 5392  021174  004537  003100          JSR     R5,.SROMCLK
 5393  021200  117777                  114377!<400*7>          ;JUMP TO ROM PC OF 1777
 5394  021202  004737  003330          JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
 5395  021206  000001                  1                       ;EXPECTED DATA
 5396  021210  120504                  CMPB    R5,R4           ;IS ROM PC CORRECT?
 5397  021212  001406                  BEQ     2$              ;BR IF YES
 5398  021214                          ERROR   5               ;ERROR, CRAM PC IS WRONG
 5399  021220  104455                  TRAP    C$ERDF
 5400  021222  000005                  .WORD   5
 5401  021224  005055                  .WORD   EMO
 5402  021226  006556                  .WORD   ERR5
 5403  021230                  2$:     ESCAPE SEG
 5404  021230  104410                  TRAP    C$ESCAPE
 5405  021232  000002                  .WORD   10000$-.
 5406  021234                          ENDSEG
 5407  021234                  10000$:
 5408  021234  104405                  TRAP    C$ESEG
 5409  021236                          BGNSEG
 5410  021236  104404                  TRAP    C$BSEG
 5411  021240                          SKIP06  6$
 5412                                  ;GOTO 6$ IF M8206
 5413  021250  004737  003166          JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
 5414  021254                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 5415  021254  004537  003100          JSR     R5,.SROMCLK
 5416  021260  100403                  100403                  ;START AT ROM PC=3
 5417  021262                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 5418  021262  004537  003100          JSR     R5,.SROMCLK
 5419  021266  103400                  100000!<400*7>  ;JUMP TO ROM PC OF 0
 5420  021270  004737  003330          JSR     PC,RAMDAT       ;;R4=CRAM PC (LCB 8 BITS)
 5421  021274  000004                  4                       ;EXPECTED DATA
 5422  021276  120504                  CMPB    R5,R4           ;IS ROM PC CORRECT?
 5423  021300  001406                  BEQ     4$              ;BR IF YES
 5424  021302                          ERROR   5               ;ERROR, CRAM PC IS WRONG
 5425  021306  104455                  TRAP    C$ERDF
 5426  021310  000005                  .WORD   5
 5427  021312  005055                  .WORD   EMO
 5428  021314  006556                  .WORD   ERR5
 5429  021316                  4$:     ESCAPE SEG
 5430  021316  104410                  TRAP    C$ESCAPE
 5431  021320  000002                  .WORD   10001$-.
 5432  021322                          ENDSEG
 5433  021322                  10001$:
 5434  021322  104405                  TRAP    C$ESEG
 5435  021324                          BGNSEG
 5436  021324  104404                  TRAP    C$BSEG
 5437  021326  004737  003166          JSR     PC,CLRALL       ;CLEAR ALL CONDITIONS
 5438  021332                          SROMCLK                 ;NEXT WORD IS INSTRUCTION,
 5439  021332  004537  003100          JSR     R5,.SROMCLK
```

```
5440  021336  100406                        100406                  ;START AT ROM PC=6
5441  021340                                SROMCLK                 ;NEXT WORD IS INSTRUCTION,
5442  021340  004537  003100                JSR     R5,.SROMCLK
5443  021344  107525                        104125!<400*7>          ;JUMP TO ROM PC OF 525
5444  021346  004737  003330                JSR     PC,RAMDAT       ;R4=CRAM PC (LSB 8 BITS)
5445  021352  000007                        7                       ;EXPECTED DATA
5446  021354  120504                        CMPB    R5,R4           ;IS ROM PC CORRECT?
5447  021356  001406                        BEQ     6$              ;BR IF YES
5448  021360                                ERROR   5               ;ERROR, CRAM PC IS WRONG
5449  021364  104455                        TRAP    C$ERDF
5450  021366  000005                        .WORD   5
5451  021370  005055                        .WORD   EM0
5452  021372  006556                        .WORD   ERR5
5453  021374                        6$:     ESCAPE SEG
5454  021374  104410                        TRAP    C$ESCAPE
5455  021376  000002                        .WORD   10002$-.
5456  021400                                ENDSEG
5457  021400                        10002$:
5458  021400  104405                        TRAP    C$ESEG
5459  021402                        ENDTST
5460  021402                        L10072:
5461  021402  104401                        TRAP    C$ETST
5462
5463  021404                                BADHEAD
5464                                        ;*************** TEST 25 ***************
5465                                        ;*
5466                                        ;*MAIN MEMORY PAGE DUAL ADDRESS TEST.
5467                                        ;*IN THIS TEST WE WILL VERIFY THAT PAGES DO
5468                                        ;*NOT DUAL ADDRESS.  THIS TEST IS DIFFERENT FROM THE
5469                                        ;*PREVIOUS DUAL ADDRESS TESTS IN THAT THE OTHER
5470                                        ;*TEST REALLY DIDN'T CHECK PAGE DUAL ADDRESSING
5471  021404                                BADHEAD
5472                                        ;*************** TEST 25 ***************
5473
5474  021404                        BGNTST
5475  021404                        T25::
5476  021404                                K4ONLY                  ;FOR 4K CPUS ONLY.
5477                                        ;DO NOT DO TEST IF M8200, OR M8204
5478  021414  104432                        TRAP    C$EXIT
5479  021416  000156                        .WORD   L10073-.
5480  021420                                MYINT
5481  021420  013701  002516                MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
5482  021424                                MSTCLR
5483  021430  005002                        CLR R2                  ;R2 WILL BE PAGE #
5484  021432  042737  000037  021456  1$:   BIC #37,2$              ;CLEAR UNUSED BITS
5485  021440  050237  021456                BIS R2,2$               ;ADD CURRENT PAGE MARKER.
5486  021444                                ROMCLK                  ;SET ADDR D
5487  021444  004537  003044                JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
5488  021450  010000                        10000
5489  021452                                ROMCLK                  ;OF PAGE X
5490  021452  004537  003044                JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
5491  021456  004000                2$:     4000                    ;THIS LOCATION MODIFIED BY LOST
5492                                                                ;FEW INSTRUCTIONS
5493  021460  010261  000004                MOV R2,4(R1)            ;PUT PAGE # INTO PART 4
5494  021464                                ROMCLK                  ;CLOCK PART 4 INTO MEMORY
5495  021464  004537  003044                JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
```

```
5496  021470  122500                          122500                  ;WHOSE PAGE # IS IN R2
5497  021472  005202                          INC R2                  ;UPDATE PAGE #
5498  021474  032702  000020                  BIT #20,R2              ;DONE ALL PAGES?
5499  021500  001754                          BEQ 1$                  ;NO-DO NEXT ONE
5500
5501                                                                  ;OK, ALL LOADS OF ALL PAGES
5502                                                                  ;CONTAIN THIER PAGE NUMBER, NOW
5503                                                                  ;WE'LL GO BACK THROUGH
5504                                                                  ;THEM CHECKING THEM OUT.
5505
5506  021502  005002                          CLR R2                  ;R2 STILL HAS PAGE NUMBER
5507
5508  021504  042737  000037  021522  3$:     BIC #37,4$
5509  021512  050237  021522                  BIS R2,4$
5510  021516                                  ROMCLK                  ;LOAD PAGE NUMBER
5511  021516  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
5512  021522  004000                  4$:     4000
5513  021524                                  ROMCLK                  ;MOVE MEM TO PART 4
5514  021524  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
5515  021530  041224                          041224
5516  021532  116104  000004                  MOVB 4(R1),R4           ;''FOUND''
5517  021536  110205                          MOVB R2,R5              ;''EXPECTED''
5518  021540  120504                          CMPB R5,R4              ;ADDRESS PROBLEM?
5519  021542  001406                          BEQ 5$
5520
5521  021544                                  ERROR 13                ;PAGE ADDRESSING ERROR IN MAIN
5522  021550  104455                          TRAP    C$ERDF
5523  021552  000015                          .WORD   13
5524  021554  005055                          .WORD   EM0
5525  021556  007542                          .WORD   ERR13
5526                                                                  ;MEMORY.
5527                                                                  ;MAR BITS 8 THROUGH 12 ARE REPRESENTED
5528                                                                  ;BY R2 (''EXP'ED'')BITS 0-4)
5529
5530  021560                          5$:     ESCAPE TST
5531  021560  104410                          TRAP    C$ESCAPE
5532  021562  000012                          .WORD   L10073-.
5533  021564  005202                          INC R2                  ;UPDATE PAGE ADDRESS
5534  021566  032702  000020                  BIT #20,R2              ;ALL DONE?
5535  021572  001744                          BEQ 3$                  ;NO-CHECK NEXT PAGE.
5536                                                                  ;YES-EXIT.
5537  021574                          ENDTST
5538  021574                          L10073:
5539  021574  104401                          TRAP    C$ETST
5540
5541
5542  021576                                  BADHEAD
5543                                          ;*************** TEST 26 ***************
5544                                          ;*
5545                                          ;*JUMP FIELD,PAGE TEST
5546                                          ;*
5547                                          ;*IN THIS TEST WILL MAKE SURE A JUMP FIELD INSTRUCTION
5548                                          ;*WORKS.  TO DO THIS, WE'LL PUT THE DESIRED PAGE,FIELD
5549                                          ;*INORMATION IN IBUS*<13> THEN ISSUE A JUMP FIELD
5550                                          ;*THEN WE'LL READ PC REG. AND VERIFY.
5551  021576                                  BADHEAD
```

```
                                        ;*************** TEST 26 ***************
                                BGNTST
                                T26::
                                        K4ONLY                  ;FOR 4K CPUS ONLY
                                        ;DO NOT DO TEST IF M8200, OR M8204
021606  104432                          TRAP    C$EXIT
021610  000132                          .WORD   L10074-.
021612                                  MYINT
021612  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
021616                                  MSTCLR
                                
021622  005002                          CLR R2                  ;R2 TO CONTAIN FIELD #
                                
021624  042737  000017  021642  1$:     BIC #17,2$              ;CLEAR ANY JUNK
021632  050237  021642                  BIS R2,2$               ;SET FIELD # INTO INSTR.
                                
021636                                  ROMCLK                  ;CLOCK FIELD BITS INTO BREG.
021636  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
021642  000400                  2$:     000400                  ;CONTAINS FIELD,PAGE BITS
021644                                  ROMCLK                  ;XFERR BREG INTO IBUS*<13>
021644  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
021650  061233                          061233
021652                                  SROMCLK                 ;GET INSTRUCTION CLOCKED.
021652  004537  003100                  JSR     R5,.SROMCLK
021656  100000                          100000                  ;BAS FORM FOR JUM FIELD INSTR.
                                
                                
021660  142761  000002  000001          BICB #BIT1,1(R1)                ;CLEAR ROMI
021666                                  ROMCLK                  ;CLOCK NEXT INSTR.
021666  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
021672  121264                          121264                  ;MOVE IBUS*TO PORT 4
021674  116104  000004                  MOVB 4(R1),R4           ;GET IT.
021700  042704  177760                  BIC     #^C<17>,R4
021704  120402                          CMPB R4,R2              ;FIELD OK?
021706  001407                          BEQ 3$                  ;IF OK GO AHEAD
021710  010205                          MOV     R2,R5
021712                                  ERROR 14                ;CHANGE FIELD INSTRUCTION   ;*** BO
021716  104455                          TRAP    C$ERDF
021720  000016                          .WORD   14
021722  005055                          .WORD   EMO
021724  007664                          .WORD   ERR14
                                                                ;FAILED.  FOR FIELD,PAGE INDICATES
                                                                ;BY ''EXPECTED'' BITS 0,1,2,3 OF
                                                                ;EXPECTED REPRESENT FIELD BITS.
                                
021726                          3$:     ESCAPE TST
021726  104410                          TRAP    C$ESCAPE
021730  000012                          .WORD   L10074-.
                                
                                
021732  005202                          INC R2                  ;UPDATE TO NEXT FIELD
021734  032702  000020                  BIT #20,R2              ;DONE ALL FIELDS?
021740  001731                          BEQ     1$
                                
021742                          ENDTST
021742                          L10074:
```

```
5608  021742  104401                          TRAP    C$ETST
5609
5610  021744                                  BADHEAD
5611                                          ;************** TEST 27 ***************
5612                                          ;*
5613                                          ;*JUMP TEST, JUMP ALWAYS, JUMP CHANGE FIELD
5614                                          ;*
5615                                          ;*IN THIS TEST, WE WILL CHECK THE ABILITY OF THE
5616                                          ;*MICRO PROCESSOR TO JUMP (BRANCH & ALWAYS INSTRUCTION)
5617                                          ;*TO LOCATIONS, FIELDS FROM OTHER LOCATIONS FIELDS.
5618                                          ;*WE ALREADY KNOW THAT THE BRANCH INSTR WORKS FROM
5619                                          ;*OTHER TEST.              PROCEDURE:
5620                                          ;*      1. START ADDR 0, FIELD 0
5621                                          ;*      2. **CALCULATE NEW ADDRS, FIELD  VIA INC,
5622                                          ;*      3. CAUSE JUMP (BRANCH) TO NEW ADDRESS
5623                                          ;*      4. READ PC FROM IBUS*12 AND IBUS*13
5624                                          ;*      5. REPEAT STEP 2-4 256.TIMES
5625                                          ;*
5626                                          ;*      TO CALCULATE NEW ADDRESS:
5627                                          ;*      1. INC LOW BYTE OF ADDRESS FOR PC ADDRES 0-7
5628                                          ;*      2. INC LOW BYTE OF NADDRESS FOR PC ADDRESS 8-11
5629                                          ;*         BITS REPRESENTED AS BITS 0-3. WHEN  0-3 OVERFLOWS,
5630                                          ;*         RESTARTS AT ZERO.
5631                                          ;*         NET RESULT IS JUMPS FROM:
5632                                          ;*             FIELD,PAGE              LOC
5633                                          ;*                   0                 0
5634                                          ;*                   1                 1
5635                                          ;*                   2                 2
5636                                          ;*                   3                 3
5637                                          ;*                  10                 7
5638                                          ;*                  11                11
5639                                          ;*                  :TO                :
5640                                          ;*                  17                377
5641                                          ;*
5642  021744                                  BADHEAD
5643                                          ;************** TEST 27 ***************
5644
5645  021744                                  BGNTST
5646  021744                          T27::
5647  021744                                  MYINT
5648  021744  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
5649  021750                                  K4ONLY                          ;4K CPUS ONLY.
5650                                          ;DO NOT DO TEST IF M8200, OR M8204
5651  021760  104432                          TRAP    C$EXIT
5652  021762  000336                          .WORD   L10075-.
5653  021764                                  MSTCLR
5654
5655  021770  012737  000000  002406          MOV     #0, FLAG        ;FLAG TO REPRESENT
5656                                                                  ;FIELD,PAGE
5657                                                                  ;TO VARIE STARTING PAGE,FIELD,
5658                                                                  ;CHANGE #0 PORTION OF INSTR.
5659  021776  012702  000000                  MOV     #0, R2          ;R2 TO CONTAIN JUMPED
5660                                                                  ;TO CHANGE STARTING IMM ADDR.,
5661                                                                  ;VARIE #0 PORTIONS OF INSTR.
5662  022002  012737  000000  002412          MOV     #0, FADR        ;ADDRESS
5663                                          ;LOOP HERE
```

```
5664  022010                          1$:
5665  022010  042737  000017  022050        BIC #17,2$              ;CLEAR JUNK FROM FIELD
5666                                        ;PORTION OF CHANGE FIELD INSTR
5667  022016  013700  002406              MOV FLAG,R0             ;INORDER TO INC, DEC FIELD,PAGE
5668  022022  042700  177760              BIC #^C<17>,R0
5669  022026  050037  022050              BIS R0,2$               ;NOW POSITION IN INSTR.
5670  022032  042737  077777  022064      BIC #077777,3$                  ;NOW FOR IMMED. BR INSTR.
5671  022040  050237  022064              BIS R2,3$               ;NOW ADD IMMEDIATE ADDR
5672
5673
5674
5675  022044                              ROMCLK
5676  022044  004537  003044              JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
5677  022050  000400              2$:     000400                  ;MOVE PAGE,FIELD # TO BREG.
5678  022052                              ROMCLK
5679  022052  004537  003044              JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
5680  022056  061233                      61233                   ;MOV    BREG TO PC HIGH REG.
5681  022060                              SROMCLK
5682  022060  004537  003100              JSR     R5,.SROMCLK
5683  022064  100000              3$:     100000                  ;NOW CLOCK IT IN BY JMP FIELD INSTR.
5684
5685  022066                              ROMCLK                  ;READ PC REG HI
5686  022066  004537  003044              JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
5687  022072  121265                      121265
5688  022074                              ROMCLK                  ;READ PC REG LOW
5689  022074  004537  003044              JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
5690  022100  121244                      121244
5691
5692  022102  016104  000004              MOV 4(R1),R4            ;READ PC REG (NOW IN SEL 4)
5693  022106  042704  170000              BIC     #170000,R4      ;STRIP FOR ONLY PAGE,FIELD BITS.
5694
5695  022112  013705  022050              MOV 2$,R5               ;NOW FROM ADDR WE WANTED TO
5696  022116  000305                      SWAB R5                 ;JUMP TO
5697  022120  042705  170377              BIC #170377,R5          ;CLEAR JUNK
5698  022124  050205                      BIS R2,R5               ;ADD IMMED ADDR
5699  022126                              SKIP06  5$
5700                                      ;GOTO 5$ IF M8206
5701  022136  105205                      INCB    R5              ;UPDATE ADDR. EXPECTED SENCE THE READ
5702  022140                        5$:
5703                                      ;OF THE IBUS <13> INC THE PC.
5704
5705  022140  020504                      CMP R5,R4               ;JUMP GO OK?
5706  022142  001406                      BEQ 4$                  ;YEA, CONTINUES
5707  022144                              ERROR 15                ;FAILED TO JUMP PROPERLY.
5708  022150  104455                      TRAP    C$ERDF
5709  022152  000017                      .WORD   15
5710  022154  005055                      .WORD   EM0
5711  022156  010006                      .WORD   ERR15
5712                                      ;"FROM ADDR" REPRESENTS
5713                                      ;THE ADDRESS WE STARTED AT
5714                                      ;"TO ADDR" REPRESENTS WHERE
5715                                      ;WE EXPECTED TO JUMP TO.
5716                                      ;"BAD ADDR" REPRESENTS WHERE
5717                                      ;WE WENT TO.
5718
5719                              .REM %
```

```
                                                15      12      11      10      9 8 7           0
                                                ----------------------------------------------
                                                !UNUSED !  FIELD        ! PAGE    !IMMED. ADDR  !
                                                !       !   BITS        ! BITS    !             !
                                                ----------------------------------------------

                                        ;THIS IS A PICTURE OF THE P.C. REG.
                                         BITS 0-7 ARE IN IBUS*<12>
                                         BITS 8-11 ARE IN IBUS*<13>
                                         THEY GOT CLOCK IN THERE VIA JUMPS TAKEN
                                         THE FIELD BITS
                                         ARE IN BIT POSITION 0,1 OF THE INSTRUCTION AT 2$.

                                         3$ WAS THE JUMP ALWAYS INSTRUCTION. THE IMMED. ADDR.
                                         WAS IN 0-7 OF THE JUMP INSTR.  THE PAGE BITS,
                                         PC REG BITS 8,9, WERE IN BITS 11,12 OF THE INSTR.
                                         JUMP INSTRUCTIONS HAVE BEEN CHECKED OUT
                                         BEFORE, SO THE IMPORTANT THING TO REMEMBER TO
                                         WATCH IS THE "FROM ADDR","TO ADDR"

                                %

022160                          4$:      ESCAPE TST
022160  104410                           TRAP    C$ESCAPE
022162  000136                           .WORD   L10075-.
022164  010437  002412                   MOV R4,FADR
022170  005237  002406                   INC FLAG                ;UPDATE PAGE,FIELD
022174  105202                           INCB R2                 ;UPDATE JMMED. ADDR
022176  001304                           BNE 1$                  ;LOOP IF NOT DONE.

                                         ;*
                                         ;*CHECK HERE TO SEE IF MASTER CLEAR CLEARS P.C. REG
                                         ;*

022200                                   SKIP06  40$
                                         ;GOTO 40$ IF M8206
022210  005737  002470                   TST     RUNB
022214  001041                           BNE     40$
022216  005737  002472                   TST     RUNINH
022222  001036                           BNE     40$
022224  052711  040000                   BIS     #40000,(R1)     ;SET MASTER CLEAR
022230  105761  000001                   TSTB    1(R1)
022234  042711  040000                   BIC     #40000,(R1)

                                         ;TO RUN THIS SECTION OF CODE YOU MUST TURN SW7 OF SWITCH PACK #E28
                                         ;OFF SO THAT M8207 NOT SELFSTARTING.

022240                                   ROMCLK                  ;WE MUST FIRST CLOCK
022240  004537  003044                   JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
022244  121265                           121265                  ;THE PC LATCH REGS
022246                                   ROMCLK                  ;BEFORE WE CAN READ THEM
022246  004537  003044                   JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
022252  121244                           121244
022254                                   ROMCLK                  ;REG PC REG HI, PUT IN PORT5
022254  004537  003044                   JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
022260  121265                           121265
```

```
5776
5777  022262                                  ROMCLK                  ;REG PC REG LOW, PUT IN PORT4
5778  022262  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
5779  022266  121244                          121244
5780  022270  005005                          CLR     R5              ;EXPECT ZERO
5781  022272  016104  000004                  MOV     4(R1),R4        ;READ PC REG FROM PORT 4&5
5782  022276  042704  170003                  BIC     #170003,R4
5783  022302  001406                          BEQ     40$             ;IF CLEARED, EXIT
5784                                                                  ;NOTE WE ALSO CLEARED BIT 1 OF THE
5785                                                                  ;PC REG, BECAUSE AFTER THE MASTER
5786                                                                  ;CLEAR, WE DID TWO INSTRUCTIONS TO
5787                                                                  ;READ IT, THUS CAUSING THE PC REG
5788                                                                  ;TO GET BUMPED.
5789
5790  022304                                  ERROR   45              ;MASTER CLEAR FAILED TO CLEAR
5791  022310  104455                          TRAP    C$ERDF
5792  022312  000055                          .WORD   45
5793  022314  005055                          .WORD   EMO
5794  022316  011124                          .WORD   ERR45
5795                                                                  ;PC REG
5796  022320                          40$:
5797  022320                          ENDTST
5798  022320                          L10075:
5799  022320  104401                          TRAP    C$ETST
5800  022322                                  BADHEAD
5801                                          ;************** TEST 28 ***************
5802                                          ;*
5803                                          ;*JUMP TEST, JUMP ALWAYS, JUMP CHANGE FIELD
5804                                          ;*
5805                                          ;*IN THIS TEST, WE WILL CHECK THE ABILITY OF THE
5806                                          ;*MICRO PROCESSOR TO JUMP (BRANCH & ALWAYS INSTRUCTION)
5807                                          ;*TO LOCATIONS, FIELDS FROM OTHER LOCATIONS FIELDS.
5808                                          ;*WE ALREADY KNOW THAT THE BRANCH INSTR WORKS FROM
5809                                          ;*OTHER TEST.              PROCEDURE:
5810                                          ;*      1. START ADDR 0, FIELD 0
5811                                          ;*      2. **CALCULATE NEW ADDRS, FIELD  VIA DEC,
5812                                          ;*      3. CAUSE JUMP (BRANCH) TO NEW ADDRESS
5813                                          ;*      4. READ PC FROM IBUS*12 AND IBUS*13
5814                                          ;*      5. REPEAT STEP 2-4 256.TIMES
5815                                          ;*
5816                                          ;*      TO CALCULATE NEW ADDRESS:
5817                                          ;*      1. DEC LOW BYTE OF ADDRESS FOR PC ADDRES 0-7
5818                                          ;*      2. DEC LOW BYTE OF NADDRESS FOR PC ADDRESS 8-11
5819                                          ;*         BITS REPRESENTED AS BITS 0-3. WHEN  0-3 OVERFLOWS,
5820                                          ;*         RESTARTS AT ZERO.
5821                                          ;*         NET RESULT IS JUMPS FROM:
5822                                          ;*              FIELD,PAGE              LOC
5823                                          ;*                   0                  0
5824                                          ;*                   17                 377
```

```
                                        ;*                      16              376
                                        ;*                      15              375
                                        ;*                      :TO             :
                                        ;*                      00              000
                                        ;*
022322                                  BADHEAD
                                        ;*************** TEST 28 ***************

022322                                  BGNTST
022322                          T28::
022322                                  MYINT
022322  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
022326                                  K4ONLY                          ;4K CPUS ONLY.
                                        ;DO NOT DO TEST IF M8200, OR M8204
022336  104432                          TRAP    C$EXIT
022340  000216                          .WORD   L10076-.
022342                                  MSTCLR

022346  012737  000000  002406          MOV     #0, FLAG                ;FLAG TO REPRESENT
                                                                        ;FIELD,PAGE
                                                                        ;TO VARIE STARTING PAGE,FIELD,
                                                                        ;CHANGE #0 PORTION OF INSTR.
022354  012702  000000                  MOV     #0, R2                  ;R2 TO CONTAIN JUMPED
                                                                        ;TO CHANGE STARTING IMM ADDR.,
                                                                        ;VARIE #0 PORTIONS OF INSTR.
022360  012737  000000  002412          MOV     #0, FADR                ;ADDRESS
                                        ;LOOP HERE
022366                          1$:
022366  042737  000017  022426          BIC #17,2$                      ;CLEAR JUNK FROM FIELD
                                                                        ;PORTION OF CHANGE FIELD INSTR
022374  013700  002406                  MOV FLAG,R0                     ;INORDER TO INC, DEC FIELD,PAGE
022400  042700  177760                  BIC #^C<17>,R0
022404  050037  022426                  BIS R0,2$                       ;NOW POSITION IN INSTR.
022410  042737  077777  022442          BIC #077777,3$                          ;NOW FOR IMMED. BR INSTR.
022416  050237  022442                  BIS R2,3$                       ;NOW ADD IMMEDIATE ADDR


022422                                  ROMCLK
022422  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
022426  000400                  2$:     000400                          ;MOVE PAGE,FIELD # TO BREG.
022430                                  ROMCLK
022430  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
022434  061233                          61233                           ;MOV    BREG TO PC HIGH REG.
022436                                  SROMCLK
022436  004537  003100                  JSR     R5,.SROMCLK
022442  100000                  3$:     100000                          ;NOW CLOCK IT IN BY JMP FIELD INSTR.

022444                                  ROMCLK                          ;READ PC REG HI
022444  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
022450  121265                          121265
022452                                  ROMCLK                          ;READ PC REG LOW
022452  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
022456  121244                          121244

022460  016104  000004                  MOV 4(R1),R4                    ;READ PC REG (NOW IN SEL 4)
```

```
5881  022464  042704  170000                  BIC     #170000,R4      ;STRIP FOR ONLY PAGE,FIELD BITS.
5882
5883  022470  013705  022426                  MOV 2$,R5               ;NOW FROM ADDR WE WANTED TO
5884  022474  000305                          SWAB R5                 ;JUMP TO
5885  022476  042705  170377                  BIC #170377,R5          ;CLEAR JUNK
5886  022502  050205                          BIS R2,R5               ;ADD IMMED ADDR
5887  022504                                  SKIP06  5$
5888                                          ;GOTO 5$ IF M8206
5889  022514  105205                          INCB    R5              ;UPDATE ADDR. EXPECTED SENCE THE READ
5890  022516                          5$:
5891                                                                  ;OF THE IBUS <13> INC THE PC.
5892
5893  022516  020504                          CMP R5,R4               ;JUMP GO OK?
5894  022520  001406                          BEQ 4$                  ;YEA, CONTINUES
5895  022522                                  ERROR 15                ;FAILED TO JUMP PROPERLY.
5896  022526  104455                          TRAP    C$ERDF
5897  022530  000017                          .WORD   15
5898  022532  005055                          .WORD   EMO
5899  022534  010006                          .WORD   ERR15
5900                                                                  ;"FROM ADDR" REPRESENTS
5901                                                                  ;THE ADDRESS WE STARTED AT
5902                                                                  ;"TO ADDR" REPRESENTS WHERE
5903                                                                  ;WE EXPECTED TO JUMP TO,
5904                                                                  ;"BAD ADDR" REPRESENTS WHERE
5905                                                                  ;WE WENT TO.
5906
5907                          .REM %
5908                                                  15      12      11      10      9 8 7           0
5909                                                  ------------------------------------------------
5910                                                  !UNUSED !  FIELD        ! PAGE    !IMMED. ADDR  !
5911                                                  !       !   BITS        ! BITS    !             !
5912                                                  ------------------------------------------------
5913
5914                                  ;THIS IS A PICTURE OF THE P.C. REG.
5915                                   BITS 0-7 ARE IN IBUS*<12>
5916                                   BITS 8-11 ARE IN IBUS*<13>
5917                                   THEY GOT CLOCK IN THERE VIA JUMPS TAKEN
5918                                   THE FIELD BITS
5919                                   ARE IN BIT POSITION 0,1 OF THE INSTRUCTION AT 2$.
5920
5921                                   3$ WAS THE JUMP ALWAYS INSTRUCTION. THE IMMED. ADDR.
5922                                   WAS IN 0-7 OF THE JUMP INSTR.  THE PAGE BITS,
5923                                   PC REG BITS 8,9, WERE IN BITS 11,12 OF THE INSTR.
5924                                   JUMP INSTRUCTIONS HAVE BEEN CHECKED OUT
5925                                   BEFORE, SO THE IMPORTANT THING TO REMEMBER TO
5926                                   WATCH IS THE "FROM ADDR","TO ADDR"
5927
5928                          %
5929
5930  022536                  4$:     ESCAPE TST
5931  022536  104410                  TRAP    C$ESCAPE
5932  022540  000016                  .WORD   L10076-.
5933  022542  010437  002412          MOV R4,FADR
5934  022546  005337  002406          DEC FLAG                ;UPDATE PAGE,FIELD
5935  022552  105302                  DECB R2                 ;UPDATE JMMED. ADDR
5936  022554  001304                  BNE 1$                  ;LOOP IF NOT DONE.
```

```
                                ENDTST
                                L10076:
                                        TRAP    C$ETST
                                        BADHEAD
                                        ;*************** TEST 29 ***************
                                        ;*
                                        ;* IN THIS TEST WE'LL VERIFY THAT THE Z BIT CAN BE READ FROM
                                        ;* IBUS*<13>. WE ALLREADY KNOW THAT THE Z BIT WORKS PROPERLY,
                                        ;* ALL WE WANT TO KNOW HERE IS THAT IT CAN BE READ.
                                        ;*
                                        BADHEAD
                                        ;*************** TEST 29 ***************

                                        BGNTST
                                T29::
                                        K4ONLY                          ;M8206 &M8207 ONLY!
                                        ;DO NOT DO TEST IF M8200, OR M8204
022570  104432                          TRAP    C$EXIT
022572  000200                          .WORD   L10077-.
022574                                  MSTCLR
022600                                  MYINT
022600  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
022604  004737  003166                  JSR     PC,CLRALL               ;CLR CONDITION CODES.
022610                                  ROMCLK                          ;NOW READ IBUS*<15>PUT IN PORT 4
022610  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
022614  121264                          121264
022616  116104  000004                  MOVB    4(R1),R4                ;READ IT FROM PORT 4
022622  042704  177477                  BIC     #177477,R4              ;STRIP ANY JUNK,C&Z BITS 6,7
022626  012705  000000                  MOV     #0,R5                   ;EXPECT IT CLEAR
022632  120405                          CMPB    R4,R5                   ;OK?
022634  001410                          BEQ     1$
022636                                  ERROR   16                      ;FAILURE OF Z&C TO BE CLEAR.
022642  104455                          TRAP    C$ERDF
022644  000020                          .WORD   16
022646  005055                          .WORD   EMO
022650  010134                          .WORD   ERR16

022652                                  ESCAPE TST
022652  104410                          TRAP    C$ESCAPE
022654  000116                          .WORD   L10077-.
022656  004737  003312          1$:     JSR     PC,SETZ                 ;SET Z BIT.
022662                                  ROMCLK                          ;NOW GO BACK AND CHECK Z BIT SET.
022662  004537  003044                  JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
022666  121264                          121264

022670  016104  000004                  MOV     4(R1),R4                ;GET INFO.
022674  042704  177477                  BIC     #^C<300>,R4             ;STRIP FOR C&Z BITS.
022700  012705  000200                  MOV     #200,R5                 ;EXPECT ONLY Z BIT SET.
022704  120405                          CMPB    R4,R5                   ;SET OK?
022706  001410                          BEQ     2$
022710                                  ERROR   16                      ;Z BIT FAILED TO SET PROPERLY.
022714  104455                          TRAP    C$ERDF
022716  000020                          .WORD   16
022720  005055                          .WORD   EMO
```

```
                                        .WORD   ERR16

                                        ESCAPE  TST
                                        TRAP    C$ESCAPE
                                        .WORD   L10077-.
                        2$:             JSR     PC,CLRALL               ;NOW TRY TO CLEAR Z BIT.
                                        ROMCLK
                                        JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
                                        121264
                                        MOV     4(R1),R4
                                        BIC     #^C<300>,R4             ;STRIP FOR C&Z BITS
                                        BEQ     3$                      ;IF ZERO,WE'RE OK
                                        CLR     R5                      ;ELSE REPORT ERROR
                                        ERROR   16                      ;Z BIT FAILED TO CLEAR PROPERLY.
                                        TRAP    C$ERDF
                                        .WORD   16
                                        .WORD   EM0
                                        .WORD   ERR16
                        3$:
                                        ENDTST
                        L10077:
                                        TRAP    C$ETST
                        ;FINDFAST
                                        BADHEAD
                                        ;*************** TEST 30 ***************
                                        ;*
                                        ;* IN THIS TEST WE'LL VERIFY THAT THE C BIT CAN BE READ FROM
                                        ;* IBUS*<13>. WE ALLREADY KNOW THAT THE C BIT WORKS PROPERLY,
                                        ;* ALL WE WANT TO KNOW HERE IS THAT IT CAN BE READ.
                                        ;*
                                        BADHEAD
                                        ;*************** TEST 30 ***************

                                        BGNTST
                        T30::
                                        K4ONLY                          ;M8206 &M8207 ONLY!
                                        ;DO NOT DO TEST IF M8200, OR M8204
                                        TRAP    C$EXIT
                                        .WORD   L10100-.
                                        MSTCLR
                                        MYINT
                                        MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
                                        JSR     PC,CLRALL               ;CLR CONDITION CODES.
                                        ROMCLK                          ;NOW READ IBUS*<13>PUT IN PORT 4
                                        JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
                                        121264
                                        MOVB    4(R1),R4                ;READ IT FROM PORT 4
                                        BIC     #177477,R4              ;STRIP ANY JUNK,C&Z BITS 6,7
                                        MOV     #0,R5                   ;EXPECT IT CLEAR
                                        CMPB    R4,R5                   ;OK?
                                        BEQ     1$
                                        ERROR   16                      ;FAILURE OF Z&C TO BE CLEAR.
                                        TRAP    C$ERDF
                                        .WORD   16
                                        .WORD   EM0
                                        .WORD   ERR16
```

```
                                                ESCAPE TST
023066  104410                                  TRAP    C$ESCAPE
023066  000116                                  .WORD   L10100-.
023070  004737  003260          1$:             JSR     PC,SETC                 ;SET C BIT.
023074                                          ROMCLK                          ;NOW GO BACK AND CHECK C BIT SET.
023074  004537  003044                          JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
023100  121264                                  121264
023102  016104  000004                          MOV     4(R1),R4                ;GET INFO.
023106  042704  177477                          BIC     #^C<300>,R4             ;STRIP FOR C&Z BITS.
023112  012705  000100                          MOV     #100,R5                 ;EXPECT ONLY C BIT SET.
023116  120405                                  CMPB    R4,R5                   ;SET OK?
023120  001410                                  BEQ     2$
023122                                          ERROR   16                      ;C BIT FAILED TO SET PROPERLY.
023126  104455                                  TRAP    C$ERDF
023130  000020                                  .WORD   16
023132  005055                                  .WORD   EM0
023134  010134                                  .WORD   ERR16

023136                                          ESCAPE  TST
023136  104410                                  TRAP    C$ESCAPE
023140  000044                                  .WORD   L10100-.
023142  004737  003166          2$:             JSR     PC,CLRALL               ;NOW TRY TO CLEAR C BIT.
023146                                          ROMCLK
023146  004537  003044                          JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
023152  121264                                  121264
023154  016104  000004                          MOV     4(R1),R4
023160  042704  177477                          BIC     #^C<300>,R4             ;STRIP FOR C&Z BITS
023164  001407                                  BEQ     3$                      ;IF ZERO,WE'RE OK
023166  005005                                  CLR     R5                      ;ELSE REPORT ERROR
023170                                          ERROR   16                      ;C BIT FAILED TO CLEAR PROPERLY.
023174  104455                                  TRAP    C$ERDF
023176  000020                                  .WORD   16
023200  005055                                  .WORD   EM0
023202  010134                                  .WORD   ERR16
023204                          3$:
023204                                          ENDTST
023204                          L10100:
023204  104401                                  TRAP    C$ETST
023206                                          BADHEAD
                                                ;*************** TEST 31 ***************
                                                ;*TEST OF PROGRAM CLOCK BIT
                                                ;*DO A MASTER CLEAR, VERIFY THAT PROGRAM CLOCK IS SET
                                                ;*WRITE PROGRAM CLOCK BIT TO A ONE, VERIFY THAT IT CLEARS,
                                                ;*AND THEN SETS SOME TIME LATER
023206                                          BADHEAD
                                                ;*************** TEST 31 ***************

023206                          BGNTST
023206                          T31::
023206                                          MYINT
023206  013701  002516                          MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
023212                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
023216  005037  002440                          CLR     TEMP            ;PREPARE FOR
023222  005037  002444                          CLR     $TMP0           ;DELAY
023226  012761  000020  000004    1$:           MOV     #20,4(R1)       ;LOAD PORT 4
```

```
6105  023236  152761  000002  000001          BISB    #BIT1,1(R1)      ;SET ROMI
6106  023244  012761  121111  000006          MOV     #121111,6(R1)    ;SEL6   INSTRUCTION
6107  023252  152761  000003  000001          BISB    #BIT1!BIT0,1(R1);SET CLOCK BIT
6108  023260  012761  121224  000006          MOV     #121224,6(R1)    ;LOAD NEXT INSTRUCTION
6109  023266  152761  000003  000001          BISB    #BIT1!BIT0,1(R1);READ CLOCK BIT
6110  023274  142761  030001  000001          BICB    #BIT!BIT0,1(R1);CLEAR MAINT BITS
6111  023302  016104  000004                  MOV     4(R1),R4         ;PUT "FOUND" IN R4
6112  023306  005037  002452                  CLR     $GDDAT           ;PUT "EXPECTED" IN $GDDAT
6113  023312  123704  002452                  CMPB    $GDDAT,R4                ;IS PGM CLOCK CLEAR?
6114  023316  001406                          BEQ     2$
6115  023320  013702  002452                  MOV     $GDDAT,R2
6116  023324                                  ERRDF   50,EMB50                 ;ERROR, PGM CLOCK IS NOT CLEAR
6117  023324  104455                          TRAP    C$ERDF
6118  023326  000062                          .WORD   50
6119  023330  005545                          .WORD   EMB50
6120  023332  000000                          .WORD   0
6121  023334                          2$:
6122  023334                                  ROMCLK                   ;NEXT WORD IS INSTRUCTION,
6123  023334  004537  003044                  JSR     R5,..ROMCLK      ;CLOCK INSTRUCTION
6124  023340  121224                          121224                   ;PORT4 LU11
6125  023342  122761  000020  000004          CMPB    #20,4(R1)        ;IS PGM CLOCK SET?
6126  023350  001420                          BEQ     3$               ;BR IF YES
6127  023352  005237  002440                  INC     TEMP             ;INCREMENT DELAY
6128  023356  005537  002444                  ADC     $TMP0            ;INCREMENT DELAY
6129  023362  022737  000006  002444          CMP     #6,$TMP0         ;IS DELAY DONE
6130  023370  001361                          BNE     2$               ;BR IF NO
6131  023372  012702  000006                  MOV     #6,R2
6132  023376  013704  002444                  MOV     $TMP0,R4
6133  023402                                  ERRDF   51,EMB51                 ;ERROR PGM CLOCK NOT SET
6134  023402  104455                          TRAP    C$ERDF
6135  023404  000063                          .WORD   51
6136  023406  005577                          .WORD   EMB51
6137  023410  000000                          .WORD   0
6138  023412                          3$:
6139
6140  023412  122737  000007  002414          CMPB    #7,WTYPE         ; ONLY DO NEXT TEST IF M8207
6141  023420  001013                          BNE     4$               ; EXIT IF NOT.
6142
6143  023422  005737  002444                  TST     $TMP0            ; IF ANY LARGE COUNT, WE'RE OK
6144  023426  001010                          BNE     4$               ; THEN EXIT
6145
6146  023430  042737  000007  002440          BIC     #7,TEMP          ; CLEAR OUT ANY SMALL COUNT
6147  023436  001004                          BNE     4$               ; IF LARGE COUNT LEFT OVER, WE'RE OK.
6148
6149  023440                                  ERRDF   100,EMB1         ; ERROR
6150  023440  104455                          TRAP    C$ERDF
6151  023442  000144                          .WORD   100
6152  023444  005410                          .WORD   EMB1
6153  023446  000000                          .WORD   0
6154                                                                   ; TIME TIOOO SHORT FOR CLOCK.  MUST BE
6155                                                                   ; DEFECTIVE CAPACITOR IN TIMEING CIRCUIT.
6156  023450                          4$:
6157
6158  023450                          ENDTST
6159  023450                          L10101:
6160  023450  104401                          TRAP    C$ETST
```

```
                                                BADHEAD
                                                ;************** TEST 32 ***************
                                                ;*FORCE POWER FAIL TEST
                                                ;*SET FORCE POWER FAIL BIT VERIFY THAT PROCESSOR TRAPS TO 24
                                                ;*GOING DOWN AND COMING UP. VERIFY ALSO THAT BUS INIT WAS
                                                ;*BLOCKED FROM GETTING TO THE M8200,4,7 DURING THE POWER FAIL
                                                ;*THIS TEST WILL TAKE LONGER THAN 2 SECONDS TO RUN. THIS TEST
                                                ;*SHOULD NOT BE RUN IF YOU HAVE VOLATILABLE MEMORY IN YOUR SYSTEM.
                                                BADHEAD
                                                ;************** TEST 32 ***************

                                        BGNTST
                                        T32::
                                                BRESET                  ;STALL FOR TIME
023452  104433                                  TRAP    C$RESET
                                                MYINT
023454  013701  002516                          MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
                                                ;R1 CONTAINS BASE M8200,4,7 ADDRESS
023460                                          MSTCLR                  ;MASTER CLEAR M8200,4,7
023464  005037  002440                          CLR     TEMP            ;PREPARE FOR DELAY
023470  013737  000024  002444                  MOV     @#24,$TMP0      ;SAVE POWER FAIL ADDRESS
023476  013746  000024                          MOV     @#24,-(SP)      ;STORE POWER FAIL ADDRESS
023502  012737  023564  000024                  MOV     #1$,@#24        ;SET U FOPR FORCE POWER FAIL
023510  012761  000002  000004                  MOV     #2,4(R1)        ;LOAD PORT4
023516  012711  001000                          MOV     #BIT9,(R1)      ;SET ROMI
023522  012761  121111  000006                  MOV     #121111,6(R1)   ;LOAD INSTRUCTION
023530  012711  005400                          MOV     #BIT9!BIT8!BIT11,(R1)   ;CLOCK INSTRUCTION
023534  005237  002440                  5$:     INC     TEMP            ;WAIT FOR POWER FAIL
023540  001375                                  BNE     5$              ;BR IF DELAY NOT DONE
023542                                          MSTCLR
023546                                          ERROR   17      ;ERROR, NO POWER FAIL
023552  104455                                  TRAP    C$ERDF
023554  000021                                  .WORD   17
023556  005055                                  .WORD   EM0
023560  010256                                  .WORD   ERR17
023562  000445                                  BR      4$
023564  012737  023602  000024          1$:     MOV     #3$,@#24        ;POWER UP ADDRESS
023572  010637  023600                          MOV     SP,2$           ;STORE STACK
023576  000000                                  HALT                    ;WAIT FOR POWER UP SEQUENCE
023600  000000                          2$:     0
023602  013706  023600                  3$:     MOV     2$,SP           ;RESTORE STACK
023606  012737  024002  000024                  MOV     #10$,@#24       ;PUT IN CASE OF FALSE POWER-UP.
023614  005037  024000                          CLR     11$
023620  005237  024000                  12$:    INC     11$     ;STALL ON POWER UP.
023624  001375                                  BNE     12$             ;WAIT HERE IF BAD,WILL POWER OUT OF HERE.
                                                                        ;ELSE PROCEED.

023626                                          POPSP2                  ;POP STACK TWICE2
023630  013701  002516                          MOV     KMCSR,R1
023634  012637  000024                          MOV     (SP)+,@#24      ;RESTORE TRUE POWER FAIL ADDRESS
023640  023737  002444  000024                  CMP     $TMP0,@#24      ;IS IT CORRECT?
023646  001413                                  BEQ     4$              ;BR IF YES
023650                                          ERROR   17              ;ERROR, STACK IS INCORRECT
023654  104455                                  TRAP    C$ERDF
023656  000021                                  .WORD   17
```

```
                                        .WORD   EM0
                                        .WORD   ERR17
023664  013737  002444  000024          MOV     $TMP0,@#24      ;RESTORE TRUE POWER FAIL ADDRESS
023672  013706  002344                  MOV     PSTACK,SP       ;RESTORE STACK
023676  032711  004000          4$:     BIT     #BIT11,(R1)     ;BIT11 STILL SET?
023702  001016                          BNE     7$
023704  005737  002470                  TST     RUNB
023710  001013                          BNE     7$
023712  011104                          MOV     (R1),R4
023714  012705  004000                  MOV     #BIT11,R5
023720                                  ERROR   35              ;OAC FAILED
023724  104455                          TRAP    C$ERDF
023726  000043                          .WORD   35
023730  005055                          .WORD   EM0
023732  010472                          .WORD   ERR35
                                                                ;TO PREVENT
                                                                ;INIT FROM
                                                                ;CLEARING CSR
023734                          EXIT    TST
023734  104432                          TRAP    C$EXIT
023736  000104                          .WORD   L10102-.
023740  012711  003000          7$:     MOV     #BIT9!BIT10,(R1) ;SEL6 = MAINT IR
023744  012705  121111                  MOV     #121111,R5      ;R5 = EXPECTED
023750  016104  000006                  MOV     6(R1),R4        ;R4 = FOUND
023754  020504                          CMP     R5,R4           ;MAINT IR SHOULD = 12111
023756  001431                          BEQ     6$              ;BR IF OK
023760                                  MSTCLR
023764                                  ERROR   35              ;IF = 0 THEN BUS INIT WAS
023770  104455                          TRAP    C$ERDF
023772  000043                          .WORD   35
023774  005055                          .WORD   EM0
023776  010472                          .WORD   ERR35
                                                                ;NOT BLOCKED FROM CLEARING
                                                                ;THE M8200,4,7
024000  000000          11$:    .WORD   0                       ;TEMP COUNT FOR STALL ON POWER UP.

024002  052711  040000  10$:    BIS     #BIT14,(R1)             ;CLR THE THING SO IT CAN'T ASSIRT AC LOW
                                                                ;AGAIN!
024006                                  MSTCLR
024012                                  ERROR   17              ;ERROR  GLIP GAVE US SECOUND UNEXPECTED
024016  104455                          TRAP    C$ERDF
024020  000021                          .WORD   17
024022  005055                          .WORD   EM0
024024  010256                          .WORD   ERR17
                                                                ;ASSERTION OF AC LOW ON UNIBUS.
                                                                ;FATEL TYPE OF ERROR.
024026  062706  000004                  ADD     #4,SP           ;RESTORE STACK.
024032  012637  000024                  MOV     (SP)+,@#24
024036                                  MSTCLR
024042                          6$:
024042                          ENDTST
024042                          L10102:
024042  104401                          TRAP    C$ETST

024044                                  BADHEAD
```

```
                                        ;*************** TEST 33 ***************
                                        ;*MICRO-PROCESSOR NOISE TEST
                                        ;*WRITE ALL ZERO'S THEN ALL ONE'S THEN A DATA PATTTERN
                                        ;*TO THE IBUS* AND IBUS REGISTERS AND TO THE SP AND MAIN MEM
                                        ;*THEN GO BACK AND READ THE DATA PATERNS TO VERIFY THAT
                                        ;*READING AND WRITING OF OTHER LOCATIONS AND REGISTERS
                                        ;*DID NOT CHANGE THE DATA.
024044                                  BADHEAD
                                        ;*************** TEST 33 ***************

024044                          BGNTST
024044                          T33::
024044                                  MYINT
024044  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
024050                                  MSTCLR                          ;MASTER CLEAR M8200,4,7
024054  005002                          CLR     R2                      ;R2 IS INDEX REGISTER
024056  042737  000017  024104  1$:     BIC     #17,2$                  ;CLEAR ADDRESS FIELD
024064  156237  025100  024104          BISB    30$(R2),2$              ;ADD IBUS* REG ADDRESS TO INSTRUCTION
024072  116261  025106  000004          MOVB    31$(R2),4(R1)           ;LOAD PORT4
024100                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,
024100  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
024104  121100                  2$:     121100                          ;WRITE IBUS* REGISTER
024106  005202                          INC     R2                      ;INC INDEX REGISTER
024110  022702  000005                  CMP     #5,R2                   ;DONE YET?
024114  001360                          BNE     1$                      ;BR IF NO
024116  005002                          CLR     R2                      ;R2 IS IBUS REGISTER ADDRESS
024120  042737  000017  024166  3$:     BIC     #17,4$                  ;CLEAR ADDRESS FIELD OF INSTRUCTIONS
024126  042737  000017  024202          BIC     #17,5$
024134  042737  000017  024214          BIC     #17,6$
024142  050237  024166                  BIS     R2,4$                   ;ADD IBUS REG ADDRESS TO INSTRUCTION
024146  050237  024202                  BIS     R2,5$
024152  050237  024214                  BIS     R2,6$
024156  105061  000004                  CLRB    4(R1)                   ;CLEAR PORT4
024162                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,
024162  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
024166  122100                  4$:     122100                          ;WRITE 0 TO IBUS REG
024170  112761  000377  000004          MOVB    #377,4(R1)              ;LOAD PORT4
024176                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,
024176  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
024202  122100                  5$:     122100                          ;WRITE ALL ONES TO  IBUS REG
024204  110261  000004                  MOVB    R2,4(R1)                ;LOAD PORT4
024210                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,
024210  004537  003044                  JSR     R5,.ROMCLK              ;CLOCK INSTRUCTION
024214  122100                  6$:     122100                          ;WRITE ITS OWN ADDRESS TO IBUS REG
024216  005202                          INC     R2                      ;NEXT ADDRESS
024220  022702  000010                  CMP     #10,R2                  ;DONE YET?
024224  001335                          BNE     3$                      ;BR IF NO
024226  005002                          CLR     R2                      ;START AT SP ADDRESS 0
024230  042737  000017  024276  7$:     BIC     #17,8$                  ;CLEAR ADDRESS FIELD
024236  042737  000017  024312          BIC     #17,9$
024244  042737  000017  024324          BIC     #17,10$
024252  050237  024276                  BIS     R2,8$                   ;ADD ADDRESS TO INSTRUCTION
024256  050237  024312                  BIS     R2,9$
024262  050237  024324                  BIS     R2,10$
024266  105061  000004                  CLRB    4(R1)                   ;CLEAR PORT4
024272                                  ROMCLK                          ;NEXT WORD IS INSTRUCTION,
```

```
6329  024272  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
6330  024276  123100                8$:       123100                  ;WRITE ZERO TO SP
6331  024300  112761  000377  000004          MOVB    #377,4(R1)      ;LOAD PORT4
6332  024306                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6333  024306  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
6334  024312  123100                9$:       123100                  ;WRITE ALL ONES TO SP
6335  024314  110261  000004                  MOVB    R2,4(R1)        ;LOAD PORT4
6336  024320                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6337  024320  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
6338  024324  123100                10$:      123100                  ;WRITE SP ADDRESS TO ITSELF
6339  024326  005202                          INC     R2              ;NEXT SP ADDRESS
6340  024330  022702  000020                  CMP     #20,R2          ;DONE YET?
6341  024334  001335                          BNE     7$              ;BR IF NO
6342  024336  005002                          CLR     R2              ;R2 = ,AOM ,E, ADDRESS
6343  024340                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6344  024340  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
6345  024344  010000                          010000                  ;MAR _ 0
6346  024346                                  ROMCLK
6347  024346  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
6348  024352  004000                          4000
6349  024354  105061  000004        11$:      CLRB    4(R1)           ;CLEAR PORT4
6350  024360                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6351  024360  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
6352  024364  122500                          122500                  ;WRITE ZEROS TO MEM
6353  024366  112761  000377  000004          MOVB    #377,4(R1)      ;LOAD PORT4
6354  024374                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6355  024374  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
6356  024400  122500                          122500                  ;WRITE ONES TO MEM
6357  024402  110261  000004                  MOVB    R2,4(R1)        ;LOAD PORT4
6358  024406                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6359  024406  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
6360  024412  136500                          136500                  ;WRITE TO MEM IT OWN ADDRESS
6361  024414  005202                          INC     R2              ;NEXT MEM ADDRESS
6362  024416  022702  001000                  CMP     #1000,R2        ;DONE YET?
6363  024422  001354                          BNE     11$             ;BR IF NO
6364
6365                                          ;NOW GO BACK AND READ EVERYTHIN
6366
6367  024424                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6368  024424  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
6369  024430  010000                          010000                  ;MAR 0
6370  024432                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6371  024432  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
6372  024436  004000                          4000                    ;MAR HI _ 0 (M8200,4,7 ONLY)
6373                                          ;WOULD BE CRAM CODE
6374  024440  005737  002432                  TST     TYPE
6375  024444  001452                          BEQ     40$
6376  024446  005005                          CLR     R5              ;R5 IS INDEX REGISTER
6377  024450  042737  000360  024512  12$:    BIC     #360,13$        ;CLEAR ADDRESS FIELD
6378  024456  116502  025100                  MOVB    30$(R5),R2      ;R2 = IBUS* ADDRESS
6379  024462  010203                          MOV     R2,R3           ;PUT IBUS* ADDRESS IN R3
6380  024464  006303                          ASL     R3              ;SHIFT ADDRESS TO BITS 4-7
6381  024466  006303                          ASL     R3
6382  024470  006303                          ASL     R3
6383  024472  006303                          ASL     R3
6384  024474  050337  024512                  BIS     R3,13$          ;ADD ADDRESS TO INSTRUCTION
```

```
6385  024500  116537  025106  002452          MOVB    31$(R5),$GDDAT  ;$GDDAT = "EXPECTED"
6386  024506                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6387  024506  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
6388  024512  121004                  13$:    121004                  ;PORT4 - IBUS* REGISTER
6389  024514  016104  000004                  MOV     4(R1),R4        ;R4 = "FOUND"
6390  024520  123704  002452                  CMPB    $GDDAT,R4               ;IBUS* CONTENTS OK?
6391  024524  001416                          BEQ     20$             ;BR IF YES
6392  024526  010237  002434                  MOV     R2,MR0
6393  024532  105037  002453                  CLRB    $GDDAT+1
6394  024536  013705  002452                  MOV     $GDDAT,R5
6395  024542                                  ERROR   29              ;IBUS* DATA ERROR
6396  024546  104455                          TRAP    C$ERDF
6397  024550  000035                          .WORD   29
6398  024552  005055                          .WORD   EM0
6399  024554  010350                          .WORD   ERR29
6400  024556                                  ESCAPE  TST
6401  024556  104410                          TRAP    C$ESCAPE
6402  024560  000334                          .WORD   L10103-.
6403  024562  005205                  20$:    INC     R5              ;INC COUNTER
6404  024564  022705  000005                  CMP     #5,R5           ;DONE YET?
6405  024570  001327                          BNE     12$             ;BR IF NO
6406
6407  024572                          40$:
6408                                  ;END CRAM,GENERAL TESTS
6409
6410  024572  005002                          CLR     R2              ;R2 = IBUS REG ADDRESS
6411  024574  042737  000360  024630  14$:    BIC     #360,15$        ;CLEAR ADDRESS FIELD OF INSTRUCTION
6412  024602  010203                          MOV     R2,R3           ;R3 = IBUS ADDRESS
6413  024604  006303                          ASL     R3              ;SHIFT ADDRESS TO BITS 4-7
6414  024606  006303                          ASL     R3
6415  024610  006303                          ASL     R3
6416  024612  006303                          ASL     R3
6417  024614  050337  024630                  BIS     R3,15$          ;ADD ADDRESS TO INSTRUCTION
6418  024620  010237  002452                  MOV     R2,$GDDAT               ;$GDDAT = "EXPECTED"
6419  024624                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6420  024624  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
6421  024630  021004                  15$:    021004                  ;PORT4 - IBUS REG
6422  024632  016104  000004                  MOV     4(R1),R4        ;IBUS = "FOUND"
6423  024636  123704  002452                  CMPB    $GDDAT,R4               ;IBUS CONTENTS OK?
6424  024642  001410                          BEQ     21$             ;BR IF YES
6425  024644  013705  002452                  MOV     $GDDAT,R5
6426  024650                                  ERROR   29              ;IBUS DATA ERROR
6427  024654  104455                          TRAP    C$ERDF
6428  024656  000035                          .WORD   29
6429  024660  005055                          .WORD   EM0
6430  024662  010350                          .WORD   ERR29
6431  024664  005202                  21$:    INC     R2              ;NEXT IBUS REGISTER
6432  024666  022702  000010                  CMP     #10,R2          ;DONE YET?
6433  024672  001340                          BNE     14$             ;BR IF NO
6434  024674  005002                          CLR     R2              ;R2 = SP ADDRESS
6435  024676  042737  000017  024714  16$:    BIC     #17,17$         ;CLEAR ADDRESS FIELD OF INSTRUCTION
6436  024704  050237  024714                  BIS     R2,17$          ;ADD ADDRESS TO INSTRUCTION
6437  024710                                  ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6438  024710  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
6439  024714  040600                  17$:    040600                  ;BR - SP
6440  024716  010237  002452                  MOV     R2,$GDDAT               ;$GDDAT = "EXPECTED"
```

```
6441  024722                                ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC-5304
6442  024722  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6443  024726  061224                        061224                  ;PORT4   BR
6444  024730  016104  000004                MOV     4(R1),R4        ;R4 = "FOUND"
6445  024734  123704  002452                CMPB    $GDDAT,R4               ;SP CONTENTS OK?
6446  024740  001412                        BEQ     22$             ;BR IF YES
6447  024742  013705  002452                MOV     $GDDAT,R5
6448  024746                                ERROR   7               ;SP DATA ERROR
6449  024752  104455                        TRAP    C$ERDF
6450  024754  000007                        .WORD   7
6451  024756  005055                        .WORD   EM0
6452  024760  007026                        .WORD   ERR7
6453  024762                                ESCAPE  TST
6454  024762  104410                        TRAP    C$ESCAPE
6455  024764  000130                        .WORD   L10103-.
6456  024766  005202                 22$:   INC     R2              ;NEXT SP LOCATION
6457  024770  022702  000020                CMP     #20,R2          ;DONE YET?
6458  024774  001340                        BNE     16$             ;BR IF NO
6459  024776  005002                        CLR     R2              ;R2 = MEMORY ADDRESS
6460  025000                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6461  025000  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6462  025004  010000                        010000                  ;MAR _ 0
6463  025006                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6464  025006  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6465  025012  004000                        4000                    ;MAR HI _ 0 (M8200,4,7 OR FAMILY ONLY)
6466  025014  010237  002452           18$: MOV     R2,$GDDAT               ;$GDDAT = "EXPECTED"
6467  025020                                ROMCLK                  ;NEXT WORD IS INSTRUCTION,
6468  025020  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6469  025024  055224                        055224                  ;PORT4   MAIN MEM
6470  025026  016104  000004                MOV     4(R1),R4        ;R4 = "FOUND"
6471  025032  123704  002452                CMPB    $GDDAT,R4               ;MAIN MEM CONTENTS OK?
6472  025036  001412                        BEQ     23$             ;BR IF YES
6473  025040  013705  002452                MOV     $GDDAT,R5
6474  025044                                ERROR   6               ;MAIN MEM DATA ERROR
6475  025050  104455                        TRAP    C$ERDF
6476  025052  000006                        .WORD   6
6477  025054  005055                        .WORD   EM0
6478  025056  006700                        .WORD   ERR6
6479  025060                                ESCAPE  TST
6480  025060  104410                        TRAP    C$ESCAPE
6481  025062  000032                        .WORD   L10103-.
6482  025064  005202                 23$:   INC     R2              ;NEXT MEM ADDRESS
6483  025066  022702  001000                CMP     #1000,R2                ;DONE YET?
6484  025072  001350                        BNE     18$             ;BR IF NO
6485  025074                                EXIT    TST
6486  025074  104432                        TRAP    C$EXIT
6487  025076  000016                        .WORD   L10103-.
6488  025100     000     002     003  30$:   .BYTE   0,2,3,5,10
6489  025103     005     010
6490
6491        025106                          .EVEN
6492  025106     001     003     004  31$:   .BYTE   1,3,4,6,10
6493  025111     006     010
```

```
                                .EVEN

025114                          ENDTST
025114                          L10103:
025114  104401                          TRAP    C$ETST

025116                                  BADHEAD
                                        ;************** TEST 34 ***************
                                        ;* THIS TEST IS DESIGNED TO MAKE SURE THAT A NODST INSTRUCTION
                                        ;* DOES NOT WRITE INTO PORT B OF THE MULTIPORT RAM.
                                        ;* TO DO THIS,WE'LL PUT A 125 INTO INDAT2,THEN WE'LL PUT A
                                        ;* 125 INTO BOTH SP1 AND BR. LAST WE'LL DO A NODST BR,SUBOC,SP1
                                        ;* IF THERE IS A WRITE INTO PORTB,INDAT2 WILL CONTAIN A 377.
025116                                  BADHEAD
                                        ;************** TEST 34 ***************

025116                          BGNTST
025116                          T34::
025116                                  MYINT
025116  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
025122                                  ROMCLK
025122  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
025126  000525                          00525                   ;PUT A 125 INTO BRG.
025130                                  ROMCLK
025130  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
025134  062221                          062221                  ;NOW INTO OIDAT2
025136                                  ROMCLK
025136  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
025142  063221                          63221                   ;NOW INTO SP1
025144                                  ROMCLK
025144  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
025150  060361                          060361                  ;NOW THE "NODST  BR,SUBOC,SP1"
                                                                ;THE NODST SHOULD NOT MODIFY INDAT2!

025152                                  ROMCLK
025152  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
025156  020420                          020420                  ;PUT CONTENT OF INDAT2 IN BRG.

025160                                  ROMCLK
025160  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
025164  061220                          061220                  ;PUT BRG INTO BSEL0

025166  111104                          MOVB    (R1),R4         ;SEE WHAT CAME BACK.
025170  012705  000125                  MOV     #125,R5         ;SHOULD BE 125  IF 377 CAME BACK,
                                                                ;YOU CAN BET THAT THE "NODST" WROTE
                                                                ;INTO THE MULTIPORT RAM!  WATCH SIGNAL
                                                                ; "D1 WRITE OUT L"

025174  020405                          CMP     R4,R5           ;NOW LOOK.
025176  001406                          BEQ     10$

025200                                  ERROR   7
025204  104455                          TRAP    C$ERDF
025206  000007                          .WORD   7
025210  005055                          .WORD   EM0
```

```
6550  025212  007026                        .WORD   ERR7
6551
6552  025214                        10$:
6553  025214                                ENDTST
6554  025214                        L10104:
6555  025214  104401                        TRAP    C$ETST
6556
6557  025216                                BADHEAD
6558                                        ;*************** TEST 35 ***************
6559                                        ;*
6560                                        ;* EXTENDED CRAM TEST FOR M8206. IN THIS TEST WE WILL LOAD DATA
6561                                        ;* THROUGHOUT THE CRAM (TEST DATA IS JUST 4K OF DIAG. CODE) AND
6562                                        ;* THEN READ IT BACK AND VERIFY THAT IT IS CORRECT.
6563  025216                                BADHEAD
6564                                        ;*************** TEST 35 ***************
6565
6566  025216                                BGNTST
6567  025216                        T35::
6568  025216                                SKIP06  10$                     ;DO TEST ONLY IF IT IS A M8206
6569                                        ;GOTO 10$ IF M8206
6570  025226                                EXIT    TST                     ;OTHERWIZE,SKIP TEST.
6571  025226  104432                        TRAP    C$EXIT
6572  025230  000132                        .WORD   L10105-.
6573
6574  025232                        10$:    MYINT
6575  025232  013701  002516                MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
6576
6577  025236  012702  012146                MOV     #ROMMAP,R2              ;GET ADDR. OF LIST.
6578
6579  025242  012711  002000                MOV     #2000,(R1)              ;SET TO WRITE DATA.
6580  025246  005003                        CLR     R3                      ;CRAM ADDR ZERO.
6581
6582  025250  010361  000004        15$:    MOV     R3,4(R1)                ;SET ADDR.
6583  025254  012261  000006                MOV     (R2)+,6(R1)             ;WRITE DATA.
6584
6585  025260  020337  002436                CMP     R3,MEMSZ                ;DONE WHOLE CRAM?
6586  025264  001402                        BEQ     20$                     ;YES,EXIT THIS LOOP.
6587  025266  005203                        INC     R3                      ;NO,UPDAT ADDR.
6588  025270  000767                        BR      15$
6589  025272  005003                20$:    CLR     R3                      ;NOW WE WILL READ BACK,STARTING AT
6590                                                                        ;CRAM ADDR. ZERO.
6591  025274  012705  012146                MOV     #ROMMAP,R5              ;GET ADDR. LIST OF DATA
6592
6593  025300  010361  000004        30$:    MOV     R3,4(R1)                ;SET ADDR.
6594
6595  025304  011502                        MOV     (R5),R2                 ;PUT EXPECTED INTO R2
6596  025306  016104  000006                MOV     6(R1),R4                ;READ ACCUAL
6597  025312  020204                        CMP     R2,R4                   ;EQUAL?
6598  025314  001411                        BEQ     40$                     ;YES,CONTINUE.
6599  025316  010300                        MOV     R3,R0
6600
6601  025320                                ERROR   1                       ;ERROR CRAM DATA TEST,DATA
6602  025324  104455                        TRAP    C$ERDF
6603  025326  000001                        .WORD   1
6604  025330  005055                        .WORD   EM0
6605  025332  006032                        .WORD   ERR1
```

```
                                                                ;READ NOT DATA THAT WAS WRITTEN.

025334                                  ESCAPE TST
025334  104410                          TRAP    C$ESCAPE
025336  000024                          .WORD   L10105-.
025340  020337  002436          40$:     CMP     R3,MEMSZ        ;ALL DONE?
025344  001002                          BNE     50$

025346                                  EXIT    TST
025346  104432                          TRAP    C$EXIT
025350  000012                          .WORD   L10105-.

025352  005203                  50$:     INC     R3              ;UPDATE ADDR.
025354  062705  000002                  ADD     #2,R5
025360  000747                          BR      30$

025362                          ENDTST
025362                          L10105:
025362  104401                          TRAP    C$ETST


025364                                  BADHEAD
                                        ;*************** TEST 36 ***************
                                        ;*
                                        ;* THIS TEST LOADS MICRO-CODE INTO A M8206 MCPU THEN EXECTUES IT.
                                        ;* THE MICRO CODE IS DESIGNED TO WRITE ALL ONES INTO THE SEL REGS.
                                        ;* THIS TEST IS ONLY PERFORMED ON AN M8206.
025364                                  BADHEAD
                                        ;*************** TEST 36 ***************

025364                                  BGNTST
025364                          T36::

025364                                  SKIP06  1$              ;ONLY DO THIS TEST IF M8206
                                        ;GOTO 1$ IF M8206
025374                                  EXIT    TST
025374  104432                          TRAP    C$EXIT
025376  000442                          .WORD   L10106-.

025400                          1$:     MYINT
025400  013701  002516                  MOV     KMCSR,R1        ;RECORD DEVICE ADDR.

025404  004537  026006                  JSR     R5,LOADER       ;LOAD THE MICRO CODE

025410  000777                          777                     ;MOVE #377,BRG
025412  061220                          061220                  ;MOVE BRG,BSEL0
025414  061222                          061222                  ;MOVE BRG,BSEL2
025416  061223                          061223                  ;MOVE   BRG,BSEL3
025420  061224                          061224                  ;MOVE   BRG,BSEL4
025422  061225                          061225                  ;MOVE   BRG,BSEL5
025424  061226                          061226                  ;MOVE BRG,BSEL6
025426  061227                          061227                  ;MOVE   BRG,BSEL7
025430  123000                          123000                  ;MOVE BSEL0,SP0
025432  101410                          101410                  ;BRANCH BACK ONE UNTIL <>377

025434  000400                          400                     ;MOVE   #0,BRG
```

```
6662  025436  061220                  61220                   ;MOVE BRG,BSEL0
6663  025440  061222                  61222                   ;MOVE BRG,BSEL2
6664  025442  061223                  61223                   ;MOVE BRG,BSEL3
6665  025444  061224                  61224                   ;MOVE BRG,BSEL4
6666  025446  061225                  61225                   ;MOVE BRG,BSEL5
6667  025450  061226                  61226                   ;MOVE BRG,,BSEL6
6668  025452  061227                  61227                   ;MOVE BRG,BSEL7
6669  025454  123000                  123000                  ;MOVE BSEL0,SP0
6670  025456  104022                  104022                  ;BRANCH BACK ONE LOCATION.
6671  025460  177777                  177777
6672
6673  025462  012711  040000          MOV     #040000,(R1)    ;INITIALIZE MCPU
6674  025466  012711  100000          MOV     #100000,(R1)    ;START CPU.
6675
6676  025472  012700  000062          MOV     #50.,R0         ;THE CYCLE TIME ON THE M8206 IS
6677                                                          ;200NS.  WE ARE ASKING THE MCPU TO
6678                                                          ;DO 8 INSTRUCTIONS.  WE'LL DELAY
6679                                                          ;100 PDP11 INSTRUCTIONS
6680                                                          ;THIS REALLY SHOULD BE PLENTY OF TIME.
6681
6682  025476  005300          20$:    DEC     R0
6683  025500  001376                  BNE     20$
6684
6685  025502  005005                  CLR     R5              ;JUST FOR TYPEOUT.
6686  025504  012705  000377          MOV     #377,R5         ;EXPECT 377
6687  025510  111104                  MOVB    (R1),R4         ;READ MCPU
6688  025512  120405                  CMPB    R4,R5           ;SEE IF OK.
6689  025514  001410                  BEQ     30$
6690
6691  025516                          ERROR   29              ;ERROR! MCPU WAS TO WRITE ALL
6692  025522  104455                  TRAP    C$ERDF
6693  025524  000035                  .WORD   29
6694  025526  005055                  .WORD   EM0
6695  025530  010350                  .WORD   ERR29
6696                                                          ;ONES INTO BSEL0,BUT INSTEAD FAILED.
6697  025532                          ESCAPE  TST
6698  025532  104410                  TRAP    C$ESCAPE
6699  025534  000304                  .WORD   L10106-.
6700
6701  025536  012705  177777  30$:    MOV     #177777,R5      ;EXPECT ALL ONES
6702  025542  016104  000002          MOV     2(R1),R4        ;RECIEVED
6703  025546  020405                  CMP     R4,R5           ;RECIEVE OK?
6704  025550  001410                  BEQ     40$
6705
6706  025552                          ERROR   29              ;ERROR! MCPU WAS TO WRITE ALL ONES
6707  025556  104455                  TRAP    C$ERDF
6708  025560  000035                  .WORD   29
6709  025562  005055                  .WORD   EM0
6710  025564  010350                  .WORD   ERR29
6711                                                          ;INTO BSEL 2&3
6712
6713  025566                          ESCAPE  TST
6714  025566  104410                  TRAP    C$ESCAPE
6715  025570  000250                  .WORD   L10106-.
6716
6717  025572  016104  000004  40$:    MOV     4(R1),R4        ;READ BSEL 4&5
```

```
6718  025576  020405                  CMP     R4,R5                   ;READ OK?
6719  025600  001410                  BEQ     50$
6720
6721  025602                          ERROR   29                      ;ERROR! FAILED TO WRITE BSEL $&5
6722  025606  104455                  TRAP    C$ERDF
6723  025610  000035                  .WORD   29
6724  025612  005055                  .WORD   EM0
6725  025614  010350                  .WORD   ERR29
6726                                                                  ; TO ALL ONES.
6727  025616                          ESCAPE  TST
6728  025616  104410                  TRAP    C$ESCAPE
6729  025620  000220                  .WORD   L10106-.
6730
6731  025622  016104  000006  50$:    MOV     6(R1),R4                ;READ BSEL 6&7
6732  025626  020405                  CMP     R4,R5                   ;READ OK?
6733  025630  001410                  BEQ     60$
6734
6735  025632                          ERROR   29                      ;ERROR! FAILED TO WRITE BSEL 6&7
6736  025636  104455                  TRAP    C$ERDF
6737  025640  000035                  .WORD   29
6738  025642  005055                  .WORD   EM0
6739  025644  010350                  .WORD   ERR29
6740                                                                  ; TO ALL ONES.
6741  025646                          ESCAPE  TST
6742  025646  104410                  TRAP    C$ESCAPE
6743  025650  000170                  .WORD   L10106-.
6744  025652  105011          60$:    CLRB    (R1)                    ;SIGNAL MCPU TO WRITE ALL ZEROS.
6745  025654  005005                  CLR     R5                      ;EXPECT TO READ ALL ZEROS.
6746
6747  025656  005004                  CLR     R4
6748  025660  111104                  MOVB    (R1),R4                 ;READ BSEL0
6749  025662  001410                  BEQ     70$                     ;EXPECT ZERO.
6750
6751  025664                          ERROR   29                      ;MCPU FAILED TO CLEAR BSEL0
6752  025670  104455                  TRAP    C$ERDF
6753  025672  000035                  .WORD   29
6754  025674  005055                  .WORD   EM0
6755  025676  010350                  .WORD   ERR29
6756
6757  025700                          ESCAPE  TST
6758  025700  104410                  TRAP    C$ESCAPE
6759  025702  000136                  .WORD   L10106-.
6760  025704  016104  000002  70$:    MOV     2(R1),R4                ;READ BSEL 2&3
6761  025710  001410                  BEQ     80$                     ;IF ZERO,OK
6762
6763  025712                          ERROR   29                      ;MCPU FAILED TO CLEAR BSEL 2&3
6764  025716  104455                  TRAP    C$ERDF
6765  025720  000035                  .WORD   29
6766  025722  005055                  .WORD   EM0
6767  025724  010350                  .WORD   ERR29
6768  025726                          ESCAPE  TST
6769  025726  104410                  TRAP    C$ESCAPE
6770  025730  000110                  .WORD   L10106-.
6771  025732                  80$:
6772  025732  016104  000004          MOV     4(R1),R4                ;READ BSEL 4&5
6773  025736  001410                  BEQ     90$
```

```
6774
6775  025740                          ERROR   29                      ;MCPU FAILED TO CLEAR BSEL 4&5
6776  025744  104455                  TRAP    C$ERDF
6777  025746  000035                  .WORD   29
6778  025750  005055                  .WORD   EM0
6779  025752  010350                  .WORD   ERR29
6780  025754                          ESCAPE  TST
6781  025754  104410                  TRAP    C$ESCAPE
6782  025756  000062                  .WORD   L10106-.
6783  025760                  90$:
6784  025760  016104  000006          MOV     6(R1),R4                ;READ BSEL 6&7
6785  025764  001406                  BEQ     95$
6786
6787  025766                          ERROR   29                      ;MCPU FAILED TO CLEAR BSEL 6&7
6788  025772  104455                  TRAP    C$ERDF
6789  025774  000035                  .WORD   29
6790  025776  005055                  .WORD   EM0
6791  026000  010350                  .WORD   ERR29
6792
6793  026002                  95$:    EXIT    TST
6794  026002  104432                  TRAP    C$EXIT
6795  026004  000034                  .WORD   L10106-.
6796
6797
6798                                  ;
6799                                  ;LOADER   SUBROUTINE USED BY THIS TEST TO LOAD MICRO CODE INTO A M8206
6800                                  ;
6801
6802  026006  012711  002000  LOADER: MOV     #2000,(R1)
6803
6804  026012  005000                  CLR     R0
6805
6806  026014  010061  000004  10$:    MOV     R0,4(R1)        ;SET ADDR.
6807  026020  005200                  INC     R0
6808  026022  011561  000006          MOV     (R5),6(R1)      ;WRIITE MICRO CODE.
6809  026026  022527  177777          CMP     (R5)+,#177777   ;SEE IF TERM.
6810  026032  001370                  BNE     10$
6811  026034  005011                  CLR     (R1)
6812  026036  000205                  RTS     R5
6813
6814  026040                          ENDTST
6815  026040                  L10106:
6816  026040  104401                  TRAP    C$ETST
6817
6818  026042                          BADHEAD
6819                                  ;*************** TEST 37 ***************
6820                                  ;*
6821                                  ;*NEGATIVE ADDRESS TEST.
6822                                  ;*      IN THIS TEST, WE'LL MAKE SURE THAT THE M8207
6823                                  ;*      DOES NOT RESPOND TO AN ADDRESS THAT ISN'T ASSIGNED
6824                                  ;*      TO IT
6825                                  ;*
6826  026042                          BADHEAD
6827                                  ;*************** TEST 37 ***************
6828
6829  026042                          BGNTST
```

```
6830  026042                          T37::
6831  026042                                MYINT
6832  026042  013701  002516                MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
6833
6834  026046  012711  000641                MOV     #641,(R1)       ;PUT A DEFINITE PATTERN IN MCPU.
6835  026052  012737  026130  000004        MOV     #20$,@#4        ;SET UP FOR TRAPS FROM NON-EX.
6836  026060  005037  000006                CLR     @#6
6837  026064  012702  160000                MOV     #160000,R2      ;GET STARTING ADDRESS.
6838
6839  026070  022712  000641        10$:    CMP     #641,(R2)       ;SEE IF CONTENTS OF THE ADDRESS
6840                                                                ;POINTED TO BY R2 EQUALS THE CONTENTS
6841                                                                ;OF THE MCPU CSR.
6842  026074  001420                        BEQ     40$
6843
6844  026076  062702  000002        15$:    ADD     #2,R2           ;UPDATE ADDRESS.
6845  026102  020227  177700                CMP     R2,#177700      ;DONE?     ;BO
6846  026106  001370                        BNE     10$             ;NO-LOOP
6847
6848  026110  013737  002464  000004  17$:  MOV     SAVE4,@#4       ;RESTORE TRAP CATCHER
6849  026116  013737  002466  000006        MOV     SAVE6,@#6       ;FROM VALUES SAVED BY INIT SECTION
6850  026124                                EXIT    TST             ;EXIT, ALL DONE
6851  026124  104432                        TRAP    C$EXIT
6852  026126  000052                        .WORD   L10107-.
6853
6854  026130  062706  000004        20$:    ADD     #4,SP           ;SAVE FROM TRAP
6855  026134  000760                        BR      15$             ;LOOP
6856
6857  026136                        40$:    ;*OH NO, WE MAY HAVE A DUAL ADDRESS PROBLEM!
6858
6859  026136  012711  000174                MOV     #174,(R1)       ;WRITE NEW PATTERN IN MCPU CSR
6860  026142  022712  000174                CMP     #174,(R2)       ;DID NEW PATTERN SHOW UP IN ADDR?
6861  026146  001403                        BEQ     60$
6862
6863  026150  012711  000641        50$:    MOV     #641,(R1)       ;PUT OLD PATTERN BACK IN MCPU CSR.
6864  026154  000750                        BR      15$             ;LOOP
6865
6866  026156  020102                60$:    CMP     R1,R2           ;IS THIS THE MCPU ADDRESS?
6867  026160  001773                        BEQ     50$             ;YES-NO ERROR
6868
6869  026162                                ERROR   40              ;DUAL ADDRESS ERROR
6870  026166  104455                        TRAP    C$ERDF
6871  026170  000050                        .WORD   40
6872  026172  005055                        .WORD   EM0
6873  026174  010636                        .WORD   ERR40
6874  026176  000744                        BR      17$
6875
6876  026200                                ENDTST
6877  026200                        L10107:
6878  026200  104401                        TRAP    C$ETST
6879
6880  026202                                BADHEAD
6881                                        ;************** TEST 38 **************
6882                                        ;*
6883                                        ;*BYTE ADDRESSING TEST
6884                                        ;*      HERE, WE'RE GOING TO MAKE SURE THAT WE CAN
6885                                        ;*      WRITE INTO ONLY A HIGH OR LOW BYTE OF THE MCPU.
```

```
6886                                            ;*
6887  026202                                    BADHEAD
6888                                            ;************** TEST 38 ***************
6889
6890  026202                                    BGNTST
6891  026202                              138::
6892  026202                                    MYINT
6893  026202  013701  002516                    MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
6894  026206  005061  000002                    CLR     2(R1)                   ;CLEAR CSR
6895  026212  112761  177777  000002            MOVB    #-1,2(R1)               ;WRITE ALL ONES INTO LOW BYTE
6896                                                                            ;OF CSR
6897  026220  032761  177400  000002            BIT     #177400,2(R1)           ;SEE IF HIGH BYTE GOT WRITTEN
6898  026226  001410                            BEQ     10$
6899
6900  026230                                    ERROR   41                      ;HIGH BYTE GOT WRITTEN INTO ON A LOW BYTE
6901  026234  104455                            TRAP    C$ERDF
6902  026236  000051                            .WORD   41
6903  026240  005055                            .WORD   EMO
6904  026242  010702                            .WORD   ERR41
6905  026244                                    ESCAPE  TST                     ;OPERATION
6906  026244  104410                            TRAP    C$ESCAPE
6907  026246  000040                            .WORD   L10110-.
6908
6909  026250  005061  000002              10$:  CLR     2(R1)
6910  026254  112761  177777  000003            MOVB    #-1,3(R1)               ;WRITE INTO HIGH BYTE
6911  026262  032761  000377  000002            BIT     #377,2(R1)              ;SEE IF LOW BYTE GOT WRITTEN
6912  026270  001406                            BEQ     20$
6913
6914  026272                                    ERROR   42                      ;LOW BYTE GOT WRITTEN INTO ON A
6915  026276  104455                            TRAP    C$ERDF
6916  026300  000052                            .WORD   42
6917  026302  005055                            .WORD   EMO
6918  026304  010744                            .WORD   ERR42
6919                                                                            ;HIGH BYTE OPERATION.
6920
6921  026306                              20$:
6922  026306                                    ENDTST
6923  026306                              L10110:
6924  026306  104401                            TRAP    C$ETST
6925
6926  026310                                    BADHEAD
6927                                            ;************** TEST 39 ***************
6928                                            ;*
6929                                            ;*IN THIS TEST WE'RE GOING TO MAKE SURE THAT THE PC
6930                                            ;*REG COUNTS UP PROPERLY. THE PC REG SHOULD INCREMENT
6931                                            ;*ONCE AFTER EACH INSTRUCTION.
6932                                            ;*
6933  026310                                    BADHEAD
6934                                            ;************** TEST 39 ***************
6935
6936  026310                                    BGNTST
6937  026310                              T39::
6938  026310                                    SKIP07  10$                     ;ONLY DO IF M8207
6939                                            ;GOTO 10$ IF M8207
6940  026320                                    EXIT TST
6941  026320  104432                            TRAP    C$EXIT
```

```
6942  026322  000122                        .WORD   L10111-.
6943
6944  026324                        10$:    MYINT
6945  026324  013701  002516                MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
6946  026330                                MSTCLR
6947  026334                                ROMCLK
6948  026334  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6949  026340  000400                        400
6950  026342                                ROMCLK
6951  026342  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6952  026346  061233                        61233
6953  026350                                SROMCLK
6954  026350  004537  003100                JSR     R5,..SROMCLK
6955  026354  100000                        100000
6956  026356  012705  000001                MOV     #1,R5           ;START AT ZERO
6957
6958  026362                        20$:    ROMCLK                  ;READ PC HIGH REG.
6959  026362  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6960  026366  121265                        121265
6961
6962  026370                                ROMCLK                  ;READ PC LOW REG.
6963  026370  004537  003044                JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
6964  026374  121244                        121244
6965  026376  016104  000004                MOV     4(R1),R4        ;GET WHOLE PICTURE
6966  026402  042704  170000                BIC     #170000,R4
6967  026406  020405                        CMP     R4,R5           ;INCREMENT OK?
6968  026410  001410                        BEQ     30$
6969
6970  026412                                ERROR   47              ;PC FAILED TO INCREMENT PROPERLY
6971  026416  104455                        TRAP    C$ERDF
6972  026420  000057                        .WORD   47
6973  026422  005055                        .WORD   EMO
6974  026424  011264                        .WORD   ERR47
6975                                                                ;SHOULD INCREMENT BY ONE
6976                                                                ;FOR EACH INSTRUCTION.
6977  026426                                ESCAPE  TST
6978  026426  104410                        TRAP    C$ESCAPE
6979  026430  000014                        .WORD   L10111-.
6980
6981  026432  062705  000002        30$:    ADD     #2,R5           ;UPDATE EXPECTED ADDRESS BY 2.
6982  026436  020527  000777                CMP     R5,#777
6983  026442  001347                        BNE     20$
6984
6985  026444                                ENDTST
6986  026444                        L10111:
6987  026444  104401                        TRAP    C$ETST
6988
6989  026446                                BADHEAD
6990                                        ;*************** TEST 40 ***************
6991                                        ;*
6992                                        ;*IN THIS TEST WE'LL MAKE SURE THAT "BRANCH FIELD H" DOESN'T
6993                                        ;*GET SUCH HIGH.
6994                                        ;*FIRST WE'LL CLEAR THE PC HIGH REG. THEN WE'LL DO A BRANCH INSTR
6995                                        ;*WITH BAB BITS 11&12 SET. IF PCR BITS 8&9 SET THEN WE'LL KNOW
6996                                        ;*WE WERE SUCCESSFUL IF PCR BITS 8&9 FAIL TO SET, WE'LL KNOW
6997                                        ;*THAT THE MUX SELECTED THE WRONG INPUT TO BE CLOCKED INTO THE PCR.
```

```
                                        ;*
026446                                  BADHEAD
                                        ;*************** TEST 40 ***************

026446                                  BGNTST
026446                          T40::
026446                                  SKIP07  10$             ;ONLY DO IF M8207
                                        ;GOTO 10$ IF M8207
026456                                  EXIT TST
026456  104432                          TRAP    C$EXIT
026460  000062                          .WORD   L10112-.

026462                          10$:    MYINT                   ;INITIALIZE PARAMETERS
026462  013701  002516                  MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
026466                                  MSTCLR                  ;CLEAR DEVICE.

026472                                  ROMCLK                  ;DO A "BRANCH ALWAYS" WITH
026472  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
026476  114400                          114400                  ;BAB BITS 11&12 SET THIS SHOULD CLOCK
                                                                ;THESE BITS INTO BITS 8&9 OF THE PCR.
026500                                  ROMCLK                  ;NOW READ THE PCR HIGH
026500  004537  003044                  JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
026504  121265                          121265                  ;AND PUT INTO PORT5.
                                                                ;REG. BR NO CLK OF BAB BITS
026506  116105  000005                  MOVB    5(R1),R5        ;READ THE PCR.
026512  112704  000003                  MOVB    #3,R4           ;EXPECT BITS 8,9 TO BE SET.
026516  042705  000374                  BIC     #374,R5         ;STRIP ANY JUNK
026522  020405                          CMP     R4,R5           ;OK?
026524  001406                          BEQ     20$

026526                                  ERROR   15              ;"BRANCH FIELD H" STUCK HIGH OR
026532  104455                          TRAP    C$ERDF
026534  000017                          .WORD   15
026536  005055                          .WORD   EM0
026540  010006                          .WORD   ERR15
                                                                ;OTHER PROBLEM IN THIS AREA.

026542                          20$:
026542                                  ENDTST
026542                          L10112:
026542  104401                          TRAP    C$ETST

026544                                  BADHEAD
                                        ;*************** TEST 41 ***************
                                        ;*
                                        ;*IN THIS TEST WE'RE GOING TO MAKE SURE THAT ONLY SP0
                                        ;*IS SELECTED FOR SOURCE WHEN THE DESTINATION
                                        ;*IS THE OUTBUS
                                        ;*FIRST WE'LL WRITE EACH SP ADDRS INTO ITSELF THEN WE'LL
                                        ;*MOV SP TO OBUS4. THAT SHOULD SELECT
                                        ;*SP ADDRESS 0. IF ANY OTHER DATA SHOWS UP, WE'LL
                                        ;*BLAME IT ON THE SELECTION OF A DIFFERENT SCRATCH PAD.
026544                                  BADHEAD
                                        ;*************** TEST 41 ***************

026544                                  BGNTST
```

```
7054  026544                          T41::
7055  026544                                MYINT
7056  026544  013701  002516                MOV     KMCSR,R1            ;RECORD DEVICE ADDR.
7057  026550  005005                        CLR     R5                  ;START WITH ADDR-ZERO
7058
7059  026552  042737  000017  026574  10$:  BIC     #17,20$             ;STRIP SP ADDR FIELD FROM INSTR
7060  026560  010561  000004                MOV     R5,4(R1)            ;PUT SP ADDR INTO PORT4.
7061  026564  050537  026574                BIS     R5,20$              ;ADD SP ADDR TO INSTR.
7062  026570                                ROMCLK
7063  026570  004537  003044                JSR     R5,.ROMCLK          ;CLOCK INSTRUCTION
7064  026574  123100                 20$:   123100                      ;WRITE TO SP
7065  026576  005205                        INC     R5                  ;UPDATE ADDRESS
7066  026600  120527  000020                CMPB    R5,#20              ;IF NOT THROUGH, REPEAT.
7067  026604  001362                        BNE     10$
7068
7069  026606                                ROMCLK                      ;NOW MOV SPO TO OBUS* PORT4
7070  026606  004537  003044                JSR     R5,.ROMCLK          ;CLOCK INSTRUCTION
7071  026612  061204                        061204                      ;
7072  026614  116104  000004                MOVB    4(R1),R4            ;READ PORT4 IT S/B ZERO
7073  026620  001410                        BEQ     30$
7074  026622  012705  000000                MOV     #0,R5
7075  026626                                ERROR   43                  ;SPO NOT SELECTED FOR SOURCE-SEE
7076  026632  104455                        TRAP    C$ERDF
7077  026634  000053                        .WORD   43
7078  026636  005055                        .WORD   EM0
7079  026640  011006                        .WORD   ERR43
7080                                                                    ;DISCUSSION IN HEADER.
7081
7082  026642                          30$:  ENDTST
7083  026642                          L10113:
7084  026642  104401                        TRAP    C$ETST
7085
7086  026644                                BADHEAD
7087                                        ;*************** TEST 42 ***************
7088                                        ;*
7089                                        ;*IN THIS TEST WE ARE GOING TO MAKE SURE THAT THE
7090                                        ;*SIGNAL "MOV INST H" (AND ITS ASSOC. TRIBS) DOESN'T GET
7091                                        ;*STUCK HIGH. IN ORDER TO DO THIS WE'LL CLEAR THE PC HIGH REG
7092                                        ;*PUT KNOWN DATA IN THE BREG AND SP1 THEN WE'LL A BRANCH
7093                                        ;*WITH CROM BITS 0-3 SET AS WELL AS CROM BIT 9 WITH CROM BITS 8 AND 11 CLEAR.
7094                                        ;*IF "MOV INST H" GETS STUCK HIGH, THE PC REG HIGH WILL GET LOADED
7095                                        ;*WITH THE CONTENTS OF THE ALU
7096  026644                                BADHEAD
7097                                        ;*************** TEST 42 ***************
7098
7099  026644                                BGNTST
7100  026644                          T42::
7101  026644                                SKIP07  10$                 ;ONLY DO IF M8207
7102                                        ;GOTO 10$ IF M8207
7103  026654                                EXIT TST                    ;ELSE EXIT
7104  026654  104432                        TRAP    C$EXIT
7105  026656  000110                        .WORD   L10114-.
7106
7107  026660                          10$:  MYINT                       ;DO INITIAL TEST SET-UP.
7108  026660  013701  002516                MOV     KMCSR,R1                    ;RECORD DEVICE ADDR.
7109  026664                                MSTCLR                      ;DO A MASTER CLEAR.
```

```
7110  026670  005737  002470            TST     RUNB
7111  026674  001034                    BNE     20$
7112
7113                                    ;TO RUN THIS SECTION OF CODE YOU MUST TURN SW7 OF SWITCH PACK #E28
7114                                    ;OFF SO THAT M8207 NOT SELFSTARTING.
7115
7116  026676  012761  000002  000004    MOV     #2,4(R1)        ;PUT A 2 INTO SP1
7117  026704                            ROMCLK                  ;PORT4 TO SCRATCH PAD 1
7118  026704  004537  003044            JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
7119  026710  123101                    123101
7120  026712  012761  000004  000004    MOV     #4,4(R1)
7121  026720                            ROMCLK
7122  026720  004537  003044            JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
7123  026724  123100                    123100
7124  026726                            ROMCLK                  ;NOW DO A BRANCH ON C-BIT SET
7125  026726  004537  003044            JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
7126  026732  141201                    141201                  ;BASED ON SP CONTENTS
7127                                                            ;OK-WHAT WE ARE REALLY
7128                                                            ;INTERESTED IN IS SEEING IF THE
7129                                                            ;PC HIGH REG GETS LOADED WITH
7130                                                            ;THE CONTENTS OF THE ALU (2)
7131                                                            ;IF THIS OCCURS, WE CAN PROBABLY
7132                                                            ;SAY THAT "MOV INSTR" REMAINED
7133                                                            ;HIGH.
7134  026734                            ROMCLK                  ;READ PC HIGH, PUT INTO PORT5
7135  026734  004537  003044            JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
7136  026740  121265                    121265
7137  026742  116104  000005            MOVB    5(R1),R4        ;READ PC REG HIGH FROM PORT
7138  026746  001407                    BEQ     20$             ;SHOULD BE CLEAR
7139  026750  005005                    CLR     R5
7140
7141  026752                            ERROR   15              ;ERROR-PC REG HIGH S/B CLEAR-SEE HEADER
7142  026756  104455                    TRAP    C$ERDF
7143  026760  000017                    .WORD   15
7144  026762  005055                    .WORD   EM0
7145  026764  010006                    .WORD   ERR15
7146                                                            ;DISCUSSION.
7147
7148  026766                    20$:
7149  026766                            ENDTST
7150  026766                    L10114:
7151  026766  104401                    TRAP    C$ETST
7152
7153  026770                            BADHEAD
7154                                    ;*************** TEST 43 ***************
7155                                    ;*TEST THAT MASTER CLEAR, CLEARS BITS IN THE NPR CONTROL REGISTER AND
7156                                    ;*MICROPROCESSOR MISCELLANEOUS REGISTER-FIRST WE'LL SET THE
7157                                    ;*PRIORITY UP SO THAT WHEN WE SET THE BUS REQUEST BIT THAT IT WON'T BUG US
7158                                    ;*THEN WE'LL SET ALL THE BITS IN BOTH REGS EXCEPT THE
7159                                    ;*NPR REQUEST. WE'LL LOOK TO SEE THAT ALL GOT SET, NEXT
7160                                    ;*WE'LL DO A MASTER CLEAR AND BE SURE THAT THEY ALL
7161                                    ;*CLEAR.
7162  026770                            BADHEAD
7163                                    ;*************** TEST 43 ***************
7164
7165  026770                            BGNTST
```

```
7166  026770                                  T43::
7167  026770                                        MYINT
7168  026770  013701  002516                        MOV     KMCSR,R1                ;RECORD DEVICE ADDR.
7169  026774                                        MSTCLR
7170  027000                                        SETPRI  #PRI07          ;DON'T ALLOW INTERRUPTS.
7171  027000  012700  000340                        MOV     #PRI07,R0
7172  027004  104441                               TRAP    C$SPRI
7173  027006  012761  177777  000004              MOV     #-1,4(R1)       ;DATA TO BE SET
7174  027014  042761  000002  000004              BIC     #2,4(R1)        ;DON'T SET AC LOW!
7175  027022                                        ROMCLK
7176  027022  004537  003044                       JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
7177  027026  121111                               121111                  ;PUT INTO MISC REG.
7178  027030  042761  000400  000004              BIC     #400,4(R1)      ;DON'T SET NPR BIT
7179  027036                                        ROMCLK
7180  027036  004537  003044                       JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
7181  027042  121130                               121130                  ;PUT INTO NPR REG
7182  027044                                        ROMCLK
7183  027044  004537  003044                       JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
7184  027050  121225                               121225                  ;MOV MISC REG (11) TO PORT5
7185
7186  027052                                        ROMCLK
7187  027052  004537  003044                       JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
7188  027056  121204                               121204                  ;MOVE NPR REG (10) TO PORT4
7189  027060  012737  146636  002452              MOV     #146636,$GDDAT  ;EXPECT ALL TO SET
7190  027066  016104  000004                       MOV     4(R1),R4        ;READ WHAT HAPPEN
7191  027072  042704  030000                       BIC     #030000,R4
7192  027076  023704  002452                       CMP     $GDDAT,R4       ;DID ALL BITS GET SET?
7193  027102  001410                               BEQ     10$             ;YES CONTINUE.
7194  027104                                        BRESET
7195  027104  104433                               TRAP    C$RESET
7196  027106                                        ERROR   46              ;SO SORT OF PROBLEM SETTING BITS
7197  027112  104455                               TRAP    C$ERDF
7198  027114  000056                               .WORD   46
7199  027116  005055                               .WORD   EM0
7200  027120  011214                               .WORD   ERR46
7201                                                                        ;IN THE NPR AND/OR MISC REG.
7202  027122                                        CKLOOP
7203  027122  104406                               TRAP    C$CLP1
7204
7205  027124  152761  000100  000001  10$:       BISB    #100,1(R1)      ;SET MASTER CLEAR
7206  027132  142761  000300  000001              BICB    #300,1(R1)      ;CLEAR MASTER CLEAR
7207
7208  027140                                        ROMCLK
7209  027140  004537  003044                       JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
7210  027144  121225                               121225                  ;MOV MISC REG (11) TO PORT5
7211
7212  027146                                        ROMCLK
7213  027146  004537  003044                       JSR     R5,..ROMCLK     ;CLOCK INSTRUCTION
7214  027152  121204                               121204                  ;MOV NPR REG (10) TO PORT4
7215  027154  016104  000004                       MOV     4(R1),R4        ;READ RESULTS
7216  027160  005037  002452                       CLR     $GDDAT          ;EXPECT ZERO
7217  027164  042704  010000                       BIC     #010000,R4      ;STRIP PROG CLK BIT
7218  027170  001407                               BEQ     20$             ;IF ALL ZERO, EVERYTHING COOL.
7219
7220  027172                                        ERROR   46              ;MASTER CLEAR FAILED TO CLEAR
7221  027176  104455                               TRAP    C$ERDF
```

```
7222  027200  000056                        .WORD   46
7223  027202  005055                        .WORD   EMO
7224  027204  011214                        .WORD   ERR46
7225                                                      ;SOME BITS IN THE NPR AND/OR MISC REGS.
7226  027206                                CKLOOP
7227  027206  104406                        TRAP    C$CLP1
7228
7229  027210                          20$:
7230  027210  012761  000014  000004        MOV     #14,4(R1)       ;NOW WE ARE GOING TO TRY TO
7231  027216                                ROMCLK                  ;SET THE EXT BITS (16&17) IN THE NPR REG.
7232  027216  004537  003044                JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
7233  027222  121110                        121110                  ;IF MASTER CLEAR FAILED TO CLEAR ITSELF
7234  027224                                ROMCLK                  ;THEN WE WILL BE UNABLE TO SET
7235  027224  004537  003044                JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
7236  027230  121205                        121205                  ;THESE BITS
7237  027232  116104  000005                MOVB    5(R1),R4        ;READ REG
7238  027236  012737  000014  002452        MOV     #14,$GDDAT      ;STORE GOOD
7239  027244  023704  002452                CMP     $GDDAT,R4       ;DID BITS SET?
7240  027250  001407                        BEQ     30$             ;YES-CONTINUE
7241
7242  027252                                ERROR   46              ;MASTER CLEAR FAILED TO CLEAR
7243  027256  104455                        TRAP    C$ERDF
7244  027260  000056                        .WORD   46
7245  027262  005055                        .WORD   EMO
7246  027264  011214                        .WORD   ERR46
7247                                                      ;ITSELF, THUS PROHIBITING US FROM
7248                                                      ;FURTHER SETTING BITS IN THE NPR REG.
7249  027266                                CKLOOP
7250  027266  104406                        TRAP    C$CLP1
7251
7252  027270                          30$:  BRESET                  ;NOW WE'LL SEE IF A BUS RESET CLEARS
7253  027270  104433                        TRAP    C$RESET
7254                                                      ;THESE BITS.
7255  027272  005737  002470                TST     RUNB            ;CAN'T DO THIS
7256  027276  001016                        BNE     40$             ;TEST IF RUN SW SET.
7257  027300                                ROMCLK
7258  027300  004537  003044                JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
7259  027304  121204                        121204                  ;READ MISC REG
7260  027306  116104  000004                MOVB    4(R1),R4
7261  027312  001410                        BEQ     40$             ;IF ZERO-END TST
7262
7263  027314  005037  002452                CLR     $GDDAT          ;S/B ZERO
7264
7265  027320                                ERROR   46              ;BUS RESET FAILED TO CLEAR NPR REG
7266  027324  104455                        TRAP    C$ERDF
7267  027326  000056                        .WORD   46
7268  027330  005055                        .WORD   EMO
7269  027332  011214                        .WORD   ERR46
7270
7271                                                      ;MASTER CLEAR WAS ABLE TO LOOK TO THE
7272                                                      ;CIRCUITRY THAT CONVERTS BUS INIT
7273                                                      ;TO "CLEAR"
7274  027334                          40$:
7275  027334                                ENDTST
7276  027334                          L10115:
7277  027334  104401                        TRAP    C$ETST
```

```
                        .SBTTL HARDWARE PARAMETER CODING SECTION



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

027336                          BGNHRD
027336  000016                  .WORD L10116-L$HARD/2
027340                  L$HARD::

027340                          GPRMD   WPM,0,0,7,0,7,YES
027340  000032                  .WORD   T$CODE
027342  027374                  .WORD   WPM
027344  000007                  .WORD   7
027346  000000                  .WORD   T$LOLIM
027350  000007                  .WORD   T$HILIM
027352                          GPRMA   ADDRES,2,0,160000,177776,YES
027352  001031                  .WORD   T$CODE
027354  027446                  .WORD   ADDRES
027356  160000                  .WORD   T$LOLIM
027360  177776                  .WORD   T$HILIM
                        ;       GPRMA   VECTOR,4,0,0,674,YES
                        ;       GPRMD   PRIRTY,6,0,7000,4,7,YES
                        ;       GPRMD   LNUNIT,10,0,3,0,3,YES
                        ;       GPRMD   SWPAC1,12,0,377,0,377,YES
                        ;       GPRMD   SWPAC2,14,0,377,0,377,YES
                        ;       GPRMD   LOOPBK,16,0,40000,0,1,YES
027362                          GPRMD   ISRUN,24,0,7,0,1,YES
027362  012032                  .WORD   T$CODE
027364  030044                  .WORD   ISRUN
027366  000007                  .WORD   7
027370  000000                  .WORD   T$LOLIM
027372  000001                  .WORD   T$HILIM
027374                          ENDHRD
                                .EVEN
027374                  L10116:

027374  044127  041511  020110  WPM:    .ASCIZ  'WHICH MICRO-CPU? (0=M8200,4=M8204,7=M8207'
027402  044515  051103  026517
027410  050103  037525  024040
027416  036460  034115  030062
027424  026060  036464  034115
027432  030062  026064  036467
027440  034115  030062  000067
027446  044515  051103  026517  ADDRES: .ASCIZ  /MICRO-CPU  CSR ADDRESS : /
027454  050103  020125  041440
027462  051123  040440  042104
027470  042522  051523  035040
027476  000040
```

```
027500  044515  051103  026517   VECTOR: .ASCIZ  /MICRO-CPU VECTOR ADDRESS : /
027506  050103  020125  042526
027514  052103  051117  040440
027522  042104  042522  051523
027530  035040  000040
027534  044515  051103  026517   PRIRTY: .ASCIZ  /MICRO-CPU PRIORITY LEVEL : /
027542  050103  020125  051120
027550  047511  044522  054524
027556  046040  053105  046105
027564  035040  000040
027570  044127  041511  020110   LNUNIT: .ASCIZ  /WHICH LINE UNIT (0-3)? 0=NONE,1=M8201,2=M8202,3=M8203 : /
027576  044514  042516  052440
027604  044516  020124  030050
027612  031455  037451  030040
027620  047075  047117  026105
027626  036461  034115  030062
027634  026061  036462  034115
027642  030062  026062  036463
027650  034115  030062  020063
027656  020072     000
027661     123  044527  041524   SWPAC1: .ASCIZ  /SWITCH PACK #1 (DDCMP LINE #) : /
027666  020110  040520  045503
027674  021440  020061  042050
027702  041504  050115  046040
027710  047111  020105  024443
027716  035040  000040
027722  053523  052111  044103   SWPAC2: .ASCIZ  /SWITCH PACK #2 (BM873 BOOT ADR) : /
027730  050040  041501  020113
027736  031043  024040  046502
027744  033470  020063  047502
027752  052117  040440  051104
027760  020051  020072     000
027765     127  046111  020114   LOOPBK: .ASCIZ  /WILL TEST CONNECTOR(S) BE USED ? 0=NO,1=YES : /
027772  042524  052123  041440
030000  047117  042516  052103
030006  051117  051450  020051
030014  042502  052440  042523
030022  020104  020077  036460
030030  047516  030454  054475
030036  051505  035040  000040
030044  044515  051103  026517   ISRUN:  .ASCIZ  "MICRO-PROCESSOR RUN SWITCH  TYPE 0 IF OFF, 1 IF ON :"
030052  051120  041517  051505
030060  047523  020122  052522
030066  020116  053523  052111
030074  044103  020040  054524
030102  042520  030040  044440
030110  020106  043117  026106
030116  030440  044440  020106
030124  047117  035040     000

        030132                   .EVEN
```

```
7390
```

```
                                .SBTTL  SOFTWARE PARAMETER CODING SECTION


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

030132                                  BGNSFT
030132  000000                          .WORD L10117-L$SOFT/2
030134                          L$SOFT::

030134                                  ENDSFT
                                        .EVEN
030134                          L10117:

                                .EVEN




030134                                          LASTAD
                                                .EVEN
030134  000000                                  .WORD   0
030136  000000                                  .WORD   0
030140                                  L$LAST::

        000001                  .END
```

```
ADDRES  027446          C$EDIT= 000003          EM1     005056          F$BGN = 000040          I$HRD = 000041
ADR   = 000020 G        C$ERDF= 000055          EM10    005260          F$CLEA= 000007          I$INIT= 000041
ADR1    012600          C$ERHR= 000056          EM11    005312          F$DU  = 000016          I$MOD = 000040
ADR2    012604          C$ERRO= 000060          EM12    005333          F$END = 000041          I$MSG = 000041
ADR4    012444          C$ERSF= 000054          EM16    005353          F$HARD= 000004          I$PROT= 000040
ADR5    012450          C$ERSO= 000057          EM17    005434          F$HW  = 000013          I$PTAB= 000041
ASSEMB= 000010          C$ESCA= 000010          EM2     005076          F$INIT= 000006          I$PWR = 000041
BIT0  = 000001 G        C$ESEG= 000005          EM27    005463          F$JMP = 000050          I$RPT = 000041
BIT00 = 000001 G        C$ESUB= 000003          EM29    005512          F$MOD = 000000          I$SEG = 000041
BIT01 = 000002 G        C$ETST= 000001          EM3     005131          F$MSG = 000011          I$SETU= 000041
BIT02 = 000004 G        C$EXIT= 000032          EM35    005434          F$PROT= 000021          I$SFT = 000041
BIT03 = 000010 G        C$GETB= 000026          EM4     005144          F$PWR = 000017          I$SRV = 000041
BIT04 = 000020 G        C$GETW= 000027          EM5     005171          F$RPT = 000012          I$SUB = 000041
BIT05 = 000040 G        C$GMAN= 000043          EM6     005216          F$SEG = 000003          I$TST = 000041
BIT06 = 000100 G        C$GPHR= 000042          EM7     0052            F$SOFT= 000005          J$JMP = 000167
BIT07 = 000200 G        C$GPLO= 000030          ENDBUG  003044          F$SRV = 000010          KMACTV  002376
BIT08 = 000400 G        C$GPRI= 000040          ENDIT   012040          F$SUB = 000002          KMCSR   002516
BIT09 = 001000 G        C$INIT= 000011          ERRFLG  002350          F$SW  = 000014          KMCSRH  002520
BIT1  = 000002 G        C$INLP= 000020          ERR1    006032 G        F$TEST= 000001          KMCTL   002522
BIT10 = 002000 G        C$MANI= 000050          ERR10   007154 G        GETPRM  011462          KMNUM   002400
BIT11 = 004000 G        C$MEM = 000031          ERR11   007276 G        G$CNTO= 000200          KMPO4   002524
BIT12 = 010000 G        C$MSG = 000023          ERR12   007420 G        G$DELM= 000372          KMPO6   002526
BIT13 = 020000 G        C$OPEN= 000034          ERR13   007542 G        G$DISP= 000003          KMRLVL  002510
BIT14 = 040000 G        C$PNTB= 000014          ERR14   007664 G        G$EXCP= 000400          KMRVEC  002506
BIT15 = 100000 G        C$PNTF= 000017          ERR15   010006 G        G$HILI= 000002          KMTLVL  002514
BIT2  = 000004 G        C$PNTS= 000016          ERR16   010134 G        G$LOLI= 000001          KMTVEC  002512
BIT3  = 000010 G        C$PNTX= 000015          ERR17   010256 G        G$NO  = 000000          LNUNIT  027570
BIT4  = 000020 G        C$QIO = 000377          ERR2    006160 G        G$OFFS= 000400          LOADER  026006
BIT5  = 000040 G        C$RDBU= 000007          ERR29   010350 G        G$OFSI= 000376          LOCK    002340
BIT6  = 000100 G        C$REFG= 000047          ERR3    006306 G        G$PRMA= 000001          LOE   = 040000 G
BIT7  = 000200 G        C$RESE= 000033          ERR35   010472 G        G$PRMD= 000002          LOGDEV  002342
BIT8  = 000400 G        C$REVI= 000003          ERR36   010614 G        G$PRML= 000000          LOKFLG  002476
BIT9  = 001000 G        C$RFLA= 000021          ERR4    006434 G        G$RADA= 000140          LOOPBK  027765
BOE   = 000400 G        C$RPT = 000025          ERR40   010636 G        G$RADB= 000000          LOT   = 000010 G
CLKX    002360          C$SEFG= 000046          ERR41   010702 G        G$RADD= 000040          L$ACP   002110 G
CLRALL  003166          C$SPRI= 000041          ERR42   010744 G        G$RADL= 000120          L$APT   002036 G
CRAMA   003602          C$SVEC= 000037          ERR43   011006 G        G$RADO= 000020          L$AU    012144 G
CZDMQ   002000 G        C$TPRI= 000013          ERR44   011054 G        G$XFER= 000004          L$AUT   002070 G
C$AU  = 000052          DFPTBL  002262 G        ERR45   011124 G        G$YES = 000010          L$AUTO  012042 G
C$AUTO= 000061          DH0     005723          ERR46   011214 G        HELP  = 000000          L$CCP   002106 G
C$BRK = 000022          DH1     005724          ERR47   011264 G        HOE   = 100000 G        L$CLEA  012134 G
C$BSEG= 000004          DH2     005755          ERR5    006556 G        IBE   = 010000 G        L$CO    002032 G
C$BSUB= 000002          DH3     005775          ERR6    006700 G        IDU   = 000040 G        L$DEPO  002011 G
C$CEFG= 000045          DIAGMC= 000000          ERR7    007026 G        IER   = 020000 G        L$DESC  002312 G
C$CLCK= 000062          EF.CON= 000036 G        EVL   = 000004 G        INIFLG  002474          L$DESP  002076 G
C$CLEA= 000012          EF.NEW= 000035 G        E$END = 002100          INSTU   003616          L$DEVP  002060 G
C$CLOS= 000035          EF.PWR= 000034 G        E$LOAD= 000035          ISR   = 000100 G        L$DISP  002132 G
C$CLP1= 000006          EF.RES= 000037 G        FADR    002412          ISRUN   030044          L$DLY   002116 G
C$CVEC= 000036          EF.STA= 000040 G        FLAG    002406          IXE   = 004000 G        L$DTP   002040 G
C$DCLN= 000044          EMB1    005410          FM1     005046          I$AU  = 000041          L$DTYP  002034 G
C$DODU= 000051          EMB50   005545          FTIME   002462          I$AUTO= 000041          L$DU    012140 G
C$DRPT= 000024          EMB51   005577          F$AU  = 000015          I$CLN = 000041          L$DUT   002072 G
C$DU  = 000053          EM0     005055          F$AUTO= 000020          I$DU  = 000041          L$DVTY  002730 G
```

```
L$EF    002052 G    L10024  010634    L10111  026444    SETBR4  003240    T$NS1 = 000005
L$ENVI  002044 G    L10025  010700    L10112  026542    SETBR7  003250    T$NS2 = 000003
L$ETP   002102 G    L10026  010742    L10113  026642    SETC    003260    T$PTNU= 000000
L$EXP1  002046 G    L10027  011004    L10114  026766    SETZ    003312    T$SAVL= 177777
L$EXP4  002064 G    L10030  011052    L10115  027334    SFPTBL  002312 G  T$SEGL= 177777
L$EXP5  002066 G    L10031  011122    L10116  027374    SSTACK  002730    T$SEK0= 010002
L$HARD  027340 G    L10032  011212    L10117  030134    STAT    002356    T$SUBN= 000000
L$HIME  002120 G    L10033  011262    MASKX   002362    STAT1   002500    T$TAGL= 177777
L$HPCP  002016 G    L10034  011330    MEMLIM  002374    STAT2   002502    T$TAGN= 010120
L$HPTP  002022 G    L10035  011336    MEMSET  003474    STAT3   002504    T$TEMP= 000005
L$HW    002262 G    L10036  012040    MEMSZ   002436    STM     005627    T$TEST= 000053
L$ICP   002104 G    L10037  012132    MRO     002434    STRTSW  002354    T$TSTM= 177777
L$INIT  011340 G    L10040  012136    NEWST   011454    SUBRPC  002346    T$TSTS= 000001
L$LADP  002026 G    L10041  012142    NEXT    002336    SVCGBL= 000000    T$$AU  = 010042
L$LAST  030140 G    L10042  012144    ONE     002372    SVCINS= 000000    T$$AUT= 010037
L$LOAD  002100 G    L10043  012254    O$APTS= 000000    SVCSUB= 000000    T$$CLE= 010040
L$LUN   002074 G    L10044  012420    O$AU  = 000001    SVCTAG= 000000    T$$DU  = 010041
L$MREV  002050 G    L10045  012550    O$BGNR= 000000    SVCTST= 000000    T$$HAR= 010116
L$NAME  002000 G    L10046  012710    O$BGNS= 000000    SV05    003632    T$$HW  = 010001
L$PRIO  002042 G    L10047  013152    O$DU  = 000001    SWPAC1  027661    T$$INI= 010036
L$PROT  002122 G    L10050  013354    O$ERRT= 000000    SWPAC2  027722    T$$MSG= 010034
L$PRT   002112 G    L10051  013566    O$GNSW= 000000    S$LSYM= 010000    T$$PRO= 010000
L$REPP  002062 G    L10052  014110    O$POIN= 000001    TEMP    002440    T$$RPT= 010035
L$REV   002010 G    L10053  014466    O$SETU= 000000    TFM1    003666    T$$SEG= 010002
L$RPT   011332 G    L10054  014734    PATCH   002774    TFM2    003712    T$$SOF= 010117
L$SOFT  030134 G    L10055  015200    PNT   = 001000 G  TFM3    003730    T$$SW  = 010002
L$SPC   002056 G    L10056  015430    PRI   = 002000 G  TFM36   004017    T$$TES= 010115
L$SPCP  002020 G    L10057  015674    PRIRTY  027534    TFM4    003755    T1      012146 G
L$SPTP  002024 G    L10060  016140    PRI00 = 000000 G  TFM40   004263    T10     014470 G
L$STA   002030 G    L10061  016404    PRI01 = 000040 G  TFM41   004105    T11     014736 G
L$SW    002312 G    L10062  016650    PRI02 = 000100 G  TFM42   004174    T12     015202 G
L$TEST  002114 G    L10063  017114    PRI03 = 000140 G  TFM43   004344    T13     015432 G
L$TIML  002014 G    L10064  017360    PRI04 = 000200 G  TFM44   004427    T14     015676 G
L$UNIT  002012 G    L10065  017640    PRI05 = 000240 G  TFM45   004466    T15     016142 G
L10001  002310      L10066  020120    PRI06 = 000300 G  TFM45A  004524    T16     016406 G
L10002  002312      L10067  020374    PRI07 = 000340 G  TFM46   004647    T17     016652 G
L10003  006156      L10070  020650    PSTACK  002344    TFM47   004733    T18     017116 G
L10004  006304      L10071  021126    QV.FLG  002477    TFM5    003773    T19     017362 G
L10005  006432      L10072  021402    RAMDAT  003330    TMMC    005010    T2      012256 G
L10006  006554      L10073  021574    REGADR  002530    TYPE    002432    T20     017642 G
L10007  006676      L10074  021742    RETADR  002352    T$ARGC= 000001    T21     020122 G
L10010  007024      L10075  022320    ROMMAP  012146    T$CODE= 012032    T22     020376 G
L10011  007152      L10076  022556    RUN     002410    T$ERRN= 000056    T23     020654 G
L10012  007274      L10077  022772    RUNB    002470    T$EXCP= 000000    T24     021130 G
L10013  007416      L10100  023206    RUNINH  002472    T$FLAG= 000040    T25     021404 G
L10014  007540      L10101  023450    SAVACT  002402    T$GMAN= 000000    T26     021576 G
L10015  007662      L10102  024042    SAVE4   002464    T$HILI= 000001    T27     021744 G
L10016  010004      L10103  025114    SAVE6   002466    T$LAST= 000001    T28     022322 G
L10017  010132      L10104  025214    SAVNUM  002404    T$LOLI= 000000    T29     022560 G
L10020  010254      L10105  025362    SAVPC   002366    T$LSYM= 010000    T3      012422 G
L10021  010346      L10106  026040    SAVSP   002364    T$LTNO= 000053    T30     022774 G
L10022  010470      L10107  026200    SETBR0  003220    T$NEST= 000000    T31     023210 G
L10023  010612      L10110  026306    SETBR1  003230    T$NS0 = 000000    T32     023452 G
```

```
T33     024044 G        T42     026644 G        WROM    003422          $GDDAT  002452          $TMP0   002444
T34     025116 G        T43     026770 G        WTYPE   002414          $LSTIN= 000000          .       = 030140
T35     025216 G        T5      012712 G        X$ALWA= 000000          $LSTTA= 000000          .MSTCL  002756
T36     025364 G        T6      013154 G        X$FALS= 000040          $REG0   002430          .REGT   003050
T37     026042 G        T7      013356 G        X$OFFS= 000400          $REG1   002426          .ROMCL  003044
T38     026202 G        T8      013570 G        X$TRUE= 000020          $REG2   002424          .SROMC  003100
T39     026310 G        T9      014112 G        ZERO    002370          $REG3   002422
T4      012552 G        UAM   = 000200 G        $BDADR  002450          $REG4   002420
T40     026446 G        VECTOR  027500          $BDDAT  002454          $REG5   002416
T41     026544 G        WPM     027374          $GCADR  002446          $TEMP0  002442

. ABS.  030140     000


 ERRORS DETECTED:  0

 CZDMQB,CZDMQB/SOL/NL:TOC=SVC34R.MLB,CZDMQB.P11
 RUN-TIME: 42 47 .5 SECONDS
 RUN-TIME RATIO: 174/90=1.9
 CORE USED:  16K  (31 PAGES)
```